



Subandi, S.T., M. Kom.
Aulia Akhrian Syahidi, S.Pd., MTA.

BASIS DATA



Teori dan Praktik
Menggunakan Microsoft Office Access



Diterbitkan Atas Kerjasama
Deepublish dengan Politeknik Banjarmasin



BASIS DATA
Teori dan Praktik
Menggunakan Microsoft Office Access

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

BASIS DATA

Teori dan Praktik

Menggunakan Microsoft Office Access

Subandi, S.T., M.Kom.
Aulia Akhrian Syahidi, S.Pd., M.T.A.



**BASIS DATA:
TEORI DAN PRAKTIK MENGGUNAKAN MICROSOFT OFFICE ACCESS**

Penulis:
Subandi & Aulia Akhrian Syahidi

ISBN:
978-602-53458-1-4

ISBN Elektronik:
978-602-53809-3-8

Desain Sampul:
Rahma Indera

Editor dan Penyunting :
Aulia Akhrian Syahidi, S.Pd., MTA

Tata Letak:
Amira Dzatin Nabila

Proofreader:
Usy Izzani F.

Penerbit:
POLIBAN PRESS
Cetakan Pertama, 2018

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa ijin tertulis dari penerbit

Redaksi:
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp: (0511)3305052 Email: press@poliban.ac.id

Dicetak oleh:
PERCETAKAN DEEPUBLISH
Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoarjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id

Katalog Dalam Terbitan (KDT)
**Subandi & Aulia Akhrian Syahidi—Cet. 1. — Basis Data: Teori dan Praktik Menggunakan
Microsoft Office Access: Poliban Press, 2018.**

xii; 276 hlm.; 15.5x23 cm

KATA PENGANTAR

Alhamdulillah puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas segala limpahan rahmat dan hidayah-Nya sehingga buku yang berjudul **“Basis Data: Teori dan Praktik Menggunakan Microsoft Office Access”** dapat terselesaikan dengan baik. Tidak lupa kami ucapkan banyak terima kasih kepada semua pihak yang telah membantu selesainya buku ini. Buku ini merupakan panduan untuk Mata Kuliah Basis Data menggunakan Microsoft Office Access 2013 baik teori maupun praktik bagi mahasiswa.

Dalam buku ini, penulis berusaha untuk memberikan pemahaman yang terstruktur dan mendalam serta disertai contoh aplikasi sederhana bagi para pembaca khususnya untuk mahasiswa yang sedang memprogramkan Mata Kuliah Basis Data menggunakan Microsoft Office Access 2013, karena di dalam buku ini terdapat teori, langkah-langkah praktik, dan latihan.

Akhir kata, kami dari penulis mengharapkan saran dan kritik yang membangun agar buku ini selalu mengembangkan isinya. Harapan kami, semoga buku ini dapat bermanfaat, berguna, dan menambah wawasan serta pengetahuan baik bagi pembaca maupun mahasiswa yang sedang memprogramkan Mata Kuliah Basis Data.

Banjarmasin, November 2018

Penulis

DAFTAR ISI

KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
BAB I PENGANTAR BASIS DATA.....	1
1.1 Basis Data	2
1.2 Sistem Basis Data	4
1.3 Komponen Sistem Basis Data	5
1.4 Keuntungan Sistem Basis Data	6
1.5 Kerugian Sistem Basis Data.....	7
1.6 Database Management System (DBMS).....	7
1.7 Fungsi DBMS	7
1.8 Komponen Utama DBMS.....	8
1.9 Penyusun Sistem Basis Data	9
1.10 Abstraksi Data.....	11
1.11 Bahasa Basis Data.....	13
1.12 Sejarah Basis Data dan Aplikasi Praktis	14
1.12.1 Tahun 1950 Hingga Awal 1960	15
1.12.2 Akhir Tahun 1960 dan 1970-an.....	15
1.12.3 Tahun 1980-an	16
1.12.4 Awal Tahun 1990-An.....	17
1.12.5 Akhir tahun 1990-an.....	17
1.12.6 Tahun 2000 dan seterusnya	17
1.13 Perkembangan Basis Data.....	18
1.13.1 Kategori DBMS	18
1.13.2 DBMS dan Aplikasi	19
1.13.3 Arsitektur Sistem.....	20
1.13.4 Perkembangan DBMS Berorientasi Kepada Pemakai.....	20
Latihan Teori.....	21

BAB II	PENGANTAR BASIS DATA LANJUTAN I.....	23
2.1	Perancangan Basis Data	24
2.1.1	Perancangan Basis Data secara Logika	25
2.1.2	Perancangan Basis Data secara Fisik.....	26
2.2	Model Data	27
2.3	Model Data Hierarki (<i>Hirarchycal</i>)	28
2.4	Model Data Jaringan (<i>Network</i>)	30
2.5	Model Data Relasional (<i>Relational</i>).....	30
2.6	<i>Entity Relationship Diagram</i> (ERD).....	33
2.6.1	Simbol-Simbol ERD	34
2.6.2	Derajat Relationship.....	37
2.6.3	<i>Cardinality Ratio</i>	38
2.6.4	Langkah-Langkah Membuat ERD.....	39
2.6.5	Contoh Kasus	40
2.6.6	ERD dengan Kamus Data.....	41
	Latihan Praktik.....	42
BAB III	PENGANTAR BASIS DATA LANJUTAN II	43
3.1	Normalisasi (<i>Normalization</i>).....	44
3.1.1	Macam-Macam <i>Anomaly</i>	44
3.1.2	Ketergantungan Fungsional (<i>Functional Dependency</i>).....	45
3.1.3	Macam-Macam Atribut Tabel	47
3.1.4	Jenis-Jenis <i>Key</i>	48
3.1.5	Langkah-Langkah Pembentukan Normalisasi	49
3.1.6	Contoh Normalisasi.....	51
3.2	Lingkungan Basis Data	54
3.2.1	Masalah dalam Basis Data	54
3.2.2	Organisasi <i>File</i> Sistem Basis Data.....	55
3.2.3	Schema dan Subschema	56
3.2.4	Kamus Data.....	57
	Latihan Praktik.....	58

BAB IV	MENGENAL DAN MEMULAI BEKERJA DENGAN MICROSOFT OFFICE ACCESS 2013.....	59
4.1	Pengenalan Microsoft Office Access 2013	60
4.2	Keunggulan Microsoft Office Access 2013	62
4.3	Kelemahan Microsoft Office Access 2013.....	62
4.4	Komponen-Komponen Microsoft Office Access 2013	63
4.4.1	Tampilan layar depan Microsoft Office Access 2013	63
4.4.2	Beberapa Komponen Microsoft Office Access 2013	64
4.4.3	Ribbon Tab File	64
4.4.4	Ribbon Tab Home	66
4.4.5	Ribbon Tab Create	66
4.4.6	Ribbon Tab External Data.....	67
4.4.7	Ribbon Tab Database Tools	67
4.4.8	Tools	67
4.4.9	Sidebar	68
4.4.10	Field	68
4.5	Objek-Objek Microsoft Office Access 2013	69
4.5.1	Table	69
4.5.2	Query	69
4.5.3	Relationship	69
4.5.4	Form.....	70
4.5.5	Report.....	70
4.5.6	Macro	71
4.5.7	Module.....	71
4.6	Struktur Basis Data (<i>Database</i>).....	71
4.7	Memulai Bekerja di dalam Microsoft Office Access 2013	74
4.8	Membuat <i>Database</i>	75
4.9	Membuka <i>Database</i>	77
4.10	Menyimpan <i>Database</i>	79

4.11	Menutup <i>Database</i>	81
	Latihan Praktik.....	81
BAB V	MEMBUAT DAN MEMODIFIKASI <i>TABLE</i>	83
5.1	Merancang <i>Tabel</i>	84
5.2	Mengelola Jendela <i>Design View</i>	87
5.3	Membuat <i>Tabel</i>	90
5.4	Fungsi Primary Key dan Foreign Key.....	94
5.5	Memodifikasi <i>Tabel</i>	95
	5.5.1 Langkah Awal Memodifikasi <i>Tabel</i>	95
	5.5.2 Mengganti Nama <i>Field</i>	96
	5.5.3 Mengganti Jenis Data (<i>Data Type</i>).....	97
	5.5.4 Menyisipkan dan Menambahkan <i>Field</i>	98
	5.5.5 Menghapus <i>Field</i>	98
	5.5.6 Menghapus <i>Tabel</i> dari <i>Database</i>	99
5.6	Mengelola <i>Tabel</i>	100
	5.6.1 Meng- <i>input</i> Data dalam <i>Tabel</i>	100
	5.6.2 Mengedit atau Mengubah Data	100
	5.6.3 Menghapus Data.....	101
	5.6.4 Mencari Data.....	102
	5.6.5 Mencari dan Mengganti Data	104
	5.6.6 Mengurutkan Data dalam <i>Tabel</i>	106
	5.6.7 Menyaring Data Menggunakan <i>Filter</i>	107
	Latihan Praktik.....	108
BAB VI	MENGGUNAKAN FIELD PROPERTIES PADA DESIGN TABLE	111
6.1	Field Properties	112
6.2	Mengatur Format Masukan Data.....	114
6.3	Menentukan <i>Input Mask</i>	116
6.4	Menentukan Nilai Otomatis (<i>Default Value</i>).....	119
6.5	Menentukan Validasi Masukan Data (<i>Validation Rule</i>).....	120

6.6	Menentukan Pesan Kesalahan pada Masukan Data (Validation Text)	122
6.7	Menggunakan Fasilitas <i>Lookup</i>	124
	Latihan Praktik.....	130
BAB VII	RELATIONSHIP	133
7.1	Relationship	134
	Latihan Praktik.....	144
BAB VIII	MEMBUAT <i>QUERY</i>.....	148
8.1	Query	149
	Latihan Praktik.....	155
BAB IX	MENGIMPLEMENTASIKAN STUDI KASUS ERD MENJADI SEBUAH <i>DATABASE</i>.....	167
9.1	Memahami ERD	168
9.2	Membuat Struktur Data secara Konseptual dari ERD	168
	Latihan Praktik.....	169
9.3	Membuat Tabel pada <i>Database</i> Microsoft Office Access 2013 Berdasarkan ERD di Atas	169
BAB X	MERANCANG DAN MEMBUAT <i>FORM</i>.....	178
10.1	Form	179
10.2	Merancang dan Membuat <i>Form</i>	180
10.3	Membuat <i>Form</i> dengan <i>Form</i>	180
10.4	Membuat <i>Form</i> dengan <i>Blank Form</i>	182
10.5	Membuat <i>Form</i> dengan <i>Form Wizard</i>	183
10.6	Membuat Navigation Form	187
10.7	Membuat <i>Form</i> dengan <i>Form Design</i>	189
10.8	Menutup dan Menyimpan <i>Form</i>	193
10.9	Bentuk <i>Form</i>	194
10.10	Menjalankan <i>Form</i>	196
10.11	Memasukkan/ <i>Entry</i> Data Melalui <i>Form</i>	197
	Latihan Praktik.....	198

BAB XI	MERANCANG DAN MEMBUAT FORM	
	LANJUTAN.....	202
11.1	Fungsi <i>Control</i>	203
11.2	Jenis-Jenis <i>Control</i>	203
11.3	Membuat <i>Field</i> Entri Data Berbentuk <i>List Box</i>	205
11.4	Membuat <i>Field</i> Entri Data Berbentuk <i>Combo Box</i>	211
11.5	Membuat <i>Field</i> Entri Data Berbentuk <i>Option Group</i>	217
11.6	Membuat Tombol Perintah (<i>Command Button</i>)	222
	Latihan Praktik.....	227
	Latihan Teori.....	228
BAB XII	MERANCANG DAN MEMBUAT REPORT.....	229
12.1	Report	230
12.2	Membuat <i>Report</i>	230
12.2.1	Membuat <i>Report</i> dengan <i>Report</i>	230
12.2.2	Membuat Report dengan Blank Report	231
12.2.3	Membuat Report dengan Report Design	232
12.2.4	Membuat Report dengan Report Wizard.....	234
12.3	Menyaring Data pada <i>Report</i>	239
12.4	Menghitung Kelompok Data pada <i>Report</i>	242
12.5	Membuat Grafik pada <i>Report</i>	244
12.6	Menutup dan Menyimpan <i>Report</i>	250
12.7	Mencetak <i>Report</i>	252
	Latihan Praktik.....	253
BAB XIII	MEMBUAT MENU NAVIGASI DAN LOGIN.....	255
13.1	<i>Menu</i> Navigasi dan <i>Login</i>	256
13.2	Membuat <i>Menu</i> Navigasi dengan <i>Form</i> , Macro, dan Visual Basic Application (VBA).....	257
13.3	Membuat Keamanan Data dengan <i>Form Login</i>	266

13.4 Mengatur Tampilan Awal Ketika <i>Database</i> Dijalankan.....	271
Latihan Praktik.....	273
Proyek Akhir Mata Kuliah	273
DAFTAR PUSTAKA.....	274
TENTANG PENULIS.....	276



PENGANTAR BASIS DATA



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, dan menguasai konsep dasar mengenai basis data dengan baik.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu mengetahui dan memahami Dasar Sistem Basis Data
2. Mahasiswa mampu mengetahui dan memahami DBMS



1.1 Basis Data

Asal kata **basis data** (*database*) adalah **basis** (*base*) dan **data** (*data*). Menurut Fred R. McFadden et al (1999) dalam *Modern Database Management* menyebutkan bahwa **data** adalah fakta-fakta tentang segala sesuatu di dunia nyata yang dapat direkam dan disimpan pada media komputer.

Menurut Jeffrey A. Hoffer, Mary B. Prescott, dan Fred R. McFadden (2005: 5), **data** merupakan gambaran objek dan peristiwa yang mempunyai arti dan penting di lingkungan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI) **basis** adalah asas, dasar, himpunan, pangkalan, kumpulan.

Menurut Jeffrey A. Hoffer, Mary B. Prescott, dan Fred R. McFadden (2005: 4), **basis data** adalah sebuah kumpulan terorganisasi dari data-data yang berhubungan secara logika. Basis data biasanya dirancang untuk memenuhi kebutuhan-kebutuhan informasi dari *multiple user* dalam sebuah organisasi.

Menurut Raghu Ramakrishnan dan Johannes Gehrke (2003: 3), **basis data** adalah kumpulan data, umumnya mendeskripsikan aktivitas suatu organisasi yang berhubungan atau lebih. Misalnya, *database* universitas mungkin berisi informasi mengenai hal berikut:

- a. Entitas seperti mahasiswa, fakultas, mata kuliah, dan ruang kuliah.
- b. Hubungan antara entitas, seperti registrasi mahasiswa dalam mata kuliah, fakultas yang mengajarkan mata kuliah, dan penggunaan ruang untuk kuliah.

Menurut V.K. Pallaw (2008: 3), *database* didefinisikan sebagai satu koleksi dari data yang saling berhubungan, disimpan bersama-sama secara logika, yang didesain untuk menjumpai informasi sesuai dengan kebutuhan dari organisasi.

Menurut James Martin mendefinisikan bahwa **basis data** adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data, sehingga mudah untuk digunakan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada



program yang akan menggunakannya, data disimpan sedemikian rupa sehingga apabila ada penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

Menurut Rosa A. S. dan M. Shalahuddin (2013: 43-44), mengemukakan **basis data** adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan. Tujuan dari dibuatnya tabel-tabel pada basis data adalah untuk menyimpan data ke dalam tabel-tabel agar mudah untuk diakses.

Secara sederhana basis data dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data.



Gambar 1.1 Data dan Informasi

Sebagaimana diketahui, manajemen modern mengikutsertakan informasi sebagai sumber daya penting yang setara dengan sumber daya manusia, uang, mesin, dan material. Menurut Hoffer, Prescott, dan McFadden (2005: 5), **informasi** merupakan data yang telah diproses dengan cara meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Informasi merupakan bentuk penyajian data melalui mekanisme pemrosesan, berguna bagi pihak tertentu, misalnya seorang manajer. Bagi pihak manajemen, informasi merupakan bahan untuk pengambilan keputusan.

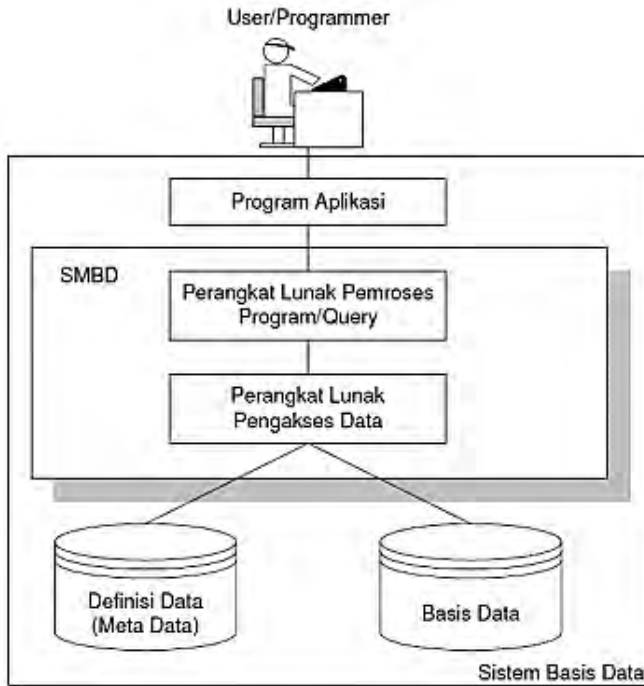


Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Dengan adanya komputer, data dapat disimpan dalam media penyimpan yang disebut *hard disk*. Dengan menggunakan media ini, kehadiran kertas yang digunakan untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama kalau dikemas dalam bentuk *database*.

1.2 Sistem Basis Data

Sistem Basis Data menurut Waliyanto (2000) adalah gabungan antara basis data dan perangkat lunak DBMS termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem. Sistem Basis Data merupakan suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan. Salah satu cara menyajikan data untuk mempermudah modifikasi adalah dengan cara pemodelan data. Model yang akan dipergunakan adalah *Entity Relationship Diagram*. Model *Entity Relationship Diagram* adalah representasi logika dari data pada suatu organisasi atau area bisnis tertentu dengan menggunakan *Entity* dan *Relationship*.



Gambar 1.2 Konsep Sistem Basis Data

(Kompilasi Ramez Elmasri, dkk 1994)

1.3 Komponen Sistem Basis Data

Sistem basis data merupakan sistem penyusunan berkas data yang saling terpadu. Mempunyai komponen-komponen sebagai berikut:

1. **Software** adalah perangkat lunak yang digunakan dalam suatu sistem basis data. Pengelolaan basis data secara fisik tidak dapat dilakukan pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang khusus yang disebut DBMS yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah, dimanipulasi, dan diambil kembali. *Software* yang termasuk DBMS seperti Dbase III+, Dbase IV, Foxbase, Foxpro, SQL, Rbase, Ms Access, Oracle, dan Informix.



2. **Hardware** adalah perangkat keras dalam suatu sistem basis data, dimana mempunyai komponen-komponen utama berupa:
 - Unit pusat pengolahan (*central processing unit/CPU*).
 - Unit penyimpanan (*storage unit*).
 - *Keyboard, monitor, printer*, dan lain-lain.
3. **Operating System** merupakan perangkat lunak yang digunakan untuk mengelola aplikasi basis data dan penggunaan sumber daya komputer.
4. **Brainware** adalah manusia merupakan elemen penting pada sistem basis data. *Brainware* ini terbagi atas empat kategori yaitu:
 - Sistem *Engineer* yaitu tenaga ahli yang bertanggung jawab atas pemasangan sistem basis data dan juga mengadakan peningkatan serta melaporkan kesalahan dari sistem tersebut kepada pihak penjual.
 - *Administrator* Basis Data yaitu tenaga ahli yang mempunyai tugas untuk mengawasi sistem basis data, merencanakan dan mengaturnya.
 - *Programmer* yaitu bertugas merancang dan membuat program aplikasi yang diperlukan oleh pemakai akhir dengan menggunakan data yang terdapat dalam sistem basis data.
 - *End User* (Pengguna Akhir) yaitu tenaga ahli yang menggunakan data untuk mengambil suatu keputusan yang diperlukan dalam suatu instansi/perusahaan.

1.4 Keuntungan Sistem Basis Data

Adapun keuntungan dari sistem basis data sebagai berikut:

1. Data dapat dipakai secara bersama-sama (*multiple user*)
2. Data dapat distandardisasikan
3. Mengurangi kerangkapan data (redundansi)
4. Adanya kemandirian (kebebasan) data atau data *independent*
5. Keamanan (*security*) data terjamin
6. Keterpaduan data terjaga (masalah integritas)



1.5 Kerugian Sistem Basis Data

Adapun kerugian dari sistem basis data sebagai berikut:

1. Diperlukan tempat penyimpanan yang besar
2. Diperlukan tenaga yang terampil dalam mengelola data
3. Perangkat lunaknya mahal
4. Kerusakan pada sistem basis data dapat mempengaruhi departemen yang terkait.

1.6 Database Management System (DBMS)

Menurut Raghu Ramakrishnan dan Johannes Gehrke (2003: 3), *Database Management System (DBMS)*, adalah perangkat lunak (*software*) yang mendesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar, memberikan fasilitas untuk melakukan fungsi pengaturan, pengawasan, pengendalian, pengolahan, dan koordinasi terhadap semua proses yang terjadi pada sistem basis data. Kebutuhan terhadap sistem tersebut, termasuk juga penggunaannya, berkembang secara pesat. Alternatif penggunaan DBMS adalah untuk menyimpan data dalam *file* dan menulis kode aplikasi tertentu untuk mengaturnya.

DBMS merupakan kumpulan *file* yang saling berkaitan bersama-sama dengan program untuk pengelolaannya. *Database* adalah kumpulan datanya, sedangkan program pengelolaannya berdiri sendiri dalam satu paket program yang berfungsi untuk membaca data, mengisi data, menghapus data, serta melaporkan data dalam *database*.

1.7 Fungsi DBMS

Adapun fungsi dari DBMS sebagai berikut:

1. **Mendefinisikan (*Defining*)** maksudnya adalah *database* melibatkan penentuan tipe data, struktur, dan batasan (*constraint*) dari data yang akan disimpan dalam *database*. Definisi *database* atau informasi deskriptif tentang *database* juga disimpan dalam bentuk *database* dari *database* katalog atau kamus data, yang disebut meta-data.



2. **Membangun (*Construction*)** maksudnya adalah proses penyimpanan data pada beberapa media penyimpanan yang dikendalikan oleh DBMS.
3. **Memanipulasi (*Manipulating*)** maksudnya adalah mencakup berbagai fungsi seperti *query database* untuk mengambil data tertentu/data yang dicari, termasuk operasi *insert*, *update*, dan *delete* serta menghasilkan *report data*.
4. **Berbagi (*Sharing*)** maksudnya adalah memungkinkan beberapa pengguna dan program untuk dapat mengakses secara bersamaan.
5. **Proteksi (*Protection*)** maksudnya adalah mencakup *system protection* yang menangani kondisi (*crash*) baik pada *hardware* ataupun *software*, dan *security protection* yang menangani pengaksesan oleh *user* yang tidak memiliki hak akses terhadap *database*. Fungsi proteksi ini sangat penting untuk penggunaan *database* jangka panjang.
6. **Perawatan (*Maintain*)** maksudnya adalah DBMS harus memiliki kemampuan dalam memberikan perawatan pada sistem basis data terhadap segala perubahan terhadap yang terjadi pada data dalam *database* kapan pun diperlukan. Fungsi perawatan ini sangat penting untuk penggunaan *database* jangka panjang.

1.8 Komponen Utama DBMS

Adapun komponen-komponen utama DBMS yaitu sebagai berikut:

1. ***Query Language***
 - Digunakan oleh bagian lain dengan sedikit perintah sederhana.
 - Contoh: SQL (*Structure Query Language*), QBE (*Query By Example*).
2. ***Report Generator***
 - Dirancang untuk membuat cetakan, yang memiliki perintah-perintah untuk membuat *header*, judul, kolom, *summary*, dan lain-lain.



3. DML (*Data Manipulation Language*)

- Terdiri dari perintah-perintah yang disediakan dalam program aplikasi untuk melakukan manipulasi data seperti *append*, *list*, atau *update*.

4. DDL (*Data Definition Language*)

- Dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur tabel, dan lain-lain.
- Hasil dari kompilasi perintah DDL menjadi kamus data, yaitu data yang menjelaskan data sesungguhnya.
- Contoh: *Create*, *Modify Report*, *Modify Structure*.

5. *Recovery*

- Merupakan kemampuan untuk mengembalikan data yang rusak atau hilang akibat operasi basis data (*insert*, *update*, *delete*, dan lain-lain)

6. *Data Dictionary*

- Digunakan untuk memelihara definisi-definisi standar seluruh rinci data dalam lingkup kecil pada sistem basis data.

7. *Database*

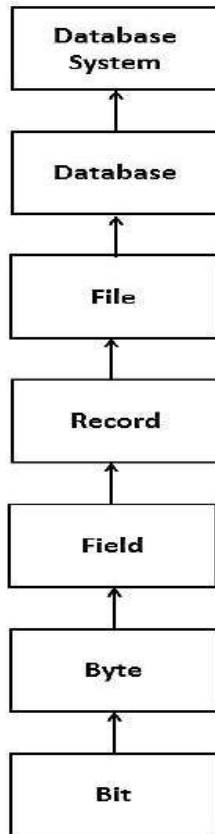
- Merupakan bagian dari DBMS yang menyediakan data dalam berbagai tipe dan format untuk memenuhi kebutuhan pemakai.

8. *Access Routine*

- Suatu rutin yang dapat dipanggil dan dipergunakan oleh program lain untuk mengakses basis data.

1.9 Penyusun Sistem Basis Data

Sistem basis data merupakan lingkup terbesar dalam organisasi data. Sistem basis data mencakup semua bentuk komponen data yang ada dalam suatu sistem. Sedangkan basis data merupakan komponen utama yang menyusun sistem basis data. Pada gambar 1.3 merupakan hierarki penyusun sistem basis data.



Gambar 1.3 Hierarki Penyusun Sistem Basis Data

Keterangan gambar 1.3 sebagai berikut:

1. **Bit** merupakan sistem angka biner yang terdiri atas angka 0 dan 1.
2. **Byte** merupakan bagian terkecil, dapat berupa karakter numerik, huruf, ataupun karakter khusus yang membentuk suatu *item* data (*field*). 1 Byte digunakan untuk mengodekan 1 karakter.
3. **Field** atau **data item** mempresentasikan suatu atribut dari suatu *record* yang menunjukkan suatu *item* dari data, misalnya nama, alamat. Kumpulan dari *field* akan membentuk suatu *record*.



4. **Record** menggambarkan suatu unit data individu tertentu. Kumpulan dari *record* akan membentuk suatu *file*.
5. **File** terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.
6. **Database** atau **basis data** merupakan sekumpulan dari berbagai macam tipe *record* yang mempunyai hubungan terhadap suatu objek tertentu.
7. **Database System** atau **sistem basis data** merupakan sekumpulan basis data yang tersusun dari beberapa *file*.

1.10 Abstraksi Data

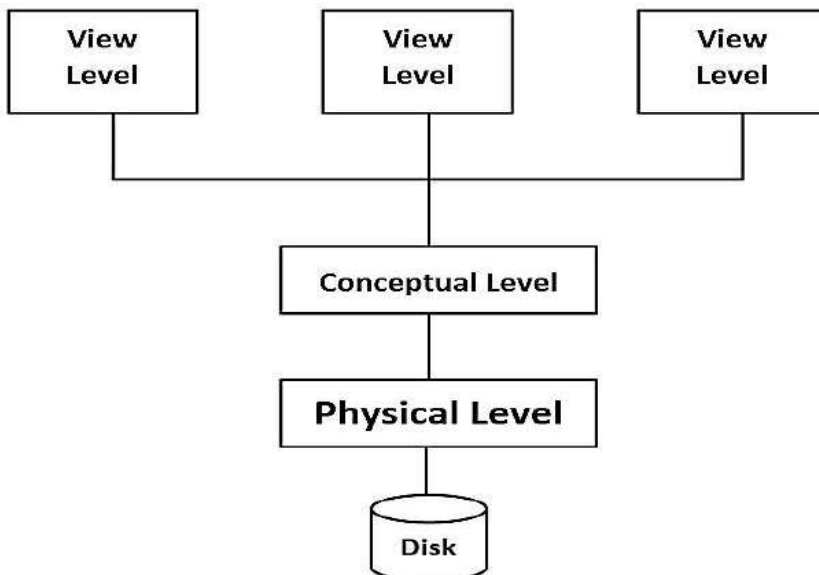
Salah satu tujuan dari DBMS untuk menyediakan fasilitas dalam melihat data kepada pemakai (*user*). Untuk itu sistem tersebut akan menyembunyikan detail tentang bagaimana data disimpan dan dipelihara. Karena itu sering kali data yang terlihat oleh pemakai sebenarnya berbeda dengan yang tersimpan secara fisik. Abstraksi data merupakan tingkatan/*level* dalam melihat bagaimana data dalam sebuah sistem basis data.

Level-level abstraksi data yaitu:

1. **Physical Level** merupakan *level* terendah dalam abstraksi data yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada *level* ini, pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri. Pemakai juga berkompeteren dalam mengetahui bagaimana representasi fisik dari penyimpanan data. Pada *level* ini, kita akan melihat data sebagai teks, angka, atau himpunan bit data.
2. **Conceptual Level** merupakan *level* berikutnya dalam abstraksi data yang menggambarkan data apa yang sebenarnya disimpan dalam basis data dan hubungannya dengan data yang lain. *Level* ini menggambarkan keseluruhan *database*. Pemakai tidak memperdulikan kerumitan dalam struktur *level physical* lagi, penggambaran cukup memakai kotak dan keterangan. *Level conceptual* ini digunakan oleh *database administrator*.



3. **View Level** merupakan *level* tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari keseluruhan basis data. Bila pada *level conceptual* data merupakan suatu kumpulan yang besar dan kompleks, pada *level* ini hanya sebagian saja yang dilihat dan dipakai. Hal ini disebabkan beberapa pemakai *database* tidak membutuhkan semua isi *database*. *Level* ini sangat dekat dengan *user*. Karena setiap *user* butuh sebagian dari *database*. Konsep dari *level* ini akan menambah pengertian mengenai kebebasan data (*independence*). Data *independence* dapat dibagi menjadi dua bagian yaitu:
- Physical Data Independence* yaitu internal skema dapat diubah oleh *database administrator* tanpa mengganggu konseptual skema.
 - Logical Data Independence* yaitu konseptual skema dapat diubah oleh *database administrator* tanpa mengganggu eksternal skema.

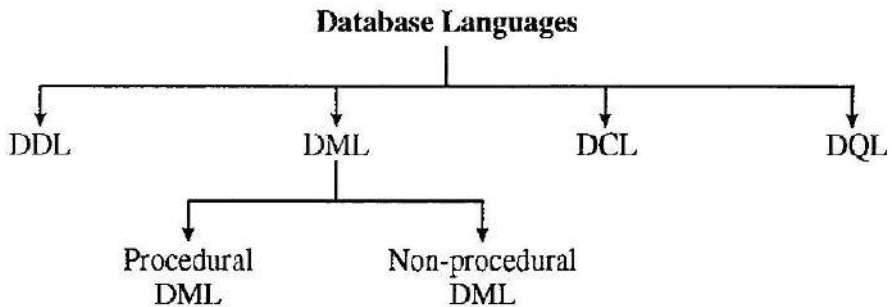


Gambar 1.4 Level Abstraksi Data Adaptasi
Raghu Ramakrishnan dan Johannes Gehrke (2003: 9)



1.11 Bahasa Basis Data

Secara normal bahasa basis data adalah SQL (*Structure Query Language*). SQL yang secara tidak resmi kini telah menjadi bahasa standar dalam *database*, karena lebih efisien dan mudah untuk dipelajari dibandingkan dengan semua bahasa *database* yang lainnya. SQL berasal dari *Sequel*, lalu didefinisikan oleh Chamberlin di IBM *Laboratory* di San Jose, California. Kemudian distandarkan oleh ANSI (*American National Standard Institute*) sebagai bahasa *Query Database Relational*. SQL ditujukan bagi pemakai yang lebih suka menyampaikan maksudnya pada penyusunan kode-kode bahasa program. Sebuah sistem basis data menyediakan beberapa jenis bahasa sebagai berikut:



Gambar 1.5 Hierarki Bahasa Basis Data

(V.K. Pallaw, 2008:20)

1. *Data Definition Language (DDL)*

DDL merupakan struktur basis data yang menggambarkan desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language (DDL)*. Dengan bahasa inilah kita dapat membuat tabel, membuat *index*, mengubah tabel dan sebagainya.

Perintahnya : ***Create Table Create Index Create View Drop Table
Alter Table Drop Index Drop View***



2. **Data Manipulation Language (DML)**

DML merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat mencakup perintah:

- a. **Select** berfungsi untuk memilih data/table pada *database*.
- b. **Insert** berfungsi untuk menyisipkan atau melakukan penambahan data baru ke *database*.
- c. **Update** berfungsi untuk mengubah data pada *database*.
- d. **Delete** berfungsi untuk menghapus data pada *database*.
- e. **Query** berfungsi untuk memanggil data yang tersimpan di dalam *database*.

Sebagai contoh untuk memilih sebuah tabel siswa pada *database*: **Select * from siswa**. DML merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data.

Ada dua jenis DML yaitu:

- a. Prosedural yaitu mensyaratkan agar pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.
- b. Non Prosedural yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa harus menyebutkan bagaimana cara mendapatkannya.

3. **Data Control Language (DCL)**

DCL merupakan bahasa yang digunakan untuk komponen dari pernyataan SQL untuk mengakses data dari *database*. Perintahnya: **Commit, Roll-Back, Save Point, Grant/Revoke, Set Transaction**.

4. **Data Query Language (DQL)**

Data Query Language adalah komponen dari SQL merupakan pernyataan yang diajukan untuk mengizinkan dalam mengambil data dari *database*, biasanya diambil dari relasi beberapa tabel. DDL dan DML bukan dua bahasa yang berbeda, keduanya ini adalah bagian dari SQL juga.

1.12 Sejarah Basis Data dan Aplikasi Praktis



Seperti juga teknologi *hardware* yang saat ini berkembang dengan sangat pesat, aplikasi-aplikasi basis data maupun sistem basis data juga berkembang. Pemrosesan data mengandalkan pertumbuhan komputer, seperti pada awal dari komputer komersial. Teknik untuk penyimpanan data dan pemrosesan berevolusi menurut waktu yang secara garis besar diuraikan dalam penjelasan sebagai berikut:

1.12.1 Tahun 1950 Hingga Awal 1960

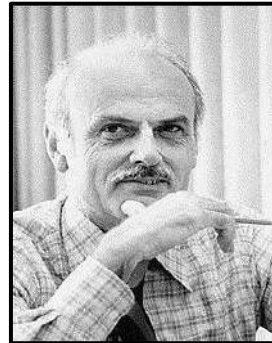
Pita magnetik dikembangkan untuk penyimpanan data. Pekerjaan pemrosesan data seperti penggajian diotomatiskan dan data disimpan di pita. Pemrosesan data mencakup pembacaan data dari satu pita atau lebih dan menulis hasilnya pada pita yang lain. Data juga dapat dimasukkan dari kartu bertebuk (*punched card*) dan keluarannya berupa cetakan di *printer*.

1.12.2 Akhir Tahun 1960 dan 1970-an

Penggunaan *hard disk* secara luas pada akhir 1960-an mengubah skenario pemrosesan data karena *hard disk* mengizinkan akses langsung ke data. Pada era ini Peter Chen (1976) dan Dr. Codd (1980) juga mendefinisikan model relasional dan cara nirprosedural untuk mencari data-data yang memenuhi kriteria tertentu (permohonan) pada model data relasional. Hasilnya, sistem basis data relasional kemudian dilahirkan dalam bentuk *Relational Database Management System* (RDBMS) dari IBM (*International Business Machine*) yang bernama DB2. Kesederhanaan model relasional serta kemampuannya untuk menyembunyikan perincian implementasi secara tuntas memudahkan kerja *programmer*.



**Gambar 1.6 Peter Chen
(Penemu Konsep Basis Data
Relasional)**



**Gambar 1.7 Dr. Edgar F. Codd
(Pencipta RDBMS yang Pertama)**

1.12.3 Tahun 1980-an

Meskipun secara akademis menarik, model relasional pada awalnya tidak digunakan secara praktis karena kinerjanya yang kurang memuaskan, basis data relasional berkinerja kurang baik dibandingkan dengan basis data hierarki serta jaringan. Keadaan ini berubah dengan kehadiran System R (suatu proyek penelitian di IBM *Research*) yang mengembangkan teknik-teknik baru untuk mengonstruksi basis data relasional yang lebih efisien. Kerja pendahuluan dari System R dilakukan oleh Astrahan dan kawan-kawan (1976), dan Chamberlin dan kawan-kawan (1981). Prototipe fungsional System R menghasilkan produk basis data relasional yang pertama, SQL/DS. Sistem basis data relasional komersial seperti IBM DB2, Oracle, Ingres, serta DEC Rdb memainkan peran penting dalam perkembangan teknik-teknik penting pada pemrosesan permohonan deklaratif. Dengan demikian, pada awal 1980-an basis data relasional dalam hal kinerjanya menjadi cukup kompetitif dibandingkan dengan sistem hierarki dan jaringan. Sejak tahun 1980-an model data relasional mulai mendominasi kancah pemrosesan data.



1.12.4 Awal Tahun 1990-An

Bahasa SQL dan PL/SQL (*Programming Language/Structured Query Language*) pertama kali dirancang oleh para peneliti dari Oracle untuk aplikasi-aplikasi pengambilan keputusan yang sangat sarat dengan operasi permohonan (pencarian data-data yang memenuhi kriteria-kriteria tertentu). Banyak *vendor* basis data memperkenalkan produk-produk basis data paralel dalam periode ini. *Vendor-vendor* basis data juga mulai memperkenalkan sistem objek relasional untuk dapat beroperasi pada tipe-tipe data yang lebih kompleks (misalnya: data-data gambar, suara, serta video) serta untuk mengadaptasi produknya dengan konsep pemrograman yang mulai marak di era ini, yaitu pemrograman berorientasi objek (*Object Oriented Programming/OOP*).

1.12.5 Akhir tahun 1990-an

Pemicunya adalah perkembangan yang luar biasa dari *World Wide Web* (WWW). Sistem basis data disebarakan secara lebih ekstensif dibandingkan dengan masa-masa sebelumnya. Sistem basis data saat ini harus mendukung tingkat pemrosesan transaksi yang sangat tinggi serta harus dijamin keandalannya (harus dapat mendukung transaksi 7 x 24 jam, tanpa sedikit pun mengurangi kinerjanya).

1.12.6 Tahun 2000 dan seterusnya

Secara alamiah kita dapat meramalkan arah teknologi basis data pada dekade ini. Jika tidak terjadi banyak kejutan, kita dapat mengharapkan kecenderungan yang saat ini berjalan akan terus berlanjut. Kemampuan-kemampuan yang kita harapkan adalah:

1. Kemampuan untuk mengelola tipe data yang semakin kompleks.
2. Perkembangan lebih lanjut dari DBMS bertipe objek relasional.
3. Basis data yang terdistribusi secara murni.
4. Tempat penyimpanan yang dapat dialamatkan dengan isinya akan semakin populer.



5. Basis data dan teknologi lain yang berkaitan, misalnya untuk aplikasi kecerdasan buatan (*Artificial Intelligence*) serta layanan informasi.

1.13 Perkembangan Basis Data

Sejak awal 1990-an, di saat komputer mikro berjaya dengan teknologi komputer jaringan (LAN) yang diperkenalkan oleh Novell, pemanfaatan Dbase III+ semakin luas dan diperkaya dengan perangkat lunak aplikasi untuk basis data semacam Clipper dan Foxpro. Sedangkan untuk komputer yang lebih tinggi sudah mulai lahir produk-produk DBMS, seperti IBM-DB2, SDK-Ingres, dan Oracle.

1.13.1 Kategori DBMS

Berdasarkan perkembangan teknologi, kita dapat mengategorikan DBMS dalam beberapa kategori yang jumlahnya bisa bertambah seiring dengan perkembangan teknologi di masa yang akan datang. Beberapa kategori DBMS itu sebagai berikut:

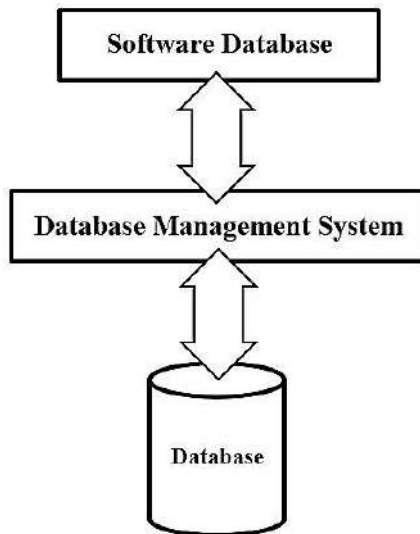
1. **DBMS Konvensional** sebagian besar DBMS yang ada saat ini masih dapat dikategorikan sebagai DBMS konvensional yang sudah menjadi standar pemakaian di berbagai tempat. DBMS konvensional ini meliputi juga DBMS untuk pemakai tunggal maupun banyak pemakai tetapi dengan menerapkan model basis data yang lama, misalnya model jaringan, model hierarki, atau pun model yang lebih populer yaitu model relasional.
2. **DBMS Berorientasi Objek** merupakan respons terhadap perkembangan yang terjadi dalam dunia pemrograman, jadi *database*-nya lebih mengarah untuk bahasa pemrograman.
3. **DBMS Objek Relational** merupakan DBMS yang kompromi antara DBMS *relational* dengan DBMS berorientasi objek. Secara objektif DBMS ini masih menerapkan model *relational* sebagai basis pengelolaan datanya.
4. **DBMS untuk Web/Internet** merupakan DBMS yang dibuat untuk keperluan khusus yaitu menangani dokumen-dokumen *web* yang banyak digunakan di dunia *internet*.



1.13.2 DBMS dan Aplikasi

Tidak sedikit orang berpendapat bahwa perangkat lunak semacam Dbase III+, Foxpro, dan MS Access tidak dapat dikategorikan sebagai DBMS, yang berpendapat seperti ini pasti orang yang beranggapan bahwa sebuah DBMS haruslah sebuah sistem (perangkat lunak) yang berdiri sendiri dan terpisah dengan aplikasi basis data. DBMS yang besar, seperti MS SQL Server, Oracle, Sybase, Informix, IBM-DB2 memang dirancang sejak awal untuk berdiri sendiri, sehingga banyak sekali fasilitas yang ditandatangani oleh DBMS ini.

Basis data dapat dianggap sebagai objek yang hidup, karena sesungguhnya aplikasi tidak pernah berinteraksi langsung dengan basis data. Tetapi selalu melalui DBMS sebagai perantara. Bahkan DBMS sendiri bisa melakukan aktivitas sendiri yang bisa ditangkap oleh aplikasi. Jadi interaksi antara aplikasi, basis data, dan DBMS merupakan interaksi dua arah.



Gambar 1.8 Aplikasi *Database*



Perangkat lunak semacam Dbase III+, Foxbase, MS Access dapat saja kita golongkan sebagai DBMS, jika kita lebih meninjau dari sisi fungsinya secara umum, yaitu untuk mengelola basis data, walaupun memang fasilitas tambahannya yang tersedia tidak selengkap yang ada pada DBMS yang besar. Aplikasi basis data yang kita buat akan menyatu dengan DBMS pada saat pemakaiannya.

1.13.3 Arsitektur Sistem

Arsitektur sistem merujuk pada konfigurasi secara keseluruhan yang akan menjadi tempat hidup dari DBMS, basis data, dan aplikasi yang memanfaatkannya. Dari waktu ke waktu sejalan dengan kemajuan teknologi, arsitektur sistem yang akan kita pilih semakin beragam dan semakin banyak variannya, yang juga berubah adalah keunggulannya. Arsitektur sistem yang saat ini boleh dikatakan paling baik dan banyak diterapkan di berbagai tempat.

Pertimbangan utama dalam menentukan arsitektur sistem yang paling cocok tentu saja bukan hanya keunggulan teknologinya, di samping faktor biaya juga harus memilih arsitektur sistem yang memang sesuai dengan kebutuhan nyata di tempat sistem yang akan menggunakannya.

Beberapa jenis arsitektur sistem yang dapat digunakan adalah:

1. Sistem tunggal (*stand alone*)
2. Sistem tersentralisasi (*centralized system*)
3. Sistem *client-server*

1.13.4 Perkembangan DBMS Berorientasi Kepada Pemakai

Berdasarkan orientasi pemakainya, kita dapat mengelompokkan DBMS dalam 2 kategori sebagai berikut:

1. DBMS yang Berorientasi untuk Satu Pemakai

MS Access, Dbase, Clipper, Foxpro, Foxbase, Borland, merupakan contoh-contoh DBMS yang lebih berorientasi untuk satu pemakai dan karena itu dapat dengan mudah dipasang di komputer pribadi (PC). Pada DBMS kelompok ini pengembangannya terjadi pada aspek-aspek sebagai berikut:



- Jika pada awalnya, struktur tabel hanya mencakup pendefinisian nama *field*, tipe, dan ukurannya. DBMS yang lebih baru juga memasukkan *feature*, deskripsi *field*, dan bentuk validasi sebagai bagian dari struktur tabel.
- Tipe data yang dapat ditangani oleh DBMS terbaru sudah semakin banyak, seperti untuk mengakomodasikan kebutuhan penyimpanan data teks yang panjang, gambar, dan data OLE.
- Adanya utilitas tambahan untuk mempermudah pemakai dalam menggunakan DBMS, seperti pembuatan *query*, pembuatan laporan, hingga pembuatan perintah *macro* secara otomatis melalui pendefinisian menu dan tampilan layar.

2. DBMS yang Berorientasi untuk Banyak Pemakai

Oracle, Borlan, Interbase, MS SQL Server, Sybase, Informix, IBM DB2 merupakan contoh-contoh DBMS yang lebih diorientasikan untuk banyak pemakai dan karena itu lebih ditujukan untuk pemakaian pada sistem jaringan komputer. Fungsi pengelolaan DBMS kelompok ini menangani aspek-aspek sebagai berikut:

- Pengamanan objek basis data terhadap pemakai yang tidak berhak (aspek *security*) dan bentuk-bentuk operasi yang tidak diperbolehkan (aspek *integrity*).
- Penanganan pemulihan data akibat kegagalan operasi basis data (aspek *recovery*), baik yang disebabkan oleh operasi-operasi basis data yang salah atau menimbulkan konflik, maupun faktor eksternal seperti mesin yang macet (*crash*), *disk* yang rusak atau terputusnya koneksi jaringan.

Latihan Teori

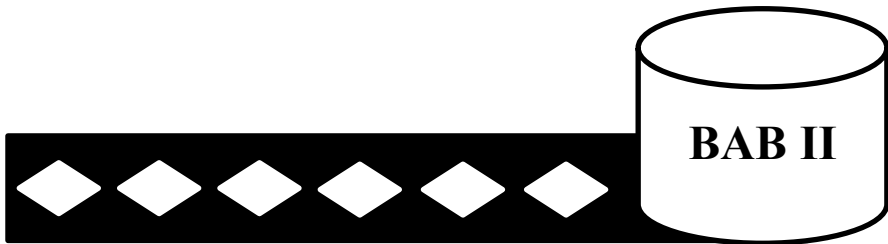
Jawablah pertanyaan di bawah ini dengan benar dan tepat!

1. Jelaskan secara ringkas sejarah basis data dan aplikasi praktis!
2. Sebutkan minimal 5 buah contoh aplikasi *database* tradisional!
3. Jelaskan pengertian dan berikan contoh dari:
 - a. Sistem tunggal (*Stand Alone*) dalam *database*



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

- b. Sistem tersentralisasi (*Centralized System*) dalam *database*
 - c. Sistem *client-server* dalam *database*
4. Jelaskan apa yang dimaksud dengan:
- a. *Data security*
 - b. *Data integrity*
 - c. *Data recovery*
 - d. *Data redundancy*
 - e. *Data independent*



PENGANTAR BASIS DATA LANJUTAN I



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu mengetahui dan memahami perancangan basis data
2. Mahasiswa mampu mengetahui dan memahami model data
3. Mahasiswa mampu mengetahui, memahami, dan membuat *Entity Relationship Diagram* (ERD)



2.1 Perancangan Basis Data

Perancangan basis data diharapkan telah memiliki gambaran serta pemahaman yang jelas tentang tempat penyimpanan data serta yang sangat penting adalah pemahaman tentang kebutuhan pengguna. Bagaimanapun, juga pemodelan data yang dikembangkan sepanjang tahap analisis secara eksplisit mengabaikan permasalahan yang berkaitan dengan teknologi-teknologi basis data, yang mulai pada tahap perancangan ini harus dipertimbangkan, sebelum pengimplementasian basis data. Sebelum dapat mengimplementasikan basis data, model konseptual harus dipetakan ke model data yang kompatibel (bersesuaian) dengan DBMS yang akan digunakan.

Aktivitas perancangan basis data akan mentransformasikan spesifikasi kebutuhan untuk tempat penyimpanan data yang dikembangkan selama analisis basis data ke dalam spesifikasi terstruktur untuk memandu implementasi langsung ke dalam sistem basis data yang digunakan. Ada dua bentuk spesifikasi, yaitu sebagai berikut:

1. **Perancangan basis data secara logika**, yang memetakan kebutuhan konseptual ke model data yang berasosiasi dengan sistem basis data yang spesifik.
2. **Perancangan basis data secara fisik**, yang mengindikasikan semua parameter untuk penyimpanan data yang kemudian menjadi masukan pada saat akan dilakukan implementasi basis data, di mana basis data secara aktual didefinisikan dengan *Data Definition Language* (DDL).

Perancangan basis data sangat diperlukan, agar kita bisa memiliki basis data yang baik dan efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam melakukan manipulasi (penambahan/*insert*, perubahan/*update*, dan penghapusan/*delete* data). Dalam merancang basis data, kita dapat melakukan dengan dua cara yaitu:

1. Menerapkan teknik normalisasi terhadap struktur tabel yang ada.
2. Membuat model *entity relationship*.

Perancangan basis data akan disebut **buruk** jika terjadi hal sebagai berikut:



1. Data yang sama tersimpan di beberapa tempat (*file* atau *record*).
2. Ketidakmampuan untuk menghasilkan informasi tertentu yang diperlukan oleh pemakai.
3. Terjadi kehilangan informasi.
4. Terjadinya redundansi (pengulangan) atau duplikasi (*duplicate*) data sehingga menimbulkan pemborosan terhadap ruang penyimpanan dan menyulitkan saat proses pembaharuan data.
5. Timbul adanya *Null Value*.

2.1.1 Perancangan Basis Data secara Logika

Perancangan basis data secara logika akan ditekankan pada model data relasional, bukan pada model berorientasi objek. Perancangan basis data secara logika adalah proses untuk mentransformasikan model data konseptual ke model data logika. Meskipun banyak model data yang lain, kita mempunyai dua alasan untuk memberikan penekanan lebih pada model data relasional. Pertama, model data relasional sekarang ini adalah model yang paling umum digunakan di aplikasi basis data kontemporer. Kedua, beberapa prinsip perancangan basis data secara logika dapat diaplikasikan ke model logika yang lain dengan baik. Sebagai contoh, basis data yang dirancang dengan *Unified Modeling Language* (UML) dapat secara serupa dipetakan ke sistem basis data relasional dengan kaidah-kaidah yang kita tangkap dari pemetaan *Entity Relationship Diagram* (ERD). Dengan demikian, buku ini tidak akan membahas tentang perancangan basis data untuk model-model UML, tetapi dengan model relasional ERD.

Sasaran perancangan basis data secara logika sesungguhnya adalah untuk menerjemahkan perancangan konseptual (yang mencerminkan kebutuhan data pada organisasi yang kita analisis) ke perancangan basis data logika yang dapat langsung diimplementasikan pada sistem basis data yang dipilih. Basis data yang dihasilkan harus memenuhi kebutuhan pengguna dalam hal berbagi data, fleksibilitas, serta kemudahan akses.



2.1.2 Perancangan Basis Data secara Fisik

Maksud perancangan basis data secara fisik ini adalah menerjemahkan deskripsi logis data ke spesifikasi teknik untuk menyimpan (*store*) serta memanggil (*retrieval*) data. Sasarannya adalah menciptakan perancangan untuk penyimpanan data yang menyediakan kinerja yang baik dan memastikan integritas, keamanan, serta kemampuan untuk dipulihkan (jika terjadi hal-hal yang merusak basis data). Perancangan basis data tidak cukup bagaimana mengimplementasikan berkas-berkas (*file*) dan basis data (misalnya, menciptakan serta memasukkan data-data ke dalamnya). Perancangan basis data secara fisik menghasilkan spesifikasi teknis sehingga pemrogram dan pengguna lain yang terlibat dalam pengembangan dapat menggunakannya sebagai panduan selama tahapan implementasi.

Sasaran perancangan basis data secara fisik sesungguhnya adalah meningkatkan efisiensi dalam pemrosesan data, dimana akan berfokus pada bagaimana pemrosesan basis data berjalan dengan efisien dan efektif, serta meminimalkan efisiensi penggunaan ruang penyimpanan.

Perancangan basis data secara fisik membutuhkan beberapa pilihan kritis yang akan berimbas pada integritas dan kinerja aplikasi. Kunci-kunci untuk melakukan pilihan-pilihan kritis mencakup sebagai berikut:

1. Pemilihan format/bentuk/ukuran penyimpanan (dinamakan tipe data) untuk tiap atribut dari model data logika. Format dipilih untuk meminimalkan dan mengoptimalkan penggunaan ruang fisik dan untuk memaksimalkan integritas data.
2. Pengelompokan atribut-atribut dari model data logika ke rekaman fisik. Meskipun kolom-kolom pada tabel relasional merupakan definisi yang alami untuk isi rekaman fisik, hal itu tidak selalu merupakan pengelompokan atribut yang dikehendaki dalam kaitannya dengan kinerja sistem basis data.
3. Perancangan rekaman-rekaman pada memori sekunder (terutama *hard disk*) sehingga rekaman-rekaman secara individual maupun kelompok rekaman (dinamakan pengorganisasian berkas) dapat disimpan, dipanggil kembali, serta diperbaharui dengan cepat. Pertimbangan-pertimbangan juga harus dilakukan untuk

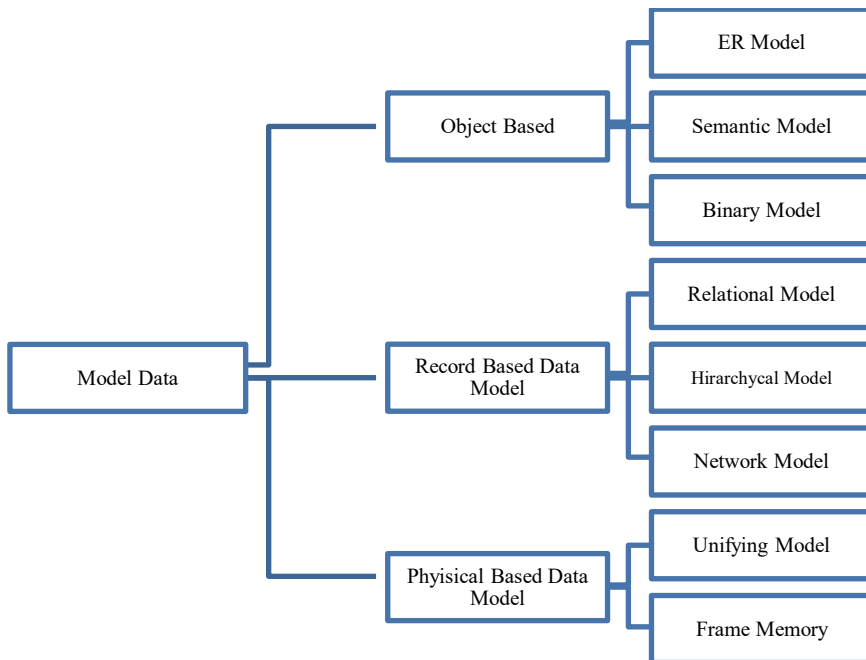


melindungi data serta untuk melakukan *recovery* saat terjadi kesalahan-kesalahan serta perusakan-perusakan pada basis data.

4. Pemilihan struktur (dinamakan indeks dan arsitektur basis data) untuk menyimpan dan menghubungkan berkas-berkas sehingga pemanggilan data berlangsung dengan cara yang efisien.
5. Penyiapan strategi-strategi untuk menangani permohonan (*query*) pada basis data dengan tujuan untuk mengoptimalkan kinerja basis data dalam menangani permohonan. Struktur basis data yang efisien hanya bermanfaat jika sistem basis data yang menangani permohonan dirancang secara cerdas.

2.2 Model Data

Ada beberapa model data di dalam suatu sistem basis data. Model data adalah pemetaan yang menunjukkan hubungan logika antara elemen data atau suatu cara untuk menjelaskan bagaimana pemakai dapat melihat data secara logik. Dalam hal ini pemakai tidak perlu memperhatikan bagaimana data disimpan dalam media penyimpanan secara fisik. Penggolongan model data dapat ditunjukkan dalam gambar 2.1 di bawah ini.



Gambar 2.1 Penggolongan Model Data

Yang akan dibahas pada bab ini adalah model data berbasis *record* dan satu buah model data berbasis objek terdiri dari:

1. Model data *hirarchycal*
2. Model data *network*
3. Model data *relational*
4. Model ER

2.3 Model Data Hierarki (*Hirarchycal*)

Sebuah basis data dengan model data hierarki terdiri atas sekumpulan *record* yang dihubungkan satu sama lain melalui *link* (berupa *pointer*) yang membentuk suatu struktur hierarki. Model data hierarki disebut juga model struktur pohon.

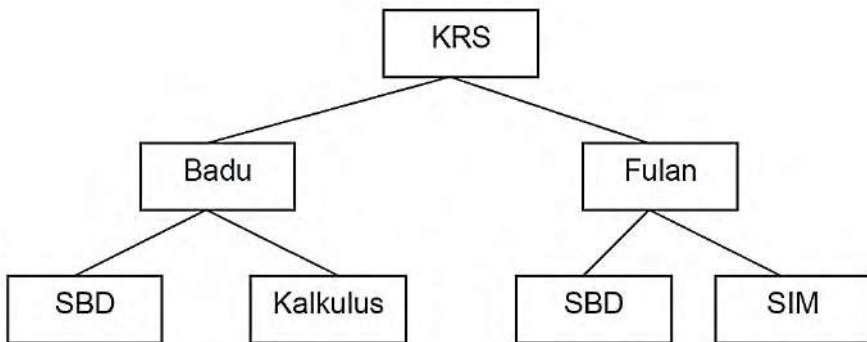
Ketentuan-ketentuan dalam model data hierarki adalah sebagai berikut:



1. Terdiri dari kumpulan *record* (R1, R2, R3,... Rn) yang masing-masing mempunyai *field* pengenalan.
2. Terdapat suatu kumpulan kaitan yang menghubungkan semua jenis *record* sehingga membuat diagram struktur data.
3. Kaitan tersebut membentuk suatu pohon yang semua ujungnya mengarah ke daun.
4. Tak mungkin ada elemen yang mempunyai *parent* lebih dari satu.
5. Setiap kaitannya membentuk hubungan tunggal-jamak.

Model Hierarki mempunyai dua konsep struktur data yaitu:

1. *Record* adalah kumpulan nilai-nilai *field* yang memberikan informasi pada suatu entitas atau *relationship*. *Record-record* yang sejenis dikelompokkan dalam *record type*. *Record type* diberikan nama dan strukturnya didefinisikan oleh kumpulan nama-nama *field* dimana masing-masing *field* mempunyai jenis data tertentu.
2. *Parent Child Relationship (PCR) Type* adalah relasi 1 to N antara dua *record type* dan di sisi lain (N) disebut *child record type* dari PCR *type*. PCR *type* terdiri dari satu *record* dari *parent record type* dan beberapa *record* dari *child record type*.



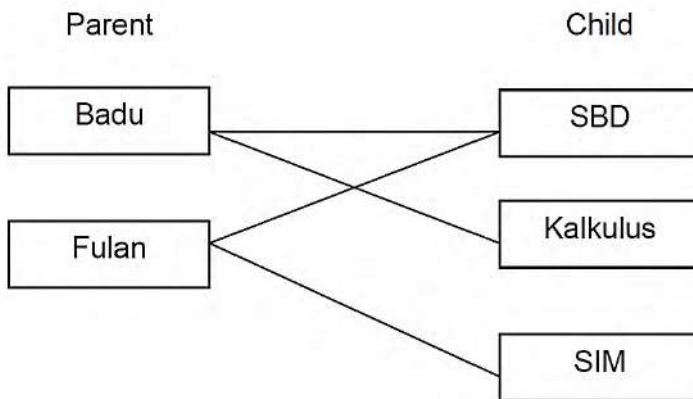
Gambar 2.2 Model Data Hierarki



2.4 Model Data Jaringan (*Network*)

Sebuah basis data dengan model jaringan akan terdiri atas sekumpulan *record* yang dihubungkan satu sama lain melalui *link* (yang berupa *pointer*). Setiap *record* adalah sekumpulan *field* yang masing-masing hanya berisi sebuah nilai data. Sebuah *link* merupakan sebuah penghubung di antara dua *record*.

Model jaringan dikenal sebagai struktur plex. Pada dasarnya struktur jaringan ini merupakan perluasan dari struktur hierarki. Kalau pada struktur hierarki, setiap *child* hanya mempunyai satu *parent*, sedangkan pada struktur jaringan setiap *child* dapat mempunyai lebih dari satu *parent*. Struktur jaringan merupakan suatu *graph*, terdiri dari suatu *node* (simpul), yang dihubungkan dengan suatu *edge*.



Gambar 2.3 Model Data Jaringan

2.5 Model Data Relasional (*Relational*)

Model data relasional ini diperkenalkan pada tahun 1970-an oleh Peter Chen dan Dr. E. F. Codd, kemudian dilanjutkan oleh IBM (Codd, 1970). Dua proyek penelitian telah dikembangkan untuk membuktikan keandalan model relasional dan untuk mengembangkan sistem prototipe. Yang pertama, di laboratorium IBM (*International Business Machine*), di



San Jose Research Laboratory, telah dikembangkan *System-R* (prototipe DBMS bertipe relasional atau RDBMS [*Relational Database Management System*]), selama akhir tahun 1970-an. Yang kedua, di University of California di Berkeley telah dikembangkan Ingres, sebuah DBMS bertipe relasional yang sangat berorientasi akademis.

Produk-produk RDBMS untuk pemanfaatan bisnis dari berbagai *vendor* mulai muncul pada tahun 1980-an. Sekarang ini RDBMS menjadi teknologi yang dominan untuk sistem basis data, tercatat ratusan produk RDBMS untuk PC hingga *mainframe*, dengan MySQL, Oracle, serta Microsoft SQL Server yang menjadi produk paling sering digunakan di Indonesia. Ketiganya sesungguhnya memiliki perintah SQL yang cukup mirip. Meskipun demikian, pemilihan penggunaannya harus disesuaikan dengan seberapa besar jumlah data (serta tipenya) sesuai dengan kebutuhan organisasi/perusahaan yang akan menggunakannya.

Model data relasional menggambarkan data dalam bentuk tabel-tabel. Model data relasional berbasis teori matematika (kalkulus relasional), yang merupakan dasar teori yang sangat kuat. Meskipun demikian, kita hanya memerlukan konsep yang sederhana untuk mendeskripsikan model relasional, yang relatif sederhana untuk dipahami dan dapat digunakan oleh mereka yang tidak akrab dengan teori yang mendasarinya, yang lebih bermanfaat saat kita melakukan pengembangan sistem informasi. Model data relasional sesungguhnya mengandung tiga komponen utama, yaitu sebagai berikut:

1. Struktur data, yakni data-data diorganisasikan dalam bentuk tabel dengan baris-baris dan kolom-kolom.
2. Manipulasi data, yakni operasi yang sangat berdaya guna (dengan SQL) digunakan untuk memanipulasi data-data yang disimpan di setiap relasi-relasi.
3. Integritas data, yakni fasilitas-fasilitas untuk mespesifikasi aturan bisnis yang memelihara integritas data saat mereka dimanipulasi.

Model data relasional digambarkan dalam bentuk tabel dua dimensi. Untuk memberikan gambaran sebuah berkas data. Kolom dari tabel relasional menunjukkan antara *field* dari suatu *record*. Sedangkan



baris menunjukkan hubungan antara *record* dalam suatu berkas data. Relasi atau keterhubungan antara satu tabel dengan tabel yang lainnya. Setiap relasi memuat himpunan kolom serta himpunan baris. Adapun sifat dari relasi yaitu sebagai berikut:

1. Tidak ada baris/*record* yang ganda
2. Baris/*record* boleh tidak berurutan
3. Kolom/*field* tidak perlu diurut
4. Semua nilai kolom/*field* adalah atomik, artinya kolom/*field* yang satu berbeda dengan yang lain.

Adapun karakteristik model data relasional yaitu sebagai berikut:

1. Semua elemen data pada suatu baris dan kolom tertentu harus mempunyai nilai tunggal.
2. Semua elemen data pada suatu kolom tertentu dalam relasi yang sama harus mempunyai jenis yang sama.
3. Masing-masing kolom dalam suatu relasi mempunyai suatu nama *field* yang unik

Contoh sebuah basis data model relasional yang terdiri dari:

1. Tabel Supplier (tabel acuan)

No_Penyuplai	Nama_Penyuplai	Kota
S1	Agung	Jakarta
S2	Budi	Bandung
S3	Laksana	Surabaya
S4	Wahyu	Bandung
S5	Zaki	Jakarta

2. Tabel Barang (tabel acuan)

Kode_Barang	Nama_Barang	Harga
B01	Monitor	800.000
B02	CPU	2.000.000
B03	Mouse	50.000
B04	Printer	1.200.000
B05	Disket	30.000



3. Tabel Transaksi (tabel relasi)

No_Penyuplai	Kode_Penyuplai	Jumlah
S1	B02	30
S1	B03	23
S1	B04	60
S2	B02	65
S2	B05	54
S3	B02	34
S4	B03	25
S4	B01	12
S5	B01	18
S5	B04	20

Tabel di atas dianggap sebagai suatu relasi. Baris dari tabel biasanya disebut *record*, sedangkan kolom disebut *field*.

Keterangan:

- **Tabel Penyuplai** terdiri dari *field*:
 1. **No_Penyuplai** (sebagai kunci utama elemen data pada tabel)
 2. **Nama_Peyuplai**
 3. **Kota**
- **Tabel Barang** terdiri dari *field*:
 1. **Kode_Barang** (sebagai kunci utama elemen data pada tabel)
 2. **Nama_Barang**
 3. **Harga**
- **Tabel Transaksi** terdiri dari *field*:
 1. **No_Penyuplai** (sebagai kunci tamu elemen data pada tabel)
 2. **Kode_Barang** (sebagai kunci tamu elemen data pada tabel)
 3. **Jumlah**

2.6 Entity Relationship Diagram (ERD)

Rosa A. S. dan M. Shalahuddin (2013:50-53) pada pertengahan tahun 1970-an basis data mulai dipergunakan. Berbagai riset dilakukan untuk menemukan cara pemodelan data yang dapat memberikan cetakan (*template*) perancangan basis data. Pemodelan awal basis data paling banyak digunakan adalah dengan *Entity Relationship Diagram* (ERD).



ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan *Object Oriented Database Management System* (OODBMS) maka perancangan basis data tidak perlu menggunakan ERD.

ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Haryy Ellis), Crow's Foot, dan beberapa notasi lainnya. Namun yang banyak digunakan adalah notasi dari Peter Chen. Peter Chen (1976) mengajukan ERD yang merupakan *tool* analisis sistem pertama yang memusatkan pada data dan keterkaitan antar data serta pengorganisasian data. Tujuan ERD adalah untuk menunjukkan objek data dan *relationship* yang ada pada objek tersebut. Di samping itu ERD ini merupakan salah satu alat untuk perancangan dalam basis data.

2.6.1 Simbol-Simbol ERD

Pada buku ini, hanya membahas tentang simbol-simbol ERD menggunakan notasi dari Peter Chen, yaitu sebagai berikut:

1. Entitas (*Entity*)

Brown (2002) menyatakan sebuah **entitas** (*entity*) adalah suatu benda yang eksistensinya terpisah dan berbeda. Entitas merupakan data inti yang akan disimpan, sebagai bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel. Entitas merupakan suatu objek yang dapat dibedakan atau dapat diidentifikasi secara unik dengan objek yang lainnya, dimana semua informasi yang berkaitan dengannya dikumpulkan. Kumpulan dari entitas yang sejenis dinamakan *entity set*.

Contoh entitas:

Proyek	Penjualan	Langganan	Kendaraan
Peralatan	Pegawai	Pasien	Mahasiswa

Simbol entitas:

nama_entitas



2. Relasi (*Relationship*)

Relasi (*relationship*) adalah hubungan yang terjadi antara satu entitas dengan entitas yang lainnya. Relasi tidak mempunyai keberadaan fisik atau konseptual kecuali yang diwarisi dari hubungan antara entitas tersebut. Kumpulan relasi yang sejenis disebut dengan *relationship diagram*.

Simbol relasi:



Contoh:



Keterangan:

Mengambil adalah *relationship set* yang terbentuk antara entitas Mahasiswa dengan entitas Mata Kuliah.

3. Atribut

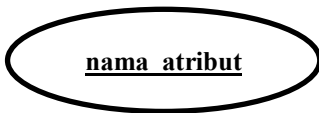
Atribut adalah karakteristik dari entitas atau dari relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Atribut sebagai *field* atau kolom data yang perlu disimpan dalam suatu entitas. Atribut merupakan bagian yang dimiliki oleh entitas.

Simbol atribut:

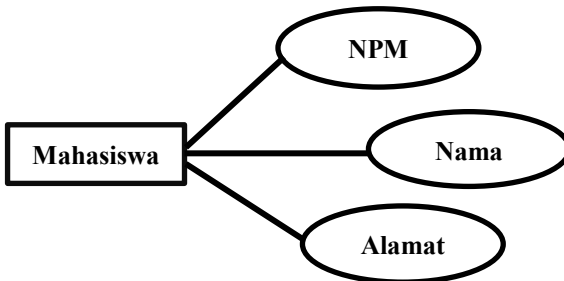




Atribut kunci primer sebagai *field* atau kolom data yang perlu disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* (*Primary Key*) yang diinginkan biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama). Simbol atribut sebagai kunci primer ditandai dengan diberikannya garis bawah pada nama atribut:



Contoh:



4. Penghubung (*line/connector*)

Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki kemungkinan jumlah pemakaian. Kemungkinan jumlah pemakaian maksimum keterhubungan antara entitas satu dengan entitas yang lainnya disebut dengan kardinalitas. Misalkan ada kardinalitas 1: M atau sering disebut *one to many* menghubungkan entitas A dengan entitas B.

Simbol penghubung: 1 _____ M

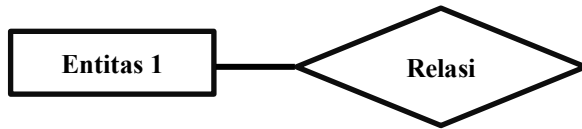


2.6.2 Derajat Relationship

Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu buah relasi menghubungkan banyak entitas), namun banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

1. **Unary (Derajat Satu)** adalah satu buah relasi menghubungkan satu buah entitas.

Contoh:



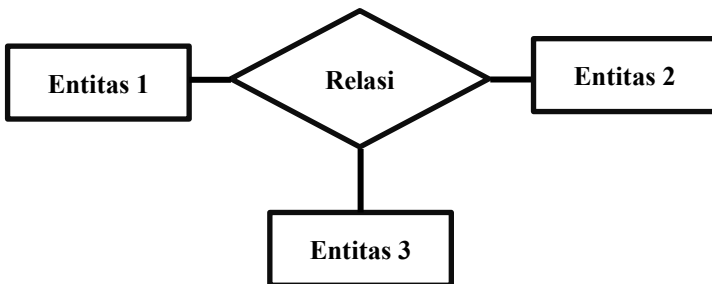
2. **Binary (Derajat Dua)** adalah satu buah relasi menghubungkan dua buah entitas.

Contoh:



3. **Ternary (Derajat Tiga)** adalah satu buah relasi menghubungkan tiga buah entitas.

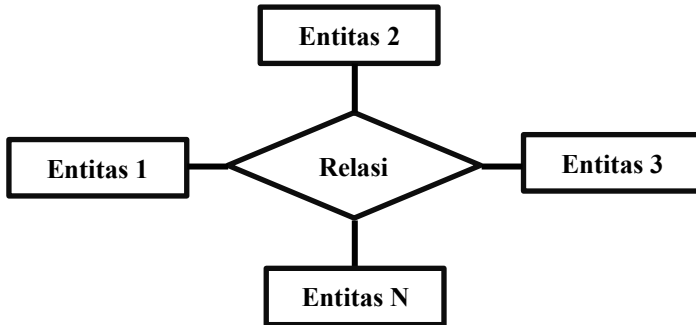
Contoh:





4. **N-ary (Derajat Lebih)** adalah satu buah relasi menghubungkan banyak entitas.

Contoh:



2.6.3 Cardinality Ratio

Yaitu menjelaskan batasan pada jumlah entitas yang berhubungan melalui suatu relasi. Adapun jenis-jenis *Cardinality Ratio* sebagai berikut:

1. **One To One (1: 1)**

Yaitu perbandingan antara entitas pertama dengan entitas kedua berbanding satu berbanding satu.

Contoh:



2. **One To Many (1: M)**

Yaitu perbandingan antara entitas pertama dengan entitas kedua berbanding satu berbanding banyak.

Contoh:





3. **Many To One (M: 1)**

Yaitu perbandingan antara entitas pertama dengan entitas kedua berbanding banyak berbanding satu.

Contoh:



4. **Many To Many (M: M)**

Yaitu perbandingan antara entitas pertama dengan entitas kedua berbanding banyak berbanding banyak, jarang terjadi untuk hubungan M: M ini.

Contoh:



2.6.4 **Langkah-Langkah Membuat ERD**

Adapun langkah-langkah membuat ERD sebagai berikut:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut kunci sebagai *primary key* dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta *foreign key*.
4. Menentukan derajat dan *cardinality ratio* relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci)



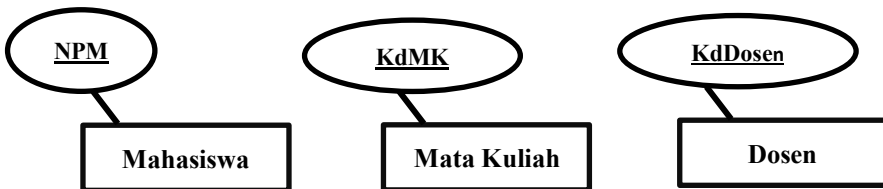
2.6.5 Contoh Kasus

Dalam materi ERD ini, kita mengambil kasus pada akademik sebagai berikut:

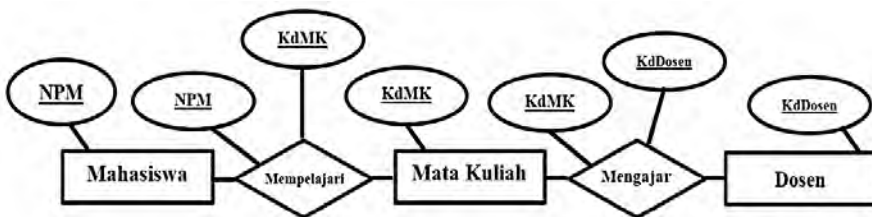
1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.



2. Menentukan atribut-atribut kunci sebagai *primary key* dari masing-masing himpunan entitas.

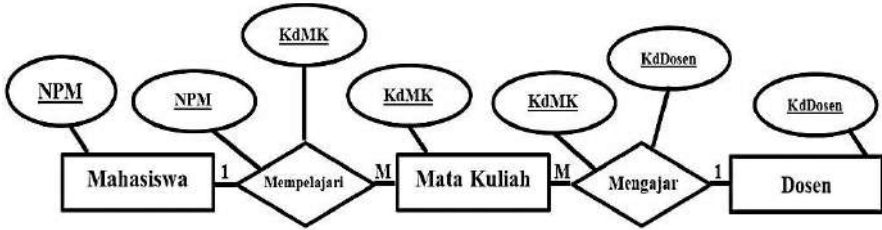


3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta *foreign key*.

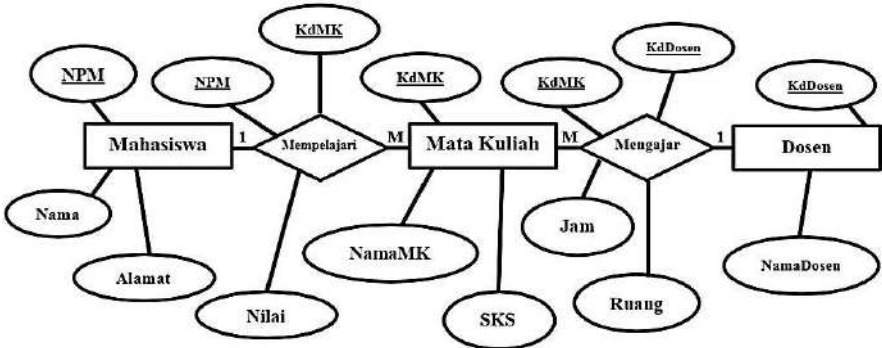




4. Menentukan derajat relasi dan *cardinality ratio* untuk setiap himpunan relasi.



5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci).



2.6.6 ERD dengan Kamus Data

- Pada sistem yang ruang lingkungannya lebar dan kompleks, penggambaran atribut-atribut dalam ERD seringkali malah mengganggu tujuan yang ingin dicapai. Oleh karena itu dapat dinyatakan dalam **Kamus Data (Data Dictionary)**.
- Kamus data berisi daftar atribut yang diapit tanda { dan }.
- Atribut yang merupakan *primary key/foreign key* digarisbawahi. Contoh:



- Adapun untuk kamus data dari kasus akademik sebagai berikut:
 - Mahasiswa = {NPM, Nama, Alamat}
 - Mata Kuliah = {KdMK, NamaMK, SKS}
 - Dosen = {KdDosen, NamaDosen}
 - Mempelajari = {NPM, KdMK, Nilai}
 - Mengajar = {KdMK, KdDosen, Jam, Ruang}

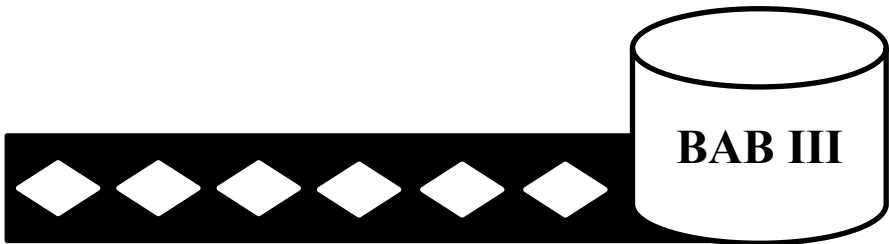
Latihan Praktik

Praktikkanlah soal di bawah ini dengan tepat!

1. Pastikan PC/Laptop Anda telah terinstal EdrawMax/PowerDesigner sebagai *software* pengolah ERD.
2. Buatlah ERD untuk kasus di bawah ini, menggunakan *software* EdrawMax/PowerDesigner dengan tahapan setiap langkah-langkah beserta kamus datanya!

Klinik Praktik Dokter Bersama

Suatu klinik memiliki praktik dokter bersama, sehingga dalam klinik tersebut memiliki banyak dokter. Seorang pasien, apabila akan berobat harus diperiksa oleh dokter dan sebaliknya dokter pun harus memeriksa pasien. Pasien yang berobat pada klinik tersebut lebih dari satu orang. Setiap selesai diperiksa pasien biasanya menerima resep berupa obat dan biasanya setiap pasien menerima beberapa jenis obat. Dapat diambil informasi tentang dokter adalah kode dokter, nama dokter, spesialis, dan tarif. Informasi tentang pasien adalah kode pasien, nama pasien, alamat, dan keluhan penyakit. Sedangkan informasi tentang obat adalah kode obat, nama obat, dan dosis.



PENGANTAR BASIS DATA LANJUTAN II



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus

1. Mahasiswa mampu mengetahui dan memahami normalisasi
2. Mahasiswa mampu mengetahui dan memahami lingkungan basis data



3.1 Normalisasi (*Normalization*)

Dalam hal ini kita akan membahas teknik normalisasi sedangkan model *entity relationship* telah dibahas pada materi sebelumnya. **Normalisasi** adalah proses pengelompokan data ke dalam bentuk tabel atau relasi untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk *database* yang mudah untuk dimodifikasi.

Pada proses normalisasi selalu diuji pada beberapa kondisi. Apakah ada kesulitan selama proses *insert*, *update*, dan *delete* dalam basis data yang disebut sebagai penyimpangan-penyimpangan atau dikenal dengan *anomaly*. Manfaat normalisasi adalah sebagai berikut:

1. Menghilangkan pengulangan atau duplikasi data.
2. Memperkecil pemborosan data.
3. Memperkecil kesalahan dalam proses penambahan, pembaharuan, dan penghapusan data.
4. Mengurangi penundaan dalam proses pemasukan dan pengeluaran data.
5. Mengurangi pemakaian kapasitas dari ruang penyimpanan (*memory*).
6. Mendukung kekonsistenan sebuah data.
7. Merupakan cara terbaik/metode untuk merancang tabel/entitas dalam satu kesatuan.

3.1.1 Macam-Macam *Anomaly*

Anomaly adalah penyimpangan dalam proses modifikasi data. Di bawah ini akan diberikan contoh *anomaly* pada tabel 3.1 sebagai berikut:

Tabel 3.1 Tabel Resep

No Pasien	Kode Obat	Harga Obat
P001	KD01	2000
P002	KD02	4500
P003	KD01	2000

Mari kita telaah untuk mengetahui dimana saja letak *anomaly* yang terjadi pada tabel resep, sesuai dengan jenis dari *anomaly* sebagai berikut:



1. Insertion Anomaly

Yaitu penyimpangan yang terjadi sebagai akibat operasi menyisipkan *record* pada sebuah relasi. Contoh: Jika ada obat baru yang akan dimasukkan/disisipkan, maka obat tersebut tidak dapat disisipkan ke dalam relasi sampai ada pasien yang mengambil jenis obat tersebut.

2. Deletion Anomaly

Yaitu penyimpangan yang terjadi sebagai akibat operasi penghapusan terhadap *record* dari sebuah relasi. Contoh: Jika pasien yang memiliki No_Pasien P001 membatalkan tidak jadi menebus resep obat tersebut, maka jika *record* tersebut dihapus akan menyebabkan hilangnya informasi tentang Kode_Obat KD01.

3. Update Anomaly

Yaitu penyimpangan yang terjadi sebagai akibat operasi perubahan *record* dari sebuah relasi. Contoh: Jika Harga_Obat untuk Kode_Obat KD01 dinaikkan dari 2000 menjadi 5000, maka harus dilakukan beberapa kali modifikasi terhadap *record-record* pasien yang menebus Kode_Obat KD01, agar data selalu tetap konsisten.

3.1.2 Ketergantungan Fungsional (*Functional Dependency*)

Ketergantungan fungsional memainkan peran kunci dalam menghasilkan perancangan basis data yang baik dari perancangan basis data yang buruk. V.K. Pallaw (2008:126) mengemukakan bahwa satu ketergantungan fungsional berada ketika nilai dari satu hal sepenuhnya ditentukan oleh yang lain. Keharusan menghilangkan masalah-masalah akibat ketergantungan yang harus dilakukan adalah jika struktur data dalam relasi dirancang sedemikian rupa sehingga atribut-atribut bukan kunci hanya tergantung pada atribut kunci dan tidak pada atribut yang lain. Misal diberikan sebuah relasi R, atribut Y dari R adalah bergantung fungsional pada atribut X dari R (dapat ditulis $R.X \rightarrow R.Y$) jika dan hanya jika setiap nilai X dalam R punya hubungan dengan tepat satu nilai Y dalam R.

Contoh:

Tabel Barang (Kode_Brg, Nama_Brg, Harga)

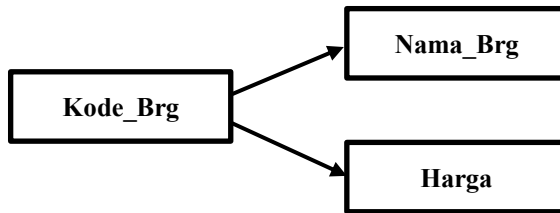


Kode Brg	Nama Brg	Harga
P01	PT. Indah Lestari	1000
P01	PT. Indah Lestari	1500
P01	PT. Indah Lestari	2000

Nama_Brg **FD** pada Kode_Brg ditulis Barang.Kode_Brg → Barang>Nama_Brg

Harga **FD** pada Kode_Brg ditulis Barang.Kode_Brg → Barang.Harga

Maka diagram *Functional Dependency*:



Gambar 3.1 Diagram Functional Dependency

Adapun untuk jenis-jenis ketergantungan fungsional sebagai berikut:

1. **Ketergantungan Fungsional Penuh (*Full Functional Dependency*)** yaitu atribut Y pada relasi R dikatakan ketergantungan fungsional penuh pada atribut X pada relasi R, jika Y tidak tergantung pada *subset* dari X (apabila X adalah *key* gabungan).

Contoh:

Tabel Kirim-Barang (No_Pem, Nama_Pem, No_Brg, Jumlah)

No_Pem	Nama_Pem	No_Brg	Jumlah
P01	PT. Indah Lestari	B01	1000
P01	PT. Indah Lestari	B02	1500
P01	PT. Indah Lestari	B03	2000
P02	PT. Sinar Surya	B03	2000
P03	CV. Harapan Masa	B02	1500



Ketergantungan Fungsional:

No_Pem → Nama_Pem

No_Brg, No_Pem → Jumlah (Tergantung penuh terhadap *key*nya)

- Ketergantungan Fungsional Transitif (*Transitive Functional Dependency*)** yaitu atribut Z pada relasi R dikatakan tergantung transitif pada atribut X, jika atribut Y tergantung pada atribut X pada relasi R dan atribut Z tergantung pada atribut Y pada relasi R, notasi ($X \rightarrow Y, Y \rightarrow Z$, maka $X \rightarrow Z$).

Contoh:

No Pem	Kode Kota	Kota	No Brg	Jumlah
P01	1	Jakarta	B01	1000
P01	1	Jakarta	B02	1500
P01	1	Jakarta	B03	2000
P02	3	Bandung	B03	2000
P03	2	Surabaya	B02	1500

Ketergantungan Transitif:

No_Pem → Kode_Kota

Kode_Kota → Kota, maka

No_Pem → Kota

3.1.3 Macam-Macam Atribut Tabel

Atribut adalah karakteristik atau sifat yang melekat pada sebuah tabel, atau disebut juga sebagai kolom data. Adapun untuk pengelompokan atribut tabel yaitu sebagai berikut:

- Atribut Key** adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam sebuah tabel secara *unique* (unik) dimana tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tertentu.
- Atribut Deskriptif** merupakan atribut yang bukan merupakan anggota dari *Primary Key*.



3. **Atribut Sederhana** adalah atribut atomik yang tidak dapat dipilih lagi. Contoh: NPM, Nama Mahasiswa.
4. **Atribut Komposit** adalah atribut yang masih bisa diuraikan lagi menjadi sub-atribut yang masing-masing memiliki makna. Contoh: Alamat → Alamat, Kota, Provinsi, Kode Pos.
5. **Atribut Bernilai Tunggal** merupakan atribut-atribut yang hanya ditujukan pada atribut yang memiliki paling banyak satu nilai untuk setiap baris data. Contoh: NPM, Nama Mahasiswa, Tanggal Lahir → hanya dapat berisi satu nilai untuk seorang mahasiswa.
6. **Atribut Bernilai Banyak** merupakan atribut-atribut yang hanya ditujukan pada atribut yang dapat diisi dengan lebih dari satu nilai, tetapi jenisnya sama. Contoh: pada tabel mahasiswa dapat ditambahkan atribut Hobi, karena seorang mahasiswa dapat memiliki beberapa hobi.
7. **Atribut Harus Bernilai (*Mandatory*)** adalah atribut yang nilainya tidak boleh kosong, atau harus ada nilainya. Misalnya NPM dan Nama Mahasiswa. Nilai NULL digunakan untuk mengisi atribut yang nilainya belum siap atau tidak ada.

3.1.4 Jenis-Jenis *Key*

Key adalah sebuah atau gabungan dari beberapa atribut/*field* yang dapat membedakan semua *record* dalam tabel secara unik. Artinya, jika suatu atribut dijadikan sebagai *key* maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut. Ada beberapa macam *key* yang dapat diterapkan pada suatu tabel sebagai berikut:

1. **Super Key** yaitu merupakan satu atau lebih atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
Contoh: Pada tabel Mahasiswa {NPM, Nm_Mhsw, Tgl_Lahir, Alamat}
Maka yang dapat dijadikan *Super Key* adalah:
 - {NPM, Nm_Mhsw, Alamat, Tgl_Lahir}
 - {NPM, Nm_Mhsw}
 - {NPM, Tgl_Lahir}

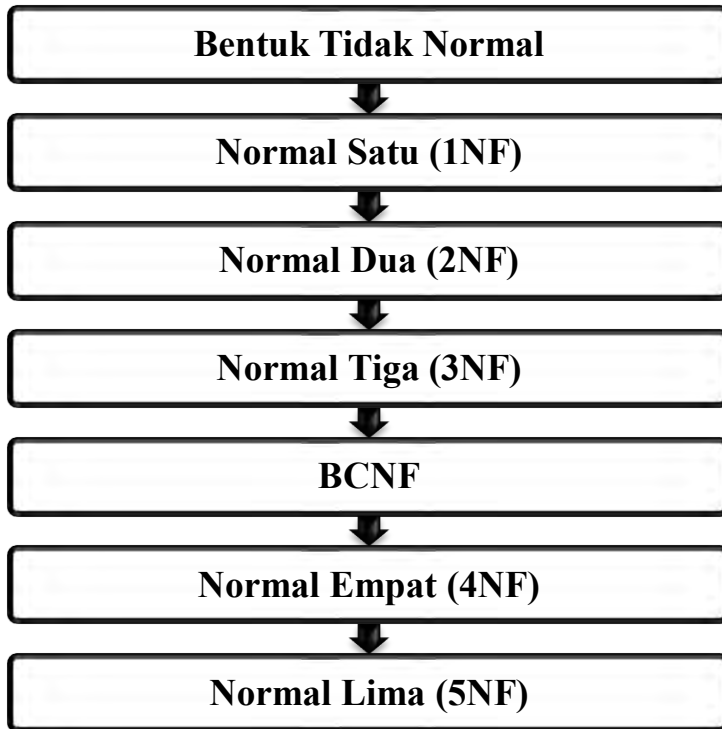


- {Nm_Mhsw}
 - {NPM}
2. **Candidate Key** adalah kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Sebuah *Candidate Key* tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi *Super Key* yang lain. Jadi sebuah *Candidate Key* pastilah *Super Key*, tetapi belum tentu sebaliknya. Pada contoh di atas, maka *Candidate Key* adalah:
 - {NPM}
 - {Nm_Mhsw}
 3. **Primary Key** adalah memilih sebuah *Candidate Key*, dimana jaminan keunikan *key*-nya lebih baik. Pada contoh di atas maka *Primary Key* adalah:
 - {NPM}
 4. **Alternate Key** adalah *Candidate Key* yang tidak dijadikan *Primary Key*.
 5. **Composite Key** adalah *key* yang terdiri dari dua atribut atau lebih, dimana atribut-atribut tersebut bila berdiri sendiri tidak menjadi identitas *record*, tetapi bila dirangkaikan menjadi satu kesatuan yang dapat mengidentifikasi *record* secara unik.
 6. **Foreign Key** adalah *non key* atribut pada sebuah relasi yang juga menjadi *primary key* atribut pada relasi lainnya. *Foreign Key* biasanya digunakan sebagai penghubung antara *record-record* dari kedua relasi tersebut.

3.1.5 Langkah-Langkah Pembentukan Normalisasi

Proses normalisasi sebagai berikut:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.
Berikut langkah-langkah pembentukan normalisasi:



Gambar 3.2 Diagram Langkah-Langkah Pembentukan Normalisasi

Keterangan gambar 3.2 sebagai berikut:

1. **Bentuk Tidak Normal (*Unnormalized Form*)** yaitu bentuk ini merupakan kumpulan data yang tidak ada keharusan mengikuti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Bentuk ini menghilangkan perulangan *group*.
2. **Bentuk Normal Satu (1NF/*First Normal Form*)** yaitu apabila relasi tersebut mempunyai nilai data yang atomik, artinya tidak ada lagi kerangkapan data. Bentuk ini akan menghilangkan ketergantungan sebagian.



3. **Bentuk Normal Dua (2NF/Second Normal Form)** yaitu apabila relasi tersebut merupakan 1NF dan setiap atribut ketergantungan fungsional penuh pada *primary key*. Bentuk ini akan menghilangkan ketergantungan transitif.
4. **Bentuk Normal Tiga (3NF/Third Normal Form)** yaitu apabila relasi merupakan 2NF dan tidak ketergantungan fungsional secara transitif pada *primary key*. Bentuk ini menghilangkan *anomaly* hasil dari ketergantungan fungsional.
5. **Bentuk Normal Boyce – Codd (BCNF/Boyce – Codd Normal Form)** ditemukan oleh R. F. Boyce dan E. F. Codd yaitu apabila relasi merupakan 3NF dan semua determinannya merupakan *candidate key*. Bentuk ini menghilangkan ketergantungan *multivalued*.
6. **Bentuk Normal Empat (4NF/Fourth Normal Form)** yaitu apabila relasi merupakan BCNF, yang mana rinci data yang ada di dalamnya tidak mengalami ketergantungan pada banyak nilai atau semua rinci data yang mengalami ketergantungan pada banyak nilai adalah juga mengalami ketergantungan. Bentuk ini menghilangkan *anomaly* yang tersisa.
7. **Bentuk Normal Lima (5NF/Fifth Normal Form)** yaitu suatu relasi berada dalam bentuk normal kelima jika relasi data yang ada di dalam struktur data tidak dapat direkonstruksi dari struktur data yang memuat rinci data lebih sedikit.

3.1.6 Contoh Normalisasi

Normalisasi pada *Database* Perkuliahan.

Asumsikan bahwa:

- Seorang mahasiswa dapat mengambil beberapa mata kuliah.
- Satu mata kuliah dapat diambil oleh lebih dari satu mahasiswa.
- Satu mata kuliah hanya diajarkan oleh satu dosen.
- Satu dosen dapat mengajar beberapa mata kuliah.
- Seorang mahasiswa pada mata kuliah tertentu hanya mempunyai satu nilai.



Tabel MAHASISWA-1 (Unnormal)

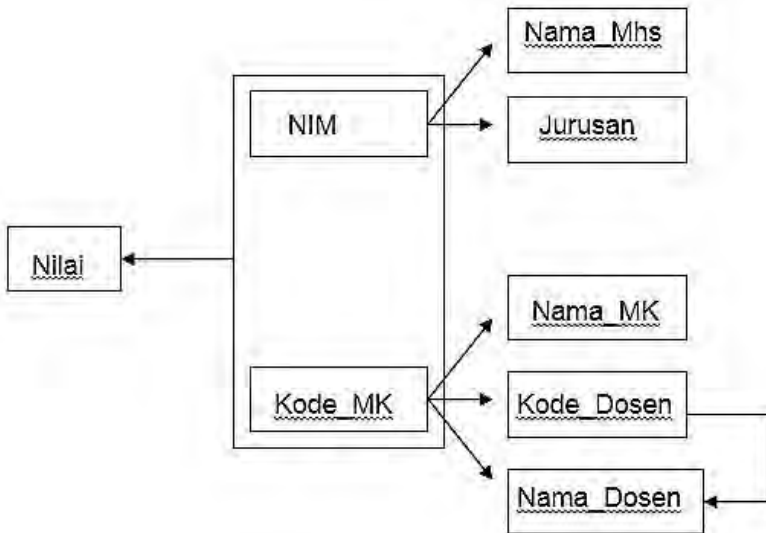
NIM	Nama Mhs	Jurusan	Kode MK	Nama MK	Kode Dosen	Nama Dosen	Nilai
2683	Mondy	TI	TI350	Manajemen Basis Data	B104	Andi	A
			TI465	Pemrograman Visual	B317	Helmy	B
5432	Boy	AK	TI350	Manajemen Basis Data	B104	Andi	C
			AK201	Akuntansi Keuangan	D310	Lia	B
			MK300	Dasar Pemasaran	B212	Dina	A

Tabel MAHASISWA-2 (1NF)

NIM	Nama Mhs	Jurusan	Kode MK	Nama MK	Kode Dosen	Nama Dosen	Nilai
2683	Mondy	TI	TI350	Manajemen Basis Data	B104	Andi	A
2683	Mondy	TI	TI465	Pemrograman Visual	B317	Helmy	B
5432	Boy	AK	TI350	Manajemen Basis Data	B104	Andi	C
5432	Boy	AK	AK201	Akuntansi Keuangan	D310	Lia	B
5432	Boy	AK	MK300	Dasar Pemasaran	B212	Dina	A



Diagram Ketergantungan Fungsional



Tabel KULIAH (2NF)

Kode_MK	Nama_MK	Kode_Dosen	Nama_Dosen
TI350	Manajemen Basis Data	B104	Andi
TI465	Pemrograman Visual	B317	Helmy
AK201	Akuntansi Keuangan	D310	Lia
MK300	Dasar Pemasaran	B212	Dina

Tabel MAHASISWA-3 (3NF)

NIM	Nama_Mhs	Jurusan
2683	Mondy	TI
5432	Boy	AK

Tabel NILAI (3NF)

NIM	Kode_MK	Nilai
2683	TI350	A
2683	TI465	B
5432	TI350	C
5432	AK201	B



NIM	Kode MK	Nilai
5432	MK300	A

Tabel MATAKULIAH-1 (3NF)

Kode MK	Nama MK	Kode Dosen
TI350	Manajemen Basis Data	B104
TI465	Pemrograman Visual	B317
AK201	Akuntansi Keuangan	D310
MK300	Dasar Pemasaran	B212

Tabel DOSEN (3NF)

Kode Dosen	Nama Dosen
B104	Andi
B317	Helmy
B104	Andi
D310	Lia
B212	Dina

3.2 Lingkungan Basis Data

3.2.1 Masalah dalam Basis Data

Penyusunan basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu sebagai berikut:

1. **Redundansi Data** yaitu munculnya data-data yang sama secara berulang-ulang pada beberapa *file* basis data yang semestinya tidak diperlukan, akan mengakibatkan proses pembaharuan lebih lama dan memungkinkan ketidakkonsistenan data.
2. **Inkonsistensi Data** yaitu munculnya data yang tidak konsisten pada *field* yang sama untuk beberapa *file* dengan kunci yang sama. Terjadi akibat kesalahan pada hasil pengolahan basis data yang tidak sesuai dengan fakta.
3. **Isolasi Data untuk Standarisasi** disebabkan oleh pemakaian beberapa *file* basis data yang tersebar dalam beberapa *file*, hal ini menyulitkan *programmer* untuk mengambil dan menyimpan data.
4. **Banyak Pemakai (Multi User)** yaitu basis data dapat diakses oleh beberapa pemakai secara simultan (serentak/bersamaan), karena



data yang diolah tidak bergantung dan menyatu dalam program namun terlepas dalam satu kelompok data.

5. **Masalah Keamanan (*Security Problem*)** yaitu pada prinsipnya *file* basis data hanya boleh diakses oleh pemakai tertentu yang mempunyai hak/kewenangan akses basis data. Pembatasan dapat dilakukan melalui DBMS atau program aplikasi basis data.
6. **Masalah Integritas (*Integrity Problem*)** yaitu untuk menjaga agar unjuk kerja sistem tetap dalam pengendalian penuh. Secara teknis maka ada kunci primer (*primary key*) yang menghubungkan beberapa *file* yang saling berkaitan.
7. **Masalah Kebebasan Data (*Data Independence Problem*)** yaitu basis data yang dirancang hendaknya tidak bergantung pada program aplikasi yang dibangun. Sehingga apabila ada perubahan terhadap *field*, tidak perlu mengubah programnya.

3.2.2 Organisasi *File* Sistem Basis Data

Adapun untuk tujuan organisasi *file* dalam sistem basis data adalah sebagai berikut:

1. Menyediakan sarana pencarian *record* bagi pengolahan, seleksi, atau penyaringan.
2. Memudahkan pembuatan atau pemeliharaan *file*.

Ada dua jenis media penyimpanan *file* sebagai berikut:

1. *Sequential Access Storage Device* (SASD)
 - Proses pembacaan *record* harus berurutan.
 - Tidak ada pengalamatan.
 - Data disimpan dalam bentuk blok.
 - Proses penulisan hanya bisa dilakukan sekali.
 - Contoh pada *magnetic tape*.
2. *Direct Access Storage Device* (DASD)
 - Proses pembacaan *record* tidak harus urut.
 - Mempunyai alamat.
 - Data dapat disimpan dalam karakter atau blok.



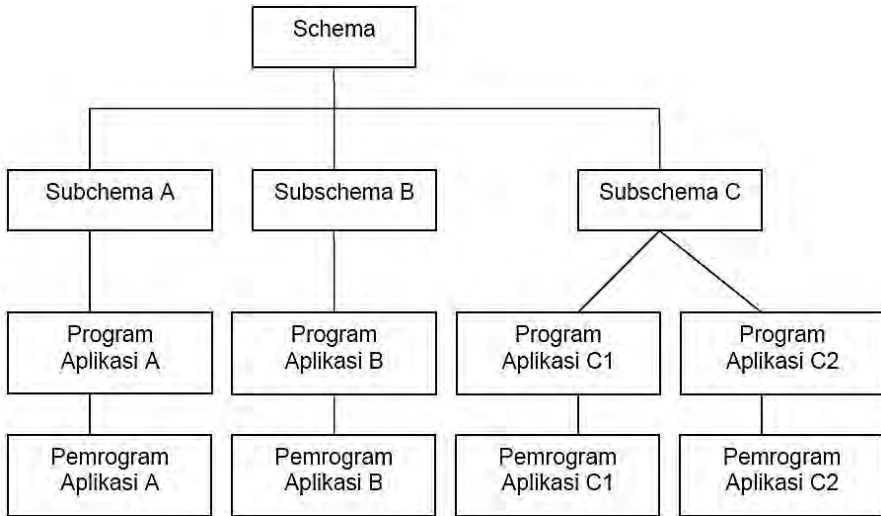
- Proses penulisan dapat dilakukan beberapa kali.
- Contoh: *hard disk*, *floppy disk*.

Ada empat jenis metode susunan *file* sebagai berikut:

1. *Sequential* (urut)
 - *Record* disimpan berdasarkan suatu kunci.
 - Pencarian *record* tertentu dilakukan *record* demi *record* berdasarkan kuncinya.
2. *Random* (acak)
 - Kunci *record* ditransformasikan ke alamat penyimpanan dalam media fisik secara acak.
3. *Indexed Sequential*
 - Merupakan gabungan antara metode urut dan acak.
 - Record disimpan secara berurutan dengan menggunakan kunci.
 - Masing-masing record memiliki indeks.
 - Pengalamatan dilakukan secara acak.
4. *Indexed Random*
 - *Record* disimpan secara acak.
 - Masing-masing *record* memiliki indeks.

3.2.3 Schema dan Subschema

Schema dan *Subschema* diperlukan untuk menggambarkan hubungan logik antara data dalam basis data. *Schema* berfungsi untuk memberikan deskripsi hubungan logik secara lengkap dari basis data, yang meliputi rinci data, *record*, *set*, dan area untuk aplikasi yang menggunakan basis data tersebut. *Subschema* merupakan deskripsi terpisah dari rinci data, *record*, *set*, dan area yang digunakan oleh program aplikasi. Gambar 3.3 merupakan diagram hubungan hierarki *schema* dan *subschema*.



Gambar 3.3 Diagram Hubungan Hierarki *Schema* dan *Subschema*

3.2.4 Kamus Data

DBMS memberikan fasilitas *data dictionary* (kamus data) untuk mendefinisikan nama-nama rinci data dan format penyimpanannya.

Kamus data digunakan untuk sebagai berikut:

1. Pada tahap analisis sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.
2. Pada tahap perancangan sistem digunakan untuk merancang masukan, laporan-laporan, dan *database*.

Kamus data berisi sebagai berikut:

1. Nama arus data.
2. Bentuk data.
3. Arus data.
4. Penjelasan atau keterangan-keterangan.
5. Periode terjadinya transaksi.

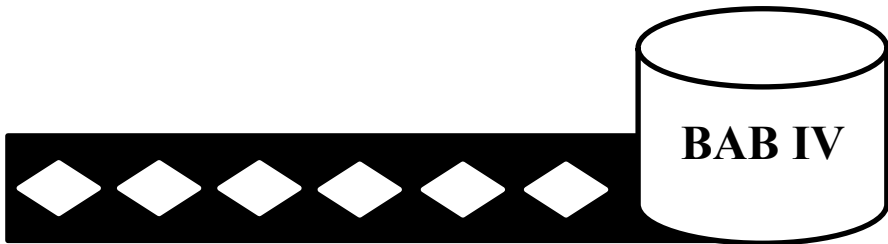


6. Volume arus data yang mengalir dalam periode tertentu.
7. Struktur data.

Latihan Praktik

Normalisasikan dan tentukanlah ketergantungan fungsional dari skema *table* di bawah ini, sesuai dengan materi yang telah Anda dapatkan pada bab ini!

NRP	Nama_Mhs	Alamat_Mhs	Tgl_Lahir	Kode_Mk	Nama_MK	SKS	Semester	Nilai	Waktu	Ruang	Nama_Dsn	Alamat_Dsn
2698100001	Marisha Koirala	Jl. Bombay No. 09, Surabaya, 60009	21-09-1979	IF-410	Struktur Data	3	2	A	Senin, 08.00 - 10.15 Kamis, 10.30 - 12.20	Ruang A	Prof. Ali Khan	Jl. Gebang 21, Surabaya, 60021
2698100001	Marisha Koirala	Jl. Bombay No. 09, Surabaya, 60009	21-09-1979	IF-411	Basis Data	3	3	A	Selasa, 10.30 - 12.20 Jumat, 08.00 - 10.15	Ruang B	Prof. Sharukh Khan	Jl. Kepuh 12, Surabaya, 60012
2698100001	Marisha Koirala	Jl. Bombay No. 09, Surabaya, 60009	21-09-1979	IF-412	Jaringan Komputer	3	4	B	Rabu, 08.00 - 10.15	Ruang J	Dr. Ajay Khan	Jl. Mulyosari 42, Surabaya, 60042
2698100002	Amir Khan	Jl. Bolly No. 12, Surabaya, 60012	12-12-1972	IF-411	Basis Data	3	3	B	Selasa, 10.30 - 12.20 Jumat, 08.00 - 10.15	Ruang B	Prof. Sharukh Khan	Jl. Kepuh 12, Surabaya, 60012
2698100002	Amir Khan	Jl. Bolly No. 12, Surabaya, 60012	12-12-1972	IF-417	Administrasi Basis Data	3	3	B	Rabu, 10.30 - 12.20 Kamis, 08.00 - 10.15	Ruang B	Prof. Sharukh Khan	Jl. Kepuh 12, Surabaya, 60012
2698100004	Salman Khan	Jl. KhanKhan No. 06, Surabaya, 60006	06-06-1976	IF-411	Basis Data	3	3	A	Selasa, 10.30 - 12.20 Jumat, 08.00 - 10.15	Ruang B	Prof. Sharukh Khan	Jl. Kepuh 12, Surabaya, 60012
2698100004	Salman Khan	Jl. KhanKhan No. 06, Surabaya, 60006	06-06-1976	IF-410	Struktur Data	3	2	C	Senin, 08.00 - 10.15 Kamis, 10.30 - 12.20	Ruang A	Prof. Ali Khan	Jl. Gebang 21, Surabaya, 60011



MENGENAL DAN MEMULAI BEKERJA DENGAN MICROSOFT OFFICE ACCESS 2013



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus

1. Mahasiswa mampu mengenal Microsoft Office Access 2013
2. Mahasiswa mampu mengetahui dan memahami keunggulan dan kelemahan Microsoft Office Access 2013
3. Mahasiswa mampu mengetahui dan memahami komponen-komponen di dalam Microsoft Office Access 2013
4. Mahasiswa mampu mengetahui dan memahami objek-objek di dalam Microsoft Office Access 2013
5. Mahasiswa mampu mengetahui dan memahami struktur *database* di dalam Microsoft Office Access 2013
6. Mahasiswa mampu memulai bekerja di dalam Microsoft Office Access 2013
7. Mahasiswa mampu membuat *database* menggunakan Microsoft Office Access 2013



4.1 Pengenalan Microsoft Office Access 2013

Microsoft Office Access adalah sebuah *software* khusus untuk menyimpan dan mengelola data secara *database*. Jika Anda mulai menyimpan data yang banyak, maka Anda sudah membutuhkan Microsoft Access. Anda bisa saja menyimpan dan mengelola data menggunakan *spreadsheet* seperti Microsoft Excel, namun jika data tersebut sudah banyak dan penambahan data terus dilakukan. Anda perlu menggunakan *software* yang khusus disediakan untuk menyimpan dan mengelola data. Saat ini sebuah *database* yang mudah dan instan tidak ada lagi selain Microsoft Access. Ada yang masih menggunakan Microsoft Foxpro atau Microsoft Visual Foxpro, tetapi tidak semudah Microsoft Access. *Software database* kelas ini umumnya disebut *desktop database*. *File database* Access bisa terdiri dari sebuah *file* saja yang bisa dibawa kemana-mana, bisa dibuka di komputer mana saja yang mempunyai Access versi yang sama atau yang lebih baru. Di dalam *file* Access kita bisa membuat *table* untuk menyimpan data, membuat *form*, *report*, dan program aplikasi untuk mengelola data.

Jika Anda memikirkan *software database* dan telah mendengar misalnya **MS SQL Server**, **Oracle**, atau **MySQL**, *software* tersebut tergolong kelas *database server*. *Software database server* umumnya harus diinstal di komputer *server*, dan akan melayani permintaan (*request*) data dari komputer *client* melalui jaringan komputer (LAN). Jika Anda mendapatkan edisi yang bisa diinstal pada laptop atau desktop, Anda tetap harus melakukan koneksi ke sistem *database* tersebut untuk membuka dan mengelola data. Anda mungkin juga mendengar pengelolaan *database* menggunakan **Visual Basic (VB)**, **Delphi**, atau **PowerBuilder**. *Software* tersebut adalah untuk membuat program aplikasi yang akan mengelola *database*, bukan *tools* untuk mengelola *database* secara langsung. *Database* yang digunakannya mungkin berupa *database* Microsoft Office Access atau salah satu *database server* tersebut di atas.

Perlu diketahui bahwa, Microsoft Office Access tergolong dalam sistem *database* relasional atau **RDBMS (Relational Database Management System)**, sama seperti MS SQL Server, Oracle, MySQL, dan



lain-lain. Dengan demikian cara dan teknologi *database* yang digunakan sama sehingga mudah dikonversi. Setelah *database* kita banyak, kita bisa melakukan *upgrade* ke sistem *database server*. Microsoft Office Access sebagai *database tool* juga bisa digunakan langsung untuk mengelola data yang disimpan pada *database server*. Microsoft Office 2013 merupakan versi terbaru dari Microsoft Office saat ini yang sebelumnya Microsoft merilis Office 2010. Menurut Wikipedia, Microsoft Office Access dirilis untuk manufaktur pada tanggal 11 Oktober 2012 dan dirilis secara umum pada tanggal 29 Januari 2013. Sekilas tidak ada yang terlalu mencolok dari perubahan yang dilakukan pihak Microsoft pada Office 2013 jika dilihat dari versi-versi sebelumnya. Berikut perbedaan secara ringkas dan umum dari setiap versi Microsoft Office Access:

- **Microsoft Office Access 2003** menggunakan format *file* dengan ekstensi **.mdb**. Pada versi 2003 tersedia sistem pengamanan yang disebut ULS (*User Level Security*) yang sangat kuat namun sulit dipahami oleh pengguna biasa. Pada versi ini masih menggunakan sistem *menubar* dan *toolbar*.
- **Microsoft Office Access 2007** memperkenalkan format *file* baru yaitu **.accdb**. Sistem pengaman ULS sudah dikeluarkan, namun jika sudah dipasang pada versi 2003, ULS tersebut masih tetap berlaku. Pada versi ini sistem *menubar* dan *toolbar* diganti dengan *ribbon*.
- **Microsoft Office Access 2010** melanjutkan versi 2007, dengan sedikit penambahan hal baru dan perubahan tampilan. Pada versi ini mulai diperkenalkan *Web Database* dengan menggunakan Microsoft SharePoint *Server* dan Access *Server*.
- **Microsoft Office Access 2013** muncul memperkenalkan *Web App* sebagai kelanjutan dari *Web Database*, di samping *database* yang asli yang disebut *desktop database*. Pada versi ini kita tidak bisa lagi membuat dan menggunakan *Access Project*, sebuah kemampuan Access untuk mengelola *database MS SQL Server* secara langsung. Jika diamati sepertinya Microsoft Office Access 2013 lebih menekankan ke model *TouchScreen* dalam setiap paket



produknya, ini mungkin ada hubungannya dengan Windows 8 yang baru dirilis juga menekankan ke model *TouchScreen*.

4.2 Keunggulan Microsoft Office Access 2013

Adapun keunggulannya antara lain sebagai berikut:

1. Aplikasinya sangat mudah didapatkan (walaupun bajakan).
2. Mudahnya memanipulasi tabel dan data.
3. Mudahnya merelasi antar tabel.
4. Dapat memberikan perintah SQL.
5. Bisa diintegrasikan dengan bahasa pemrograman, misalnya Visual Basic dan lain-lain.
6. Dapat menyimpan data atau *file* dalam jumlah yang besar (jauh lebih besar dari Paradox).
7. Data dimasukkan dengan cara yang hampir sama dengan Microsoft Office Excel.
8. Untuk menampilkan isi *table* dan melakukan pemasukan (juga *edit* dan hapus) data, bisa menggunakan *datasheet view* tanpa perlu membuat apa pun.
9. Bisa menerapkan kaidah *database* sehingga mengurangi duplikasi data, sehingga data lebih akurat dan konsisten.
10. Pemasukan data lebih mudah dan cepat dengan *form*.
11. Informasi bisa dihasilkan dengan *query* dan ditampilkan dalam *datasheet*, *form*, dan *report*.
12. *Multiuser*.
13. Program aplikasi yang berjalan otomatis, sehingga mengurangi pekerjaan prosedural yang rutin atau berulang-ulang.
14. Menerapkan aturan keamanan (*security*) yang berlapis-lapis untuk menjaga data maupun program aplikasi.

4.3 Kelemahan Microsoft Office Access 2013

Adapun kelemahannya antara lain sebagai berikut:

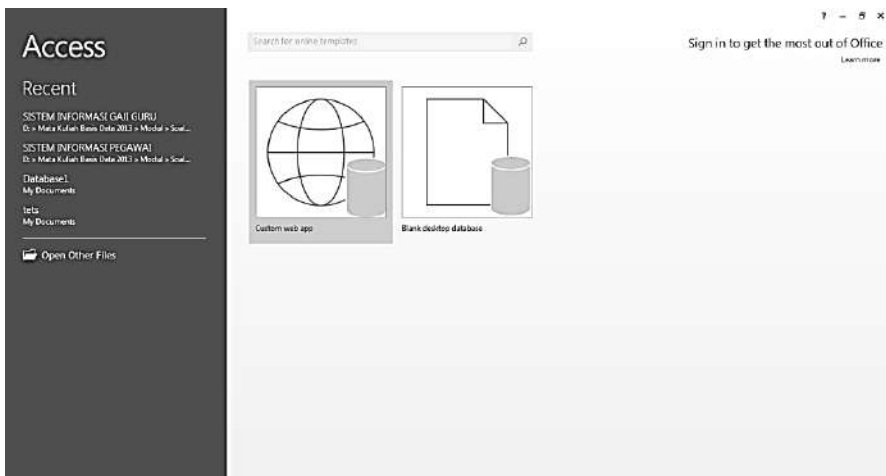
1. Aplikasi yang asli ini berbayar (tidak gratis).



2. Jika diakses melalui jaringan tidak begitu bagus, sehingga banyak pengguna menggunakan solusi sistem manajemen basis data yang bersifat *client/server*.
3. Hanya bisa dijalankan di sistem operasi Windows saja.

4.4 Komponen-Komponen Microsoft Office Access 2013

4.4.1 Tampilan layar depan Microsoft Office Access 2013

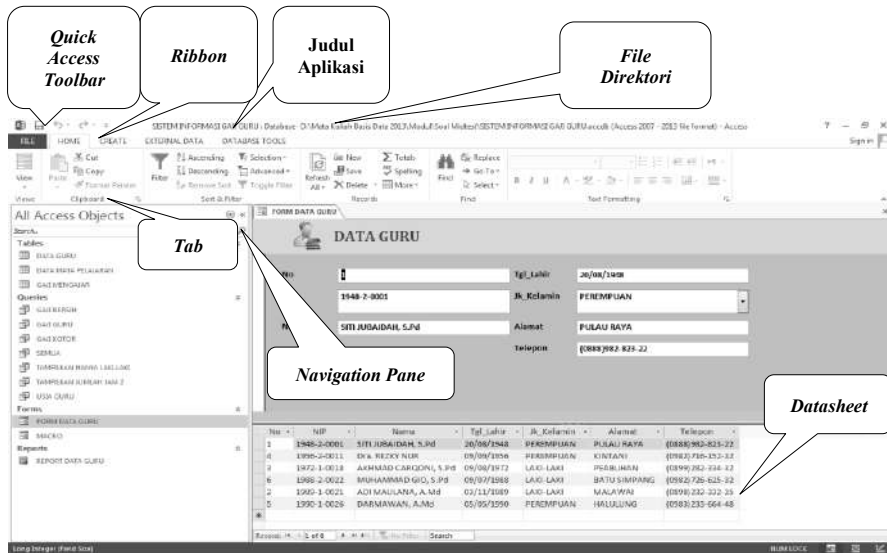


Gambar 4.1 Tampilan Layar Depan Microsoft Office Access 2013

Ketika Anda menjalankan Microsoft Office Access 2013, akan tampak tampilan seperti gambar di atas. Kita bisa membuat *file database baru*, baik berupa *Web App* maupun berupa *Desktop Database*. Tersedia juga *templete* yang akan membantu kita dalam membuat *database* sesuai subjeknya. Selain itu pada tampilan ini kita bisa membuka *file database* yang sudah ada, baik dengan memilih dari daftar *Recent* maupun dengan *Open Other Files*. *File* yang dibuat dengan Access disebut *file database* berekstensi **.accdb**.



4.4.2 Beberapa Komponen Microsoft Office Access 2013

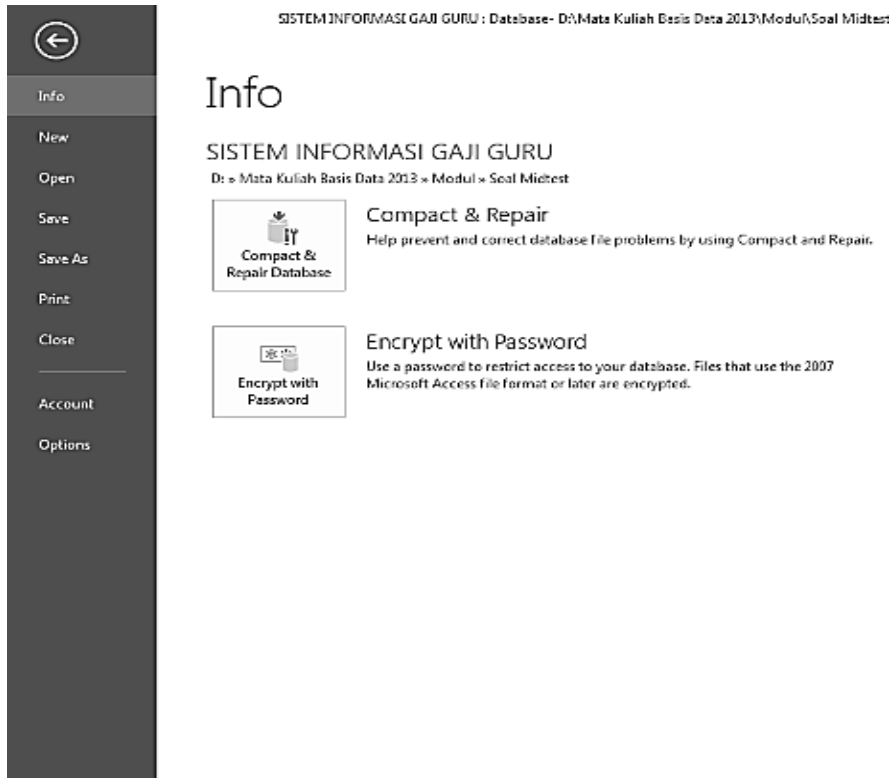


Gambar 4.2 Komponen Microsoft Office Access 2013

- **Quick Access Toolbar** adalah *toolbar* mini yang memuat perintah-perintah yang paling sering dipakai.
- **Ribbon** yaitu tombol-tombol perintah Access, *Ribbon* merupakan pengganti dari *Menu Bar* dan *Toolbar*.
- **Navigation Pane** yaitu bidang yang memuat daftar objek yang ada dalam *file database*.
- **Document Tab** yaitu *tab* untuk setiap objek yang sedang dibuka.
- **Datasheet** yaitu tampilan berbentuk tabel untuk mengelola data.

Selain komponen yang terlihat pada gambar, masih banyak komponen lain yang akan muncul sesuai konteks. Pada *Ribbon* terdapat beberapa *Tab* yang dapat digunakan oleh *user* untuk manajemen basis data.

4.4.3 Ribbon Tab File



Gambar 4.3 Ribbon Tab File

Ribbon Tab File yang mana di dalamnya terdapat beberapa pilihan atau perintah:

- Info untuk me-*repair* dan memproteksi *password*.
- *New* untuk membuat *file* baru.
- *Open* untuk membuka *file* yang tersimpan dalam suatu direktori *hard disk*.
- *Save* untuk melakukan penyimpanan.
- *Save As* untuk melakukan penyimpanan jenis ekstensi *file* yang berbeda.



- *Print* untuk melakukan pencetakan.
- *Close* untuk menutup halaman *database* yang sedang terbuka.
- *Account* untuk melihat keterangan akun yang sedang menggunakan dan juga keterangan produk.
- *Option* berisi tentang pengaturan opsional.

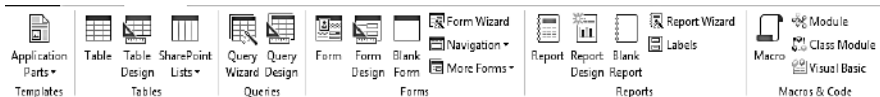
4.4.4 Ribbon Tab Home



Gambar 4.4 Ribbon Tab Home

Ribbon Tab Home berisi *tools* yang dapat digunakan oleh *user* untuk mengedit format dari data *file* basis data.

4.4.5 Ribbon Tab Create

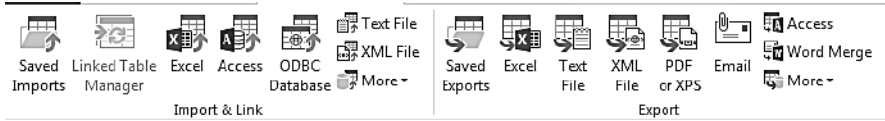


Gambar 4.5 Ribbon Tab Create

Ribbon Tab Create yang berisi *tool* yang dapat digunakan oleh *user* untuk melakukan penambahan *table*, *query*, *form*, *report*, dan lain-lain.



4.4.6 Ribbon Tab External Data



Gambar 4.6 Ribbon Tab External Data

Ribbon Tab External Data yang berisi tool yang dapat digunakan oleh user untuk melakukan penambahan data dari file lain.

4.4.7 Ribbon Tab Database Tools



Gambar 4.7 Ribbon Tab Database Tools

Ribbon Tab Database Tools yang berisi tool yang dapat digunakan oleh user untuk melakukan pembuatan atau menghubungkan database atau table yang telah dibuat.

4.4.8 Tools



Gambar 4.8 Contoh Tools

Tools merupakan rincian yang ada dalam Ribbon Tab.



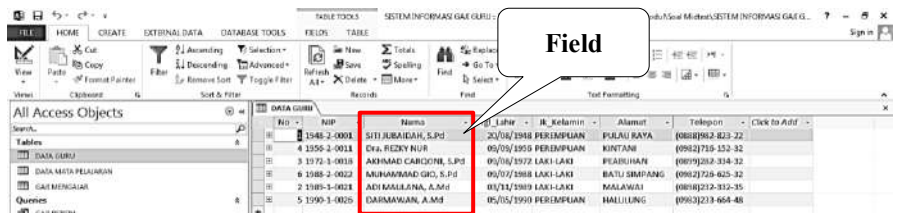
4.4.9 Sidebar



Gambar 4.9 Sidebar

Sidebar yang berisi objek-objek yang sedang aktif dalam halaman file seperti table, form, query, report, dan lain-lain.

4.4.10 Field



Gambar 4.10 Field

Field yang merupakan tempat untuk melakukan pengeditan data dalam Microsoft Office Access 2013.



4.5 Objek-Objek Microsoft Office Access 2013

4.5.1 Table

No	NIP	Nama	Tgl_Lahir	Jk_Kelamin	Alamat	Telepon
1	1943-2-0001	SITI JUBAIDAH, S.Pd	20/08/1948	PEREMPUAN	PULAU RAYA	(0888)982-823-22
4	1955-2-0011	Dra. REZKY NUR	09/09/1956	PEREMPUAN	KINTANI	(0982)716-152-32
3	1972-1-0018	AKHMAD CARQONI, S.Pd	09/08/1972	LAKI-LAKI	PEABUHAN	(0899)282-334-32
6	1988-2-0022	MUHAMMAD GIO, S.Pd	09/07/1988	LAKI-LAKI	BATU SIMPANG	(0982)726-025-32
2	1989-1-0021	ADI MAULANA, A.Md	03/11/1989	LAKI-LAKI	MALAWAI	(0898)232-332-35
5	1990-1-0026	DARMAWAN, A.Md	05/05/1990	PEREMPUAN	HALULUNG	(0983)233-664-48

Gambar 4.11 Contoh Table

4.5.2 Query

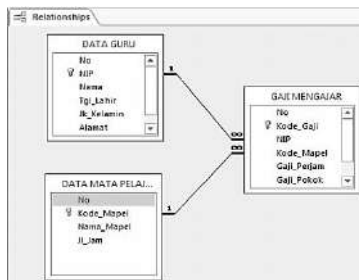
Field: NIP Nama Jk_Kelamin
 Table: DATA GURU DATA GURU DATA GURU
 Sort:
 Show:
 Criteria: Like "LAKI-LAKI"
 or:

TAMPILKAN HANYA LAKI-LAKI

NIP	Nama	Jk_Kelamin
1989-1-0021	ADI MAULANA, A.Md	LAKI-LAKI
1972-1-0018	AKHMAD CARQONI, S.Pd	LAKI-LAKI
1988-2-0022	MUHAMMAD GIO, S.Pd	LAKI-LAKI

Gambar 4.12 Contoh Desain dan Hasil Query

4.5.3 Relationship



Gambar 4.13 Contoh Relationship



4.5.4 Form

FORM DATA GURU

DATA GURU

No: Tgl_Lahir:

NIP: Jk_Kelamin:

Nama: Alamat:

Telepon:

No	NIP	Nama	Tgl_Lahir	Jk_Kelamin	Alamat	Telepon
1	1948-2-0001	SITI JUBAIDAH, S.Pd	20/08/1948	PEREMPUAN	PULAU RAYA	(0888)982-823-22
4	1956-2-0011	Dra. REZKY NUR	09/09/1956	PEREMPUAN	KINTANI	(0982)716-152-32
3	1972-1-0018	AKHMAD CARQONI, S.Pd	09/08/1972	LAKI-LAKI	PEABUHAN	(0899)282-334-32
6	1988-2-0022	MUHAMMAD GIO, S.Pd	09/07/1988	LAKI-LAKI	BATU SIMPANG	(0982)726-625-32
2	1989-1-0021	ADI MAULANA, A.Md	03/11/1989	LAKI-LAKI	MALAWAI	(0898)232-332-35
5	1990-1-0026	DARMAWAN, A.Md	05/05/1990	PEREMPUAN	HALULUNG	(0983)233-664-48

Record: 4 of 6

Gambar 4.14 Contoh Form

4.5.5 Report

REPORT DATA GURU

DATA GURU 25 Januari 2015 14:26:53

No	NIP	Nama	Tgl_Lahir	Jk_Kelamin	Alamat	Telepon
1	1948-2-0001	SITI JUBAIDAH, S.Pd	20/08/1948	PEREMPUAN	PULAU RAYA	(0888)982-823-22
2	1989-1-0021	ADI MAULANA, A.Md	03/11/1989	LAKI-LAKI	MALAWAI	(0898)232-332-35
3	1972-1-0018	AKHMAD CARQONI, S.Pd	09/08/1972	LAKI-LAKI	PEABUHAN	(0899)282-334-32
4	1956-2-0011	Dra. REZKY NUR	09/09/1956	PEREMPUAN	KINTANI	(0982)716-152-32
5	1990-1-0026	DARMAWAN, A.Md	05/05/1990	PEREMPUAN	HALULUNG	(0983)233-664-48
6	1988-2-0022	MUHAMMAD GIO, S.Pd	09/07/1988	LAKI-LAKI	BATU SIMPANG	(0982)726-625-32

Page 1 of 1

Gambar 4.15 Contoh Report



4.5.6 Macro

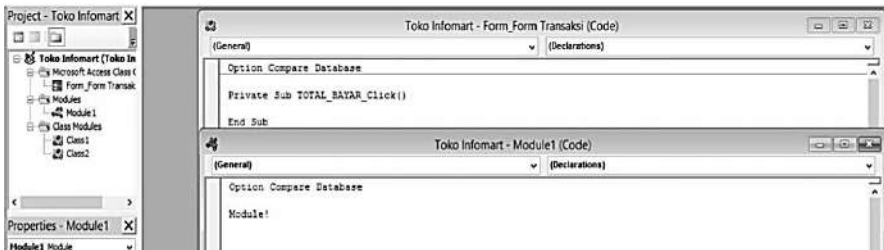
```
MACRO : Command47 : On Click

If [Form],[Dirty] Then
    RunMenuCommand
        Command SaveRecord
    End If
GoToRecord
    Object Type
    Object Name
        Record New
        Offset
Submacro: ErrorHandler
    MsgBox
        Message [MacroError].[Description]
        Beep Yes
        Type None
        Title
End Submacro

Add New Action
```

Gambar 4.16 Contoh Macro

4.5.7 Module



Gambar 4.17 Contoh Module VBA

4.6 Struktur Basis Data (Database)

Sebelum mulai membuat *table* dan objek aplikasi, terlebih dahulu kita perlu merancang struktur *database*. Struktur *database* adalah



konfigurasi *table* yang menyimpan data. Kalau di dalam Microsoft Office Access sendiri struktur *database* adalah sekumpulan objek yang terdiri dari *table, query, form, report*. Objek-objek ini ditampung dalam satu wadah atau *database*. Kita menentukan struktur *database* berdasarkan cakupan data yang ingin disimpan dalam *database*.

Sebagai gambaran struktur *database* misalnya *database* gaji guru, pada contoh *database* ini administrator dipermudah untuk mengelola data dan melaksanakan tugasnya untuk menggaji guru. Harap diperhatikan bahwa *scope* (cakupan) *database* yang akan kita buat ini sudah dibatasi, yaitu untuk memenuhi kebutuhan dari pengelolaan gaji guru saja. Tujuan pembatasan cakupan *database* tersebut agar terjadi fokus dan mengetahui data apa yang perlu disimpan. Di kemudian hari, *database* bisa saja kita kembangkan lagi untuk mengakomodasi fungsi lain, atau sebaliknya dipecah menjadi beberapa *database*.

Dalam hal pelaksanaan tugas administrator untuk pengelolaan gaji guru, kita mendapatkan fakta sebagai berikut:

1. Tersedianya daftar guru, berarti harus ada data guru, karena siapa yang akan digaji.
2. Tersedianya daftar mata pelajaran, berarti harus ada data mata pelajaran, karena gaji guru disesuaikan dengan jumlah jam mengajarnya, semakin banyak jam mengajar semakin banyak hasil gaji yang didapat.
3. Tersedianya daftar gaji mengajar, berarti harus ada data gaji mengajar sebagai hasil guru mengajar. Di dalam gaji mengajar terdapat beberapa biaya tunjangan lain.

Untuk itu administrator akan meng-*input* data guru, data mata pelajaran, dan data gaji mengajar. Dan akan diproses pada *query* untuk menghasilkan keluaran berupa gaji yang didapat oleh guru yang bersangkutan. Berdasarkan fakta di atas, kita bisa memulai menentukan struktur *database* berupa daftar tabel yang perlu dibuat yaitu:

1. Tabel data guru
2. Tabel data mata pelajaran
3. Tabel gaji mengajar



Pada kenyataannya, dalam penyusunan *database* kita masih dihadapkan pada ketentuan-ketentuan atau kaidah *database*. Salah satu kaidahnya yang umum dan harus dipatuhi adalah:

“Tidak boleh mencatat data yang sama lebih dari satu kali”

Pelanggaran terhadap aturan ini akan menghasilkan akibat:

1. Pemborosan data (*redundancy*).
2. Data yang tidak konsisten (*inconsistency*).

Demikianlah proses kita memilih dan memilah data yang akan disimpan dalam *database*.

DATA GURU						
No	NIP	Nama	Tgl_Lahir	Jk_Kelamin	Alamat	Telepon
1	1543-2-0001	SITI JUBAIDAH, S.Pd	20/08/1948	PEREMPUAN	PULAU RAYA	(0688)982-823-22
4	1955-2-0011	Dra. REZKY NUR	09/09/1956	PEREMPUAN	KINTANI	(0982)716-152-32
3	1972-1-0018	AKHMAD CARQONI, S.Pd	09/08/1972	LAKI-LAKI	PEABUHAN	(0899)282-334-32
6	1988-2-0022	MUHAMMAD GIO, S.Pd	09/07/1988	LAKI-LAKI	BATU SIMPANG	(0582)726-625-32
2	1989-1-0021	ADI MAULANA, A.Md	03/11/1989	LAKI-LAKI	MALAWAI	(0898)232-332-35
5	1990-1-0026	DARMAWAN, A.Md	05/05/1990	PEREMPUAN	HALULUNG	(0583)233-664-48

DATA MATA PELAJARAN			
Nc	Kode_Mape	Nama_Mapel	Jl_Jam
1	NN-011-01	MATEMATIKA	4
2	NN-012-02	B. INDO	3
3	NN-013-03	B. ING	5
4	NN-014-04	IPS	2
5	NN-015-05	IPA	3
6	NN-016-06	PKn	4

GAJI MENGAJAR							
No	Kode_Gaj	NIP	Kode_Mapel	Gaji_Perjar	Gaji_Pokok	Tj_Kesehata	Asuransi
1	1-GM-01	1948-2-0001	NN-012-02	Rp20.000	Rp1.000.000	Rp50.000	Rp70.000
2	1-GM-02	1955-2-0011	NN-011-01	Rp20.000	Rp890.000	Rp50.000	Rp70.000
3	1-GM-03	1972-1-0018	NN-013-03	Rp20.000	Rp870.000	Rp50.000	Rp70.000
4	1-GM-04	1989-1-0021	NN-015-05	Rp20.000	Rp840.000	Rp50.000	Rp70.000
5	1-GM-05	1990-1-0026	NN-016-06	Rp20.000	Rp800.000	Rp50.000	Rp70.000
6	1-GM-06	1988-2-0022	NN-014-04	Rp20.000	Rp840.000	Rp50.000	Rp70.000

Gambar 4.18 Database Gaji Guru



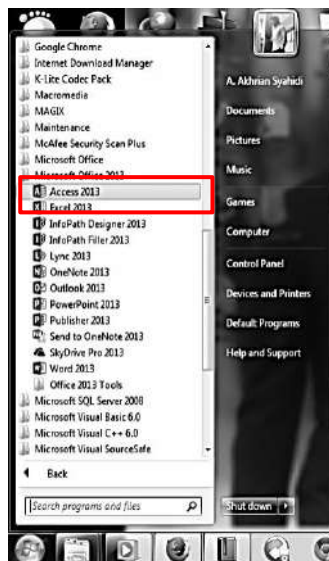
GAJI KOTOR							
NIP	Nama	Jl_Jam	Gaji_Perjari	Gaji_Pokok	Tj_Kesehata	Asuransi	GAJI KOTOR
1948-2-0001	SITI JUBAIDAH, S.Pd	3	Rp20.000	Rp1.000.000	Rp50.000	Rp70.000	Rp1.180.000
1956-2-0011	Dra. REZKY NUR	4	Rp20.000	Rp890.000	Rp50.000	Rp70.000	Rp1.090.000
1972-1-0018	AKHMAD CARQONI, S.Pd	5	Rp20.000	Rp870.000	Rp50.000	Rp70.000	Rp1.090.000
1989-1-0021	ADI MAULANA, A.Md	3	Rp20.000	Rp840.000	Rp50.000	Rp70.000	Rp1.020.000
1990-1-0026	DARMAWAN, A.Md	4	Rp20.000	Rp800.000	Rp50.000	Rp70.000	Rp1.000.000
1988-2-0022	MUHAMMAD GIO, S.Pd	2	Rp20.000	Rp840.000	Rp50.000	Rp70.000	Rp1.000.000

Gambar 4.19 Tampilan Query Gaji Kotor sebagai Hasil Database Gaji Guru

4.7 Memulai Bekerja di dalam Microsoft Office Access 2013

Langkah-langkah untuk menjalankan Microsoft Office Access 2013 sebagai berikut:

1. Pastikan PC/Laptop Anda telah terinstal Microsoft Office Access 2013.
2. Pilih **Start** → **All Programs** → **Microsoft Office 2013** → **Microsoft Office Access 2013** pada menu yang tersedia dalam windows.

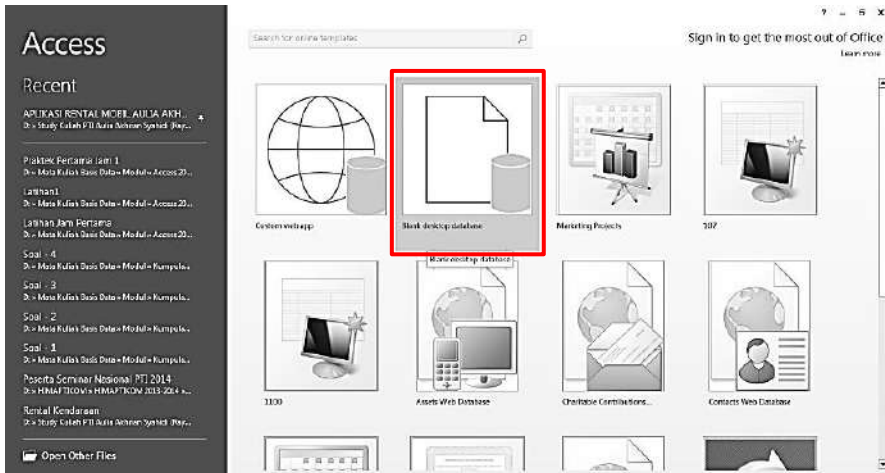


Gambar 4.20 Area Menu Microsoft Office



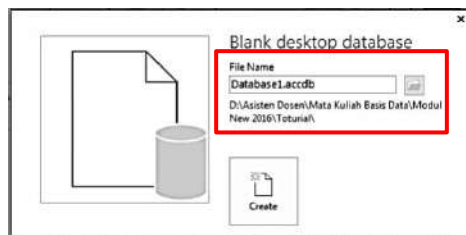
4.8 Membuat Database

1. Tampilan awal Microsoft Office Access 2013




Gambar 4.20 Tampilan Awal Microsoft Office Access 2013

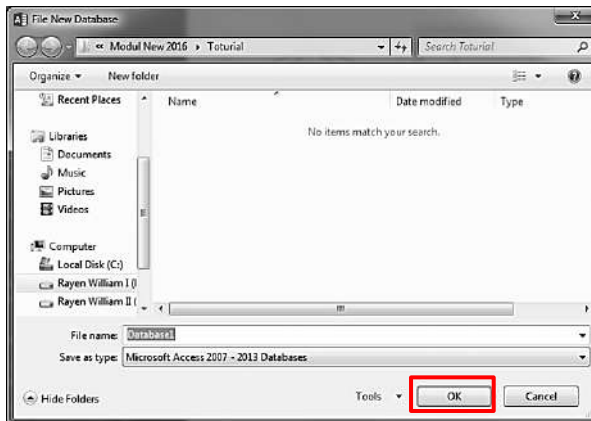
2. Pada tampilan awal tadi pilih dan klik **Blank desktop database**, dimana jika Anda membuat sebuah *database* baru yang masih kosong.
3. Ketik nama *file database* yang akan dibuat pada bagian **File Name**, secara default Microsoft Office Access 2013 akan memberi nama *file database* dengan **Database1.accdb**.




Gambar 4.21 Kotak Dialog **Blank desktop database**

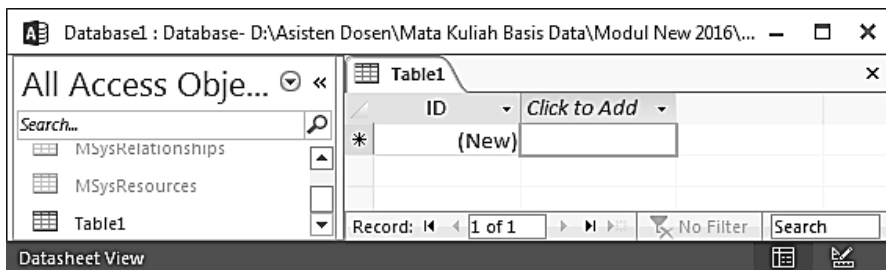


4. Kemudian klik tombol **Browse**  yang ada di sebelah kanan **File Name** untuk menentukan lokasi penyimpanan *file*, sehingga akan tampil kotak dialog **File New Database**.



Gambar 4.22 Kotak Dialog **File New Database**

5. Tentukan lokasi *drive* dan *folder* tempat penyimpanan *file* → klik **OK**.
6. Klik tombol **Create**  sehingga akan tampil jendela **Database Microsoft Office Access 2013**.




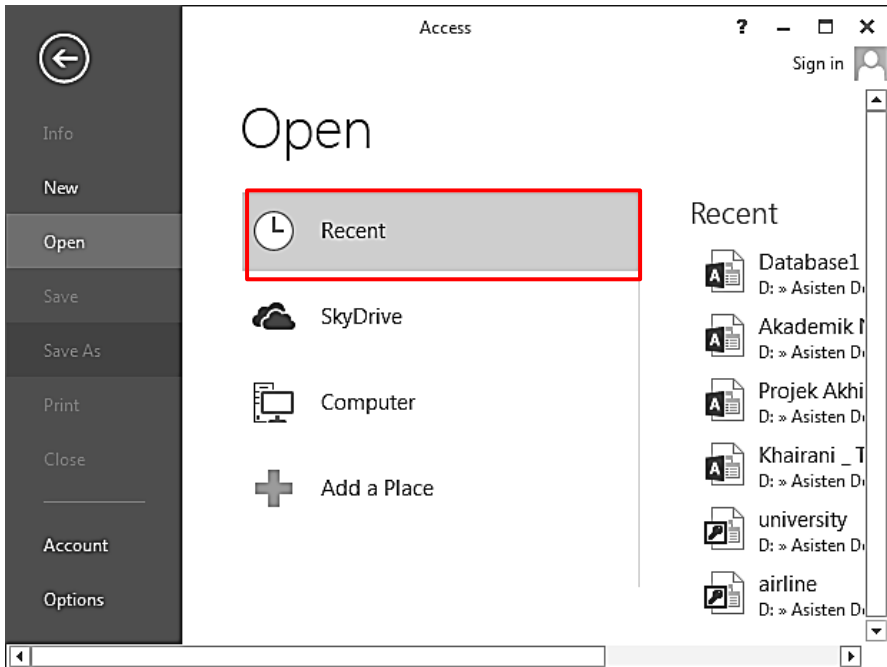
Gambar 4.23 Tampilan Jendela **Database**



4.9 Membuka Database

Untuk membuka *file database* dapat Anda lakukan pada tampilan awal program Microsoft Office Access 2013, yaitu dengan memilih nama *file* yang akan dibuka pada bagian **Recent**, atau pilihan **Open Other Files** untuk mencari *file database* yang lain. Selain itu Anda dapat menggunakan langkah berikut ini untuk membuka *file database*:

1. Klik menu **File**  → pilih **Open** atau tekan tombol **Ctrl + O** pada *keyboard*. Sehingga muncul **Backstage Open** seperti gambar 4.24.

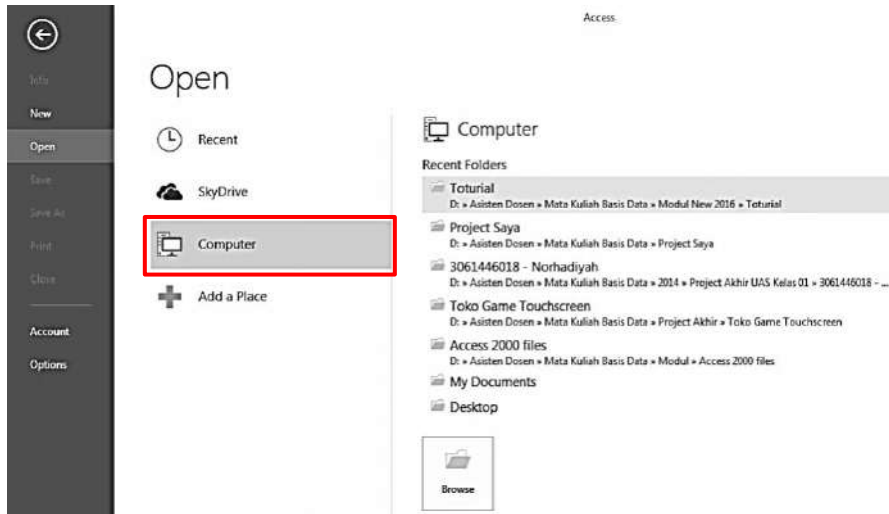


Gambar 4.24 Tampilan Backstage Open – Recent

2. Pilih **Recent** jika ingin membuka *file database* yang pernah dibuka sebelumnya yang dapat Anda pilih pada bagian sebelah kanan.

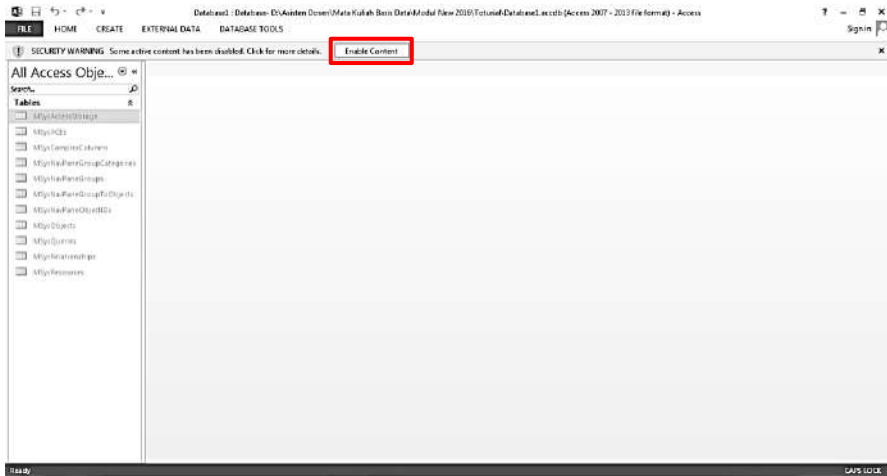


3. Jika ingin membuka *file* yang lain pilih **Computer** pada **Backstage Open**, kemudian klik tombol **Browse** untuk menentukan lokasi *file* dan nama *file database* yang ingin dibuka. Kemudian klik tombol **Open** atau tekan **Enter** untuk membukanya.



Gambar 4.25 Tampilan Backstage Open – Computer

4. Perlu diperhatikan ketika Anda membuka *file database* biasanya muncul **Security Warning**, pilih dan klik saja **Enable Content**.

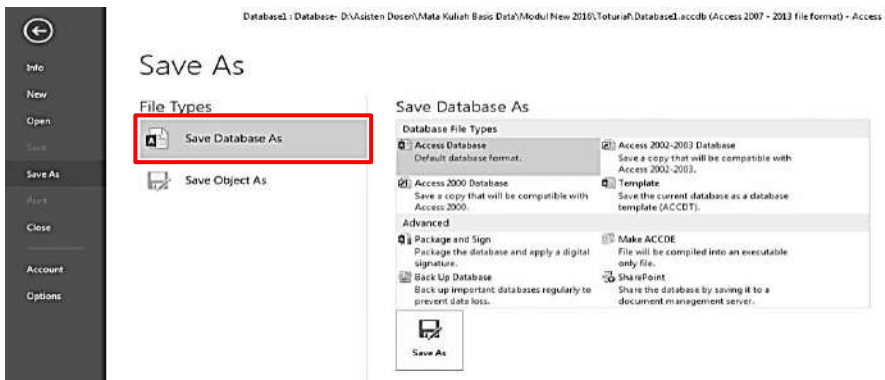


Gambar 4.26 Area Database1

4.10 Menyimpan Database

Langkah untuk menyimpan *file database* adalah sebagai berikut:

1. Tekan tombol **Ctrl + S** pada *keyboard* atau klik menu **File** dan pilih **Save** atau **Save As**.
2. Sehingga akan tampil **Backstage Save As** seperti gambar 4.27.



Gambar 4.27 Tampilan Backstage Save As



3. Jika pada saat menyimpan Anda tidak mengganti tipe *file*, maka secara *default* akan tersimpan dengan format *database* Microsoft Office Access 2007-2013 (*.accdb). Di mana *file* tersebut hanya dapat dioperasikan pada program Microsoft Office Access 2007 sampai 2013 saja, tidak untuk versi sebelumnya seperti Microsoft Office Access 2000 atau 2003 (*.mdb). Tetapi jika Anda ingin menyimpan *file database* dengan format Microsoft Office Access versi sebelumnya atau yang lainnya atau sebagai objek lain, dapat Anda lakukan dengan cara memilih tipe *file* pada bagian **File Types**.

Save Database As, jika disimpan dalam format *file database* lain.

- **Access Database**, sebagai *file database* Microsoft Office Access default.
- **Access 2000 Database**, sebagai *file database* Microsoft Office Access 2000.
- **Access 2002-2003 Database**, sebagai *file database* Microsoft Office Access 2002-2003.
- **Template**, sebagai *file template database* Microsoft Office Access.

Save Object As, jika ingin disimpan dalam bentuk objek lain.

- **Save Object As**, sebagai objek *database* lain.
- **PDF or XPS**, sebagai *file* PDF atau XPS.

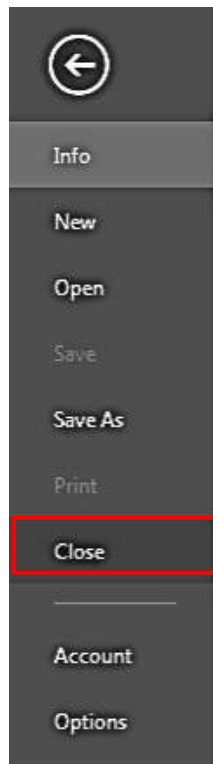


4. Klik tombol **Save As** sehingga akan ditampilkan kotak dialog **Save As**.
5. Tentukan *drive* dan *folder* tempat menyimpan *file*, kemudian ketik nama *file database* pada bagian **File Name**.
6. Klik tombol **Save** untuk menyimpan *file*.



4.11 Menutup Database

Langkah untuk menutup *file database* yaitu klik menu **File** dan pilih **Close**. Selanjutnya jendela *database* akan ditutup dan jendela Microsoft Office Access 2013 akan kembali kosong.



Gambar 4.28 Tampilan *Backstage Close*

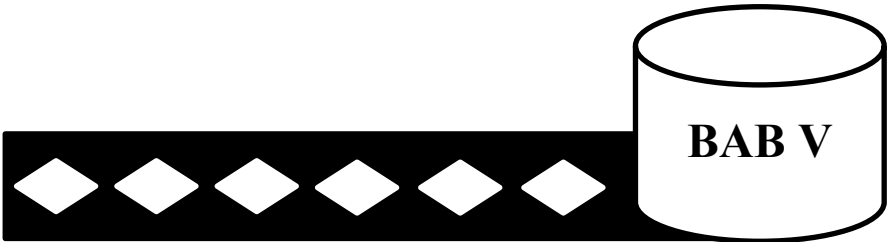
Latihan Praktik

1. Buat *folder* baru di dalam *Local D/E* penyimpanan PC/Laptop Anda, berikan nama *folder* **Latihan Basis Data [Nama Anda-NPM]**, nanti setiap Anda praktikum datanya terus menerus disimpan dalam *folder* tersebut.



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

2. *Print screen* tahapan memulai Microsoft Office Access 2013 sampai dengan pembuatan *database*.
3. Buat *database* baru dengan nama: **Sistem Informasi Kepegawaian** dan simpan *filenya* dalam *folder Latihan Basis Data [Nama Anda-NPM]* yang telah dibuat sebelumnya.
4. Setelah dibuat *database Sistem Informasi Kepegawaian* kemudian *close database*.



MEMBUAT DAN MEMODIFIKASI *TABLE*



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu merancang dan membuat *Table*
2. Mahasiswa mampu memahami fungsi *Primary Key* dan *Foreign Key* pada *Table*
3. Mahasiswa mampu memodifikasi *Table*
4. Mahasiswa mampu mengelola *Table*



5.1 Merancang Tabel

Tabel atau *table* adalah objek *database* yang menyimpan data. Untuk memulai membuat *tabel* Anda bisa melakukannya pada *file database* mana saja dan tidak perlu khawatir, karena tabel yang sudah Anda buat dapat di-*import* ke *file database* yang lain jika diperlukan. Selain menjadi objek utama *database*, sebuah tabel bisa dipandang sebagai kumpulan data tentang topik yang khusus, seperti data karyawan, data penyewaan, dan lain-lain. Dengan menggunakan tabel terpisah untuk masing-masing topik berarti Anda sudah melakukan suatu tindakan yang amat berguna bagi *database*. Sebab dengan demikian akan mengurangi beberapa kesalahan terutama di dalam memasukkan (*entry*) data. Tabel terdiri dari dua bagian utama yaitu:

1. **Kolom** yaitu bagian yang **menurun/vertikal** atau lebih dikenal dengan istilah **Field**. **Field** adalah **item data yang ditunjukkan dalam bentuk kolom**.
2. **Baris** yaitu bagian **mendatar/horizontal** atau sering disebut **Record**. **Record** yaitu **sebaris data yang mencakup field dalam sebuah tabel**.

No	Kode_DVD	Judul_DVD	Harga_Rent	Stock
1	AAA-001	AYAT-AYAT CINTA	Rp25.000	10
2	AAB-002	1001 MALAM	Rp45.000	5
3	AAC-003	SANG PEMBERANI	Rp35.000	7
4	AAD-004	HABIBI AINUN	Rp78.000	9
5	AAE-005	TRANSFORMER	Rp99.000	8

Gambar 5.1 *Field dan Record* dalam Sebuah *Table*

Sebelum membuat tabel, terlebih dahulu Anda perlu menyusun *field* yang akan Anda masukkan ke dalam tabel. Inilah yang membedakan antara pemakaian *database* dengan *spreadsheet* di mana Anda bisa langsung memasukkan data pada kolom-kolom yang tersedia. Struktur



tabel, atau susunan *field*, bisa Anda rancang dengan bantuan *spreadsheet*. Jika tersedia data (yang telah digunakan selama ini) akan lebih baik lagi karena Anda bisa menentukan tipe datanya dan seberapa besar ukuran datanya.

Sebagai contoh, berikut ini Anda akan membuat tabel untuk menyimpan data karyawan beserta unsur gajinya. *Item* data yang Anda peroleh dari data karyawan beserta unsur gajinya sebagai berikut:

- Nomor Induk Karyawan
- Nama Karyawan
- Alamat
- *Handphone*
- Jenis Kelamin
- Agama
- Tempat Lahir
- Tanggal Lahir
- Pendidikan
- Tanggal Mulai Bergabung
- Golongan
- Skala
- Departemen
- Gaji Pokok
- Tunjangan Transportasi
- Tunjangan Kesehatan
- Tunjangan Perumahan
- Tunjangan Telekomunikasi
- Tunjangan Makan

Dari *item* data di atas dapat Anda lihat bahwa sebaiknya data yang berkaitan dengan gaji (mulai gaji pokok dan seterusnya) dipisahkan pada tabel sendiri karena jumlah *item* datanya cukup banyak dan tidak selalu digunakan bersama data karyawan lainnya. Selain itu, nama departemen sebaiknya dipisah juga karena kemungkinan banyak karyawan yang termasuk dalam departemen yang sama (**ingat: data yang sama/berulang-**



ulang sebaiknya dipisah ke dalam tabel sendiri agar tidak terjadi redundancy).

Berdasarkan pertimbangan di atas, Anda akan membuat tiga tabel yaitu:

1. Tabel Karyawan
2. Tabel Gaji
3. Tabel Departemen

Struktur data untuk tabel Karyawan:

- Nomor Induk Karyawan
- Nama Karyawan
- Alamat
- Handphone
- Jenis Kelamin
- Agama
- Tempat Lahir
- Tanggal Lahir
- Pendidikan
- Tanggal Mulai Bergabung
- Golongan
- Skala
- Departemen

Struktur data untuk tabel Gaji:

- Nomor Induk Karyawan
- Gaji Pokok
- Tunjangan Transportasi
- Tunjangan Kesehatan
- Tunjangan Perumahan
- Tunjangan Telekomunikasi
- Tunjangan Makan

Perhatikan untuk data **Nomor Induk Karyawan** digunakan lagi pada tabel Gaji karena data inilah yang menjadi kunci untuk



menghubungkan kedua tabel tersebut di atas. Di sini seorang karyawan hanya mempunyai sebuah *record* di tabel Gaji.

Struktur data untuk tabel Departemen:

- Kode Departemen
- Nama Departemen
- Lokasi Kerja

Untuk data **Kode Departemen** akan diisikan pada tabel Karyawan (dengan memakai data Departemen sebagai penghubung), sehingga data ini menjadi kunci untuk menghubungkan kedua tabel tersebut (tabel Karyawan dan tabel Departemen).

Tata cara memisahkan tabel dan menghubungkannya dengan kunci penghubung inilah yang disebut *database* relasional.

5.2 Mengelola Jendela *Design View*

Jika Anda membuat tabel dengan *Design View*, akan ditampilkan dua buah bidang atau *pane*, yaitu sebagai berikut:

1. *Pane* bagian atas digunakan untuk meng-*input field*, yang berisi kolom **Field Name**, **Data Type**, dan **Description**.
2. *Pane* bagian bawah berisi **Field Properties** yang aktif, digunakan untuk mengatur tatanan *field*.

Sedangkan untuk berpindah antar *pane*, gunakan tombol **F6** atau klik bagian *pane* tersebut.



dahulu berbagai tipe data yang tersedia pada Microsoft Office Access 2013 sebagai berikut:

Tabel 5.1 Tipe Data

Tipe Data	Keterangan
<i>Short Text</i> (versi sebelumnya <i>Text</i>)	Menyimpan teks (huruf dan angka) dengan panjang maksimum 255 karakter.
<i>Long Text</i> (versi sebelumnya <i>Memo</i>)	Menyimpan teks yang panjangnya melebihi 255 karakter. Panjang <i>field</i> maksimum 65.535 karakter (manual) dan 2 GB (program)
<i>Number</i>	Menyimpan data berupa angka yang bisa dikalkulasi, namun bukan merupakan data keuangan. Tipe data <i>Number</i> bisa berupa: <i>Byte</i> , <i>Integer</i> , <i>Long Integer</i> , <i>Single</i> , <i>Double</i> , <i>Decimal</i> , dan <i>Replication ID</i> .
<i>Currency</i>	Menyimpan data yang berjenis angka dengan format mata uang agar tidak terjadi pembulatan ketika dikalkulasi.
<i>Auto Number</i>	Menghasilkan angka otomatis (selalu meningkat atau acak) ketika <i>record</i> diciptakan. <i>Auto Number</i> secara default berupa <i>Long Integer</i> , namun bisa juga berupa <i>Replication ID</i> .
<i>Date/Time</i>	Menyimpan data waktu yaitu hari, tanggal, jam, dan menit.
<i>Yes/No</i>	Menyimpan data boolean misalnya <i>Yes/No</i> , <i>True/False</i> , dan <i>On/Off</i> .
<i>Hyperlink</i>	Menyimpan data berupa <i>hyperlink</i> yaitu <i>path UNC (Universal Naming Convention)</i> dan <i>URL (Uniform Resource Location)</i> dengan panjang maksimum 64.000 karakter.
<i>OLE Object</i>	Menyimpan objek misalnya dokumen, gambar, suara yang dibuat dari program lain menggunakan protokol <i>OLE (Object Linking and Embedding)</i> menyimpan hingga 1 GB.
<i>Lookup Wizard</i>	Membuat <i>lookup</i> di mana <i>user</i> bisa memilih data dari <i>tabel</i> lain atau <i>query</i> menggunakan fasilitas <i>Combo Box</i> . <i>Lookup Wizard</i> bukanlah tipe data yang sebenarnya karena tipe data hasil <i>wizard</i> disesuaikan dengan tipe data darimana data berasal.
<i>Attachment</i>	Merupakan tipe data untuk <i>field</i> yang akan menyimpan <i>file-file attachment</i> (lampiran).



Tippe Data	Keterangan
Calculated	Membuat <i>field</i> yang merupakan hasil kalkulasi dari <i>field</i> lainnya.

5. **Description/Keterangan**

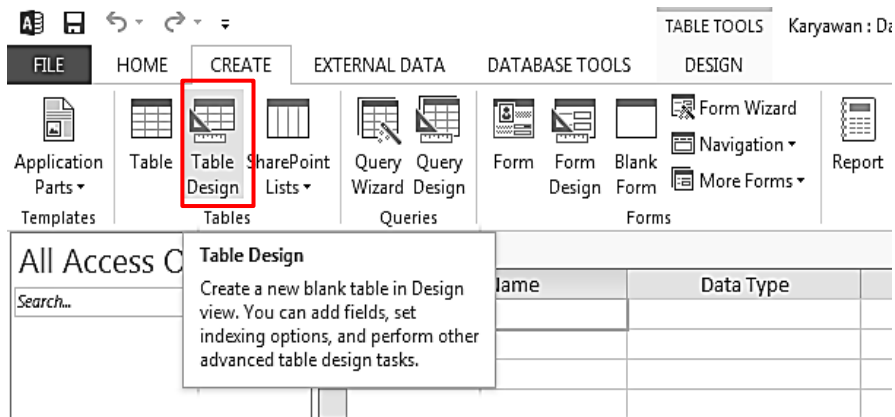
Untuk menampilkan teks keterangan pada bagian *Status Bar* dalam jendela *Datasheet* sebuah tabel/*query*. Batas maksimal jumlah teks yang dapat digunakan adalah 255 karakter, untuk *Description* ini bersifat opsional saja, untuk memberikan keterangan dari *field*.

6. **Field Properties/Properti Kolom**

Untuk mengatur beberapa ketentuan dari suatu *field* yang aktif, dan ketentuan yang ditampilkan disesuaikan dengan tipe data dari *field* tersebut.

5.3 **Membuat Tabel**

Sekarang mari Anda membuat tabel Karyawan menggunakan **Table Design**. Klik *ribbon CREATE* → **Table Design**. Akan muncul tampilan sebagai berikut:



Gambar 5.2 Membuat Tabel dengan *Table Design*

Masukkanlah definisi *field* sebagai berikut lengkap dengan tipe data dan ukurannya. Untuk *field* berjenis Tanggal (*Data/Time*) Anda tidak

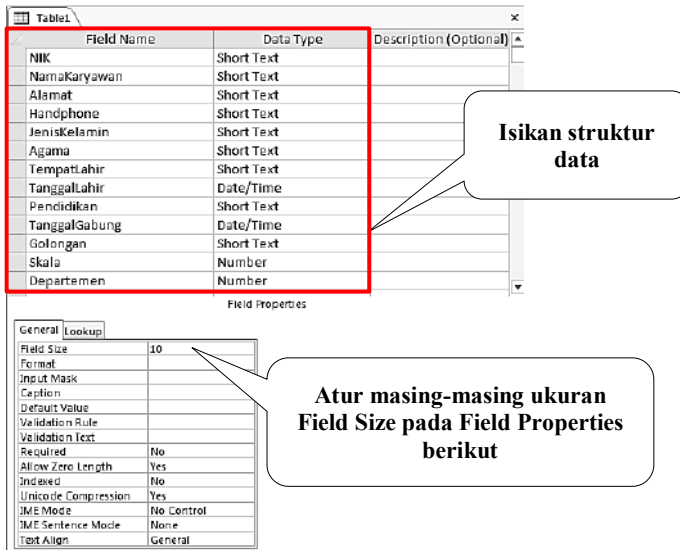


perlu menentukan ukurannya karena sudah ditentukan secara *default* oleh Microsoft Office Access 2013.

Tabel 5.2 Tabel Karyawan

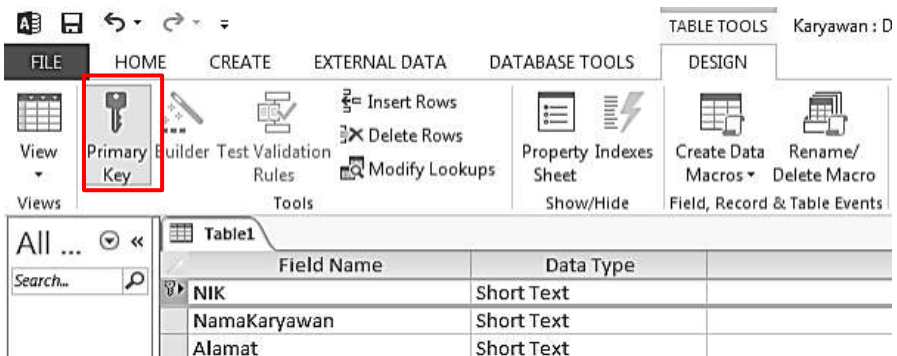
Nama <i>Field</i>	Tipe Data	Ukuran (<i>Field Size</i>)
NIK	<i>Short Text</i>	10
NamaKaryawan	<i>Short Text</i>	50
Alamat	<i>Short Text</i>	255
Handphone	<i>Short Text</i>	20
JenisKelamin	<i>Short Text</i>	1
Agama	<i>Short Text</i>	15
TempatLahir	<i>Short Text</i>	50
TanggalLahir	<i>Date/Time</i>	
Pendidikan	<i>Short Text</i>	10
TanggalGabung	<i>Date/Time</i>	
Golongan	<i>Short Text</i>	10
Skala	<i>Number</i>	<i>Integer</i>
Departemen	<i>Number</i>	<i>Long Integer</i>

Perhatikanlah pada struktur data di atas Anda tidak menggunakan spasi pada nama *field*, kalau ingin ada spasi ganti dengan ***underscore*** () dan menggunakan huruf besar (kapital) pada setiap awal kata. Untuk menentukan *field size*, lakukan pada pengaturan *Field Properties* pada gambar 5.3 berikut:



Gambar 5.3 Menentukan Ukuran *Field*

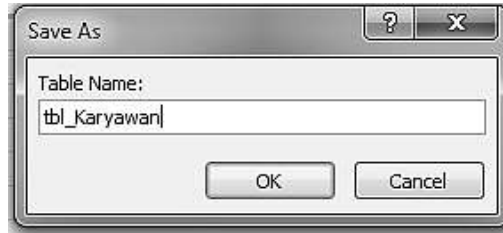
Setelah selesai memasukkan struktur data dan *field*, klik *field* NIK → klik *Ribbon Design* → klik *Primary Key* untuk menjadikan NIK sebagai *Primary Key*.



Gambar 5.4 Menentukan *Primary Key*



Simpan tabel dengan mengklik *Save* atau **Ctrl+S**. Beri nama tabel dengan **tbl_Karyawan** → **OK**.



Gambar 5.5 Menentukan Ukuran *Field*

Jika sebelum menentukan *Primary Key* Anda sudah melakukan penyimpanan tabel, maka akan muncul peringatan dari Microsoft Office Access 2013:



Gambar 5.6 Peringatan Karena Tabel Tidak Mempunyai *Primary Key*

Seharusnya pada peringatan ini Anda menjawab *No* atau mengklik *Cancel*. Namun jika Anda menjawab *Yes* maka Microsoft Office Access 2013 akan otomatis menambahkan sebuah *field* bernama **ID** sebagai *Primary Key*.



	Field Name	Data Type
🔑	ID	AutoNumber
	NIK	Short Text
	NamaKaryawan	Short Text
	Alamat	Short Text
	Handphone	Short Text

Gambar 5.7 *Field ID* Tambahan Microsoft Office Access 2013 Sebagai *Primary Key*

Anda bisa membuang *field* ID tambahan Microsoft Office Access 2013 dengan cara menyeleksinya (klik *field selector* di ujung kiri) lalu tekan **Delete**. Klik *field* NIK → klik **Ribbon Design** → klik **Primary Key** untuk menjadikan NIK sebagai *Primary Key*. Berikut ini adalah beberapa hal yang sering terjadi dan perlu diketahui ketika mendesain tabel, yaitu:

1. Jika ada *field* yang terlewat (tidak dimasukkan), Anda bisa menyisipkan *field* tersebut.
2. Letak *field* bisa dipindahkan. Caranya: klik dan seret (*drag*) *field selector* ke posisi yang diinginkan.
3. Anda bisa meng-*copy-paste field* dari/ke tabel yang lain.
4. Jika tidak mempunyai *field* untuk *Primary Key* atau tidak ingin memaksa pemakai mengisi data untuk *Primary Key*, Anda bisa menggunakan *field Auto Number*, yaitu seperti *field ID* yang ditambahkan oleh Microsoft Office Access 2013. *Field* bertipe *Auto Number* akan otomatis terisi angka yang terus meningkat sehingga tidak mungkin sama.

5.4 Fungsi **Primary Key** dan **Foreign Key**

Pada struktur data Karyawan yang telah dibahas tadi Anda menggunakan *field Nomor Induk Karyawan* sebagai identitas data karyawan. Setiap karyawan mempunyai Nomor Induk Karyawan yang berbeda (tidak mungkin sama atau unik). Nomor Induk Karyawan ini



disebut **Primary Key** (kunci primer/utama) karena mewakili *record* masing-masing karyawan. Jadi **Primary Key** adalah sebuah *field* yang bisa menjadi identitas *record* karena berbeda (unik) untuk setiap *record*-nya.

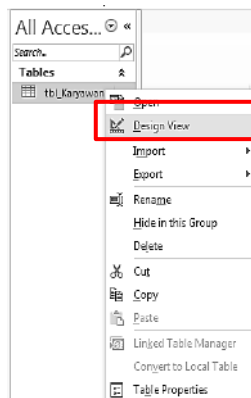
Catatan: *Primary Key* bisa juga terdiri dari gabungan beberapa *field* asalkan gabungan *field* tersebut membentuk nilai yang unik sehingga bisa menjadi identitas *record*-nya.

Pada struktur data **Gaji** Anda juga menggunakan *field* **Nomor Induk Karyawan** sebagai *Primary Key* atau identitas *record* pada tabel tersebut. Pada struktur data **Departemen**, Anda menggunakan *field* **Kode Departemen** sebagai *Primary Key* atau identitas departemen. Kode Departemen diisikan pada *field* Departemen pada tabel **Karyawan** (untuk menghubungkannya). Oleh karena mendapatkan data dari *Primary Key* suatu tabel, maka *field* **Departemen** pada tabel **Karyawan** disebut **Foreign Key** (atau **Secondary Key**) bagi tabel **Departemen**. Jadi **Foreign Key** adalah *field* yang menghubungkan tabel dengan *Primary Key* pada tabel lain.

5.5 Memodifikasi Tabel

5.5.1 Langkah Awal Memodifikasi Tabel

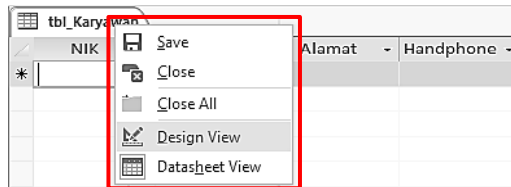
Untuk memodifikasi tabel yang sudah Anda buat dengan cara: pada *Navigation Pane* klik kanan tabel → pilih dan klik **Design View**.



Gambar 5.8 Menampilkan Tabel dalam *Design View*



Cara lain adalah melakukan klik kanan pada *document tab* dari tabel yang sedang aktif lalu pilih **Design View**.



Gambar 5.9 Menampilkan *Design View* Tabel yang Sedang Aktif

Beberapa hal yang perlu diketahui di dalam memodifikasi tabel sebagai berikut:

1. Anda bisa menambahkan *field* baru, menghapus *field*, atau mengubah nama *field*, tipe data, atau properti *field* lainnya.
2. Jika mengubah nama *field* sementara nama *field* sudah digunakan pada *query*, *form*, atau *report*, maka Microsoft Office Access 2013 akan otomatis menyesuaikan nama *field* tersebut dengan nama yang baru disebut **Name AutoCorrect**.
3. Jika mengubah tipe data sementara tabel sudah mengandung data, Microsoft Office Access 2013 akan melakukan konversi data yang ada ke tipe yang baru. Kemungkinan bisa berhasil atau gagal yang berarti data pada *field* tersebut hilang. Proses konversi juga bisa memakan waktu yang sangat lama jika pada *field* tersebut terdapat *validation rule*.

5.5.2 Mengganti Nama *Field*

Setelah merancang tabel terkadang ada suatu perubahan yang mengharuskan modifikasi pada tabel yang dibuat. Misalnya mengganti *field*, menambahkan *field*, atau menyisipkan *field*, ataupun menghapus *field* tersebut.

Cara memperbaiki/mengganti nama *field* dengan cara: aktifkan dulu bagian **Design View** tabel yang diinginkan (misalnya **tbl_Karyawan**)



→ klik *field* yang akan diganti namanya (misalnya *field* **Alamat**) → kemudian ketikkan nama *field* yang baru misalnya **AlamatLengkap**.

Field Name	Data Type
NIK	Short Text
NamaKaryawan	Short Text
AlamatLengkap	Short Text
Handphone	Short Text
JenisKelamin	Short Text
Agama	Short Text
TempatLahir	Short Text
TanggalLahir	Date/Time
Pendidikan	Short Text
TanggalGabung	Date/Time
Golongan	Short Text
Skala	Number
Departemen	Number

Gambar 5.10 Mengganti Nama *Field*

5.5.3 Mengganti Jenis Data (*Data Type*)

Cara memperbaiki/mengganti jenis data dengan cara: aktifkan dulu bagian *Design View* tabel yang diinginkan (misalnya **tbl_Karyawan**) → klik *field* **Handphone** → pada bagian *Data Type* ganti yang sebelumnya adalah **Short Text** menjadi **Number**.

Field Name	Data Type
NIK	Short Text
NamaKaryawan	Short Text
AlamatLengkap	Short Text
Handphone	Short Text
JenisKelamin	Short Text
Agama	Long Text
TempatLahir	Number
TanggalLahir	Date/Time
Pendidikan	Currency
TanggalGabung	AutoNumber
Golongan	Yes/No
Skala	OLE Object
Departemen	Hyperlink

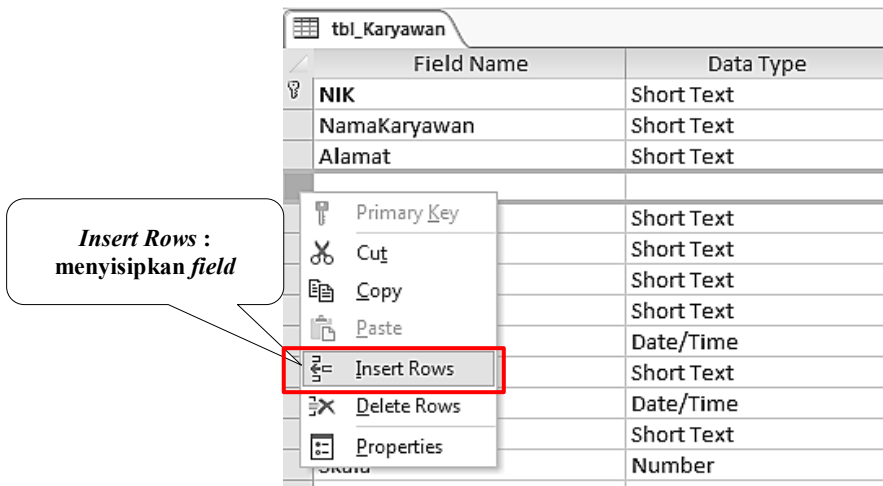
General Lookup

Gambar 5.11 Mengganti *Data Type*



5.5.4 Menyisipkan dan Menambahkan *Field*

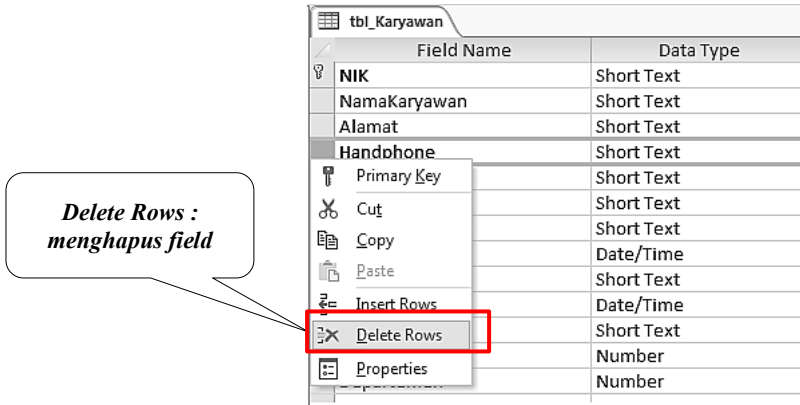
Untuk menyisipkan atau menambahkan *field* dengan cara: aktifkan dulu bagian **Design View** tabel yang diinginkan (misalnya **tbl_Karyawan**) → klik *field selector* (sebelah kiri nama *field*) → lalu tekan tombol **Insert Rows**.



Gambar 5.12 Menyisipkan *Field*

5.5.5 Menghapus *Field*

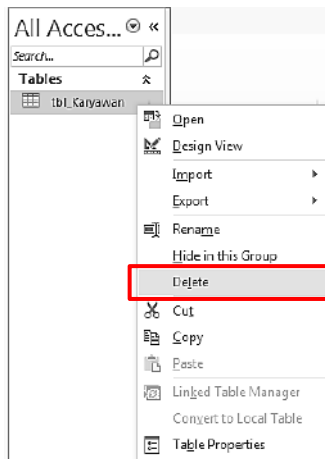
Cara menghapus *field* dengan cara: aktifkan dulu bagian **Design View** tabel yang diinginkan (misalnya **tbl_Karyawan**) → klik kanan *field* yang ingin di hapus misalnya *field Handphone* → klik **Delete Rows**.



Gambar 5.13 Menghapus Field

5.5.6 Menghapus Tabel dari Database

Tabel yang sudah tidak diperlukan lagi bisa dihapus. Caranya: pada *Navigation Pane* klik kanan tabel → pilih dan klik *Delete*.



Gambar 5.14 Menghapus Tabel

Perhatian: Tabel yang sudah dihapus tidak bisa dikembalikan lagi.



5.6 Mengelola Tabel

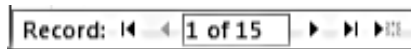
5.6.1 Meng-input Data dalam Tabel

Untuk meng-input data di dalam tabel dengan cara: klik *double* tabel (misalnya **tbl_Karyawan**). Kemudian tambahkan datanya seperti di bawah ini.

NIP	Nama Karyawan	Alamat/Leag	Handphone	Jenis Kel	Agama	Tempat Lahi	Tanggal Lahi	Pendidikan	Tanggal Kabi	Golongan	Skala	Departemen
K01	Akhrad Ali Rizaldi	Jl. Simpang 02	878143006	L	Islam	Surabaya	28/02/1990	S1	29/01/2000	IIIA	1	1
K02	Susanita Agranea	Jl. Kebon Jenuh	982919111	P	Islam	Bandung	19/09/1990	S1	01/01/2012	IIIA	1	2
K03	I Made Idha Nuanlara	Jl. Kebon Sirih	992871722	P	Hindu	Bali	06/06/1989	S1	08/08/2010	IIIA	1	1
K04	Albert Sitohang	Jl. Siliwangi Jal	828919189	L	Kristen	Medan	09/09/1988	S1	08/08/2010	IIIA	1	2
K05	I Nyoman Windu Asmara	Jl. Didina 9 Jak	928282827	L	Hindu	Bali	09/08/1989	S1	09/01/2007	IIIA	2	1
K06	Lee Xing Wee	Jl. Gattot Subro	918181818	L	Budha	Korea	16/08/1987	S1	09/01/2008	IIIA	2	3
K07	Rizki Azrian Gerandho	Jl. Lingkar Utara	829191982	L	Islam	Banyuwangi	16/11/1982	S1	28/01/2000	IIIA	2	3
*			0								0	0

Gambar 5.15 Menambah/Mengisi Data

Untuk memindah posisi *record* aktif dengan memilih tombol *record navigator* yang ada pada bagian bawah jendela *Datasheet* dari sebuah tabel.



Gambar 5.16 Record Navigator

Keterangan gambar 5.16 sebagai berikut:

- *First Record* berfungsi untuk menuju ke awal *record*.
- *Last Record* berfungsi untuk menuju ke akhir *record*.
- *Previous Record* berfungsi untuk menuju ke *record* sebelumnya.
- *Next Record* berfungsi untuk menuju ke *record* sesudahnya.
- *New Record* berfungsi untuk menuju ke *record* baru.
- **1 of 15** atau tekan F5, ketik nomor *record* dan tekan *enter* berfungsi untuk menuju ke *record* tertentu.

5.6.2 Mengedit atau Mengubah Data



Untuk mengedit/mengubah data di dalam tabel dengan cara: klik *double* tabel (misalnya **tbl_Karyawan**) → arahkan kursor ke arah *record* yang akan dirubah datanya, misalnya mengubah data pada *record* ke-2. Maka pilih *record* ke-2 pada tabel tersebut lalu ketikkan perubahannya misalnya pada Nama Karyawan Susanti Agranea sebenarnya namanya Susanti Anggraeni.

NIP	NamaKaryawan	AlamatLeng	Handphone	JenisKel	Agama	TempatLahi	TanggalLahi	Pendidikan	TanggalGab	Golongan	Skala	Departemen
K01	Akhmad Ali Rizaldi	Jl. Simpang 02	878143556	L	Islam	Surabaya	28/02/1980	S1	29/01/2000	IIA	1	1
K02	Susanti Anggraeni	Jl. Kebon Jeruk	982919111	P	Islam	Bandung	15/09/1990	S1	01/01/2012	IIA	1	2
K03	I Made Idha Nuantera	Jl. Kebon Sirih	992871722	P	Hindu	Bali	06/06/1989	S1	08/08/2010	IIA	1	1
K04	Albert Sitohang	Jl. Siliwangi Jal	828919189	L	Kristen	Medan	09/09/1988	S1	08/08/2010	IIA	1	2
K05	I Nyoman Windu Asmara	Jl. Didina 9 Jak	928282827	L	Hindu	Bali	09/08/1989	S1	09/01/2007	IIA	2	1
K06	Lee King Woe	Jl. Gatot Subro	918181818	L	Budha	Korea	16/08/1987	S1	09/01/2008	IIA	2	3
K07	Rizki Azrian Gerardo	Jl. Lingkar Utar	829191982	L	Islam	Banjarmasin	16/11/1982	S1	28/01/2000	IIA	2	3
*			0								0	0

Gambar 5.17 Mengedit/Mengubah data

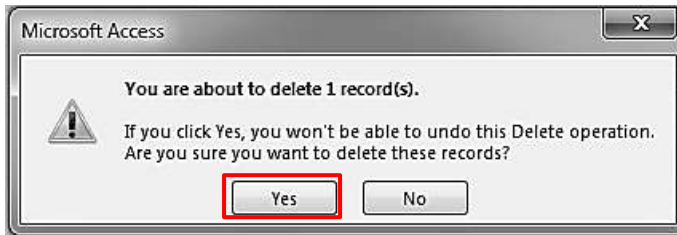
5.6.3 Menghapus Data

Untuk menghapus data di dalam tabel dengan cara: klik *double* tabel (misalnya **tbl_Karyawan**) → arahkan kursor ke arah *record* yang akan dihapus datanya, misalnya menghapus data pada *record* ke-7. Maka pilih dan klik kanan *record* ke-7 pada tabel tersebut → klik **Delete Record**.

NIP	NamaKaryawan	AlamatLeng	Handphone	JenisKel	Agama	TempatLahi	TanggalLahi	Pendidikan	TanggalGab	Golongan	Skala	Departemen
K01	Akhmad Ali Rizaldi	Jl. Simpang 02	878143556	L	Islam	Surabaya	28/02/1980	S1	29/01/2000	IIA	1	1
K02	Susanti Anggraeni	Jl. Kebon Jeruk	982919111	P	Islam	Bandung	15/09/1990	S1	01/01/2012	IIA	1	2
K03	I Made Idha Nuantera	Jl. Kebon Sirih	992871722	P	Hindu	Bali	06/06/1989	S1	08/08/2010	IIA	1	1
K04	Albert Sitohang	Jl. Siliwangi Jal	828919189	L	Kristen	Medan	09/09/1988	S1	08/08/2010	IIA	1	2
K05	I Nyoman Windu Asmara	Jl. Didina 9 Jak	928282827	L	Hindu	Bali	09/08/1989	S1	09/01/2007	IIA	2	1
K06	Lee King Woe	Jl. Gatot Subro	918181818	L	Budha	Korea	16/08/1987	S1	09/01/2008	IIA	2	3
K07	Rizki Azrian Gerardo	Jl. Lingkar Utar	829191982	L	Islam	Banjarmasin	16/11/1982	S1	28/01/2000	IIA	2	3
*			0								0	0

Gambar 5.18 Menghapus Data

Kemudian muncul konfirmasi apakah benar akan menghapus *record* tersebut, jika memilih **Yes** maka terhapus data pada *record* ke-7, sedangkan **No** *record* ke-7 tidak jadi terhapus.



Gambar 5.19 Konfirmasi Penghapusan Record

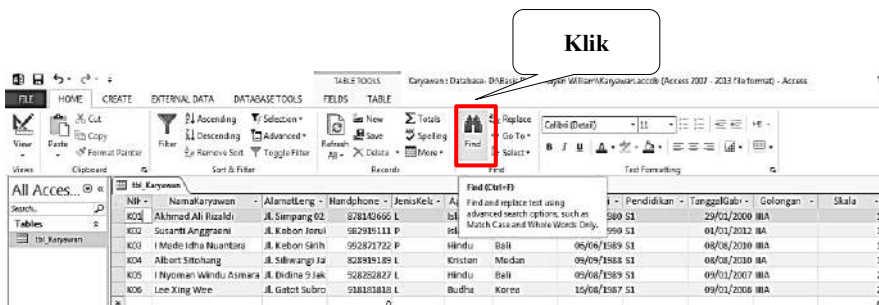
Maka data dari *record* ke-7 sudah terhapus.

Nik	NamaKaryawan	AlamatLeng	Handphone	JenisKekel	Agama	TempatLahi	TanggalLahi	Pendidikan	TanggalGab	Golongan	Skala	Departemen
K01	Ahmad Ali Rizaldi	Jl. Sumpang 02	878143666 L	Islam	Surabaya	28/02/1980 S1	29/01/2000	IIIA			1	1
K02	Susanti Anggraeni	Jl. Kebon Jeruk	982919111 P	Islam	Bandung	19/09/1990 S1	01/01/2012	IIA			1	2
K03	I Made Itha Nuantara	Jl. Kebon Sirih	992871722 P	Hindu	Bali	06/06/1989 S1	08/08/2010	IIIA			1	1
K04	Albert Sitohang	Jl. Silwangi Jai	828919189 L	Kristen	Medan	09/09/1988 S1	08/08/2010	IIA			1	2
K05	I Nyoman Windu Asmara	Jl. Didina 9 Jak	928282827 L	Hindu	Bali	09/08/1989 S1	09/01/2007	IIIA			2	1
K06	Lee Xing Wee	Jl. Gatot Subro	918181818 L	Budha	Korea	16/08/1987 S1	09/01/2008	IIIA			2	3
			0								0	0

Gambar 5.20 Data Telah Terhapus

5.6.4 Mencari Data

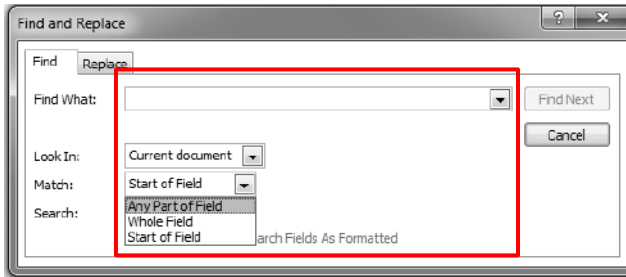
Untuk melakukan pencarian data di dalam tabel dengan cara: klik *double* tabel (misalnya *tbl_Karyawan*) → pada *Ribbon Home* → klik *Find* atau *Ctrl+F*.





Gambar 5.21 Ribbon Home Bagian Find

Kemudian muncul kotak dialog *Find and Replace*.



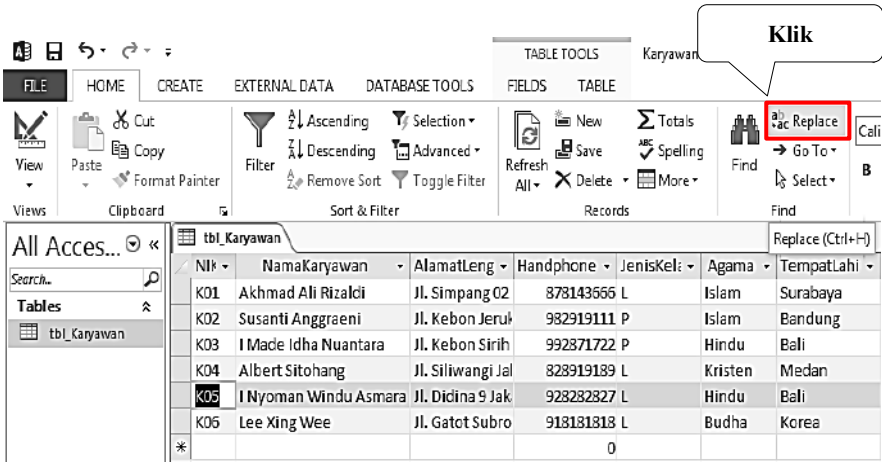
Gambar 5.22 Kotak Dialog *Find and Replace*

Keterangan:

Pada bagian *Match* terdapat 3 buah *combo box* yang masing-masing pilihan memiliki fungsi sebagai berikut:

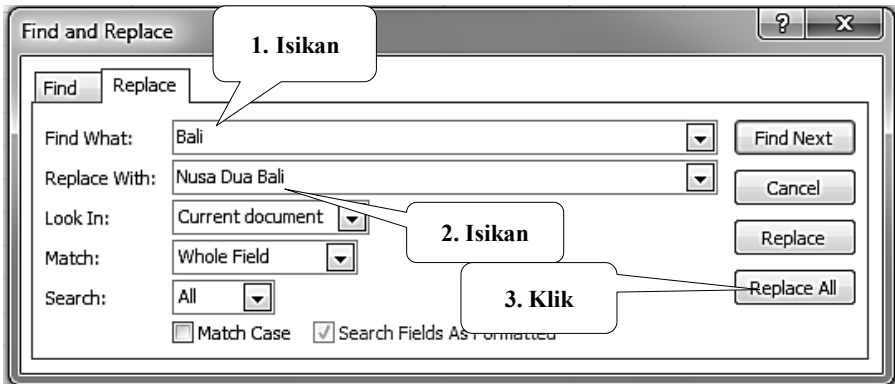
1. ***Any part of field:*** pencarian data yang mengandung kata atau teks yang dicari. Contoh: misal yang dicari kata **is** maka yang akan muncul adalah **islam** dan **kristen**.
2. ***Whole field:*** pencarian data yang mengandung teks atau kata yang utuh. Contoh: yang dicari itu kata **Bali** maka hasil pencarian kata **Bali** secara utuh.
3. ***Start of field:*** pencarian data yang diawali dengan kata atau teks yang dicari. Contoh: yang dicari adalah awalan kata **K** maka semua yang mengandung awalan **K** itu yang dimunculkan.

Misalnya Anda akan mencari data NIK **K05** atas nama siapa. Ketikkan **K05** pada isian *Find What* → klik *Find Next* dan akan muncul tampilan seperti dibawah ini.



Gambar 5.24 Ribbon Home Bagian Replace

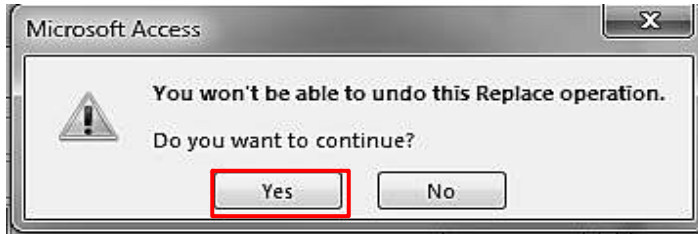
Muncul kotak dialog *Find and Replace*, misalnya Anda akan mencari data yang mengandung kata **Bali** kemudian menggantinya dengan **Nusa Dua Bali**. Caranya: pada bagian *Find What* ketikkan **Bali** dan pada bagian *Replace With* **Nusa Dua Bali** → klik *Replace All*.



Gambar 5.25 Kotak Dialog Find and Replace



Kemudian akan muncul konfirmasi, klik **Yes**.



Gambar 5.26 Konfirmasi *Replace*

Hasil dari operasi *Find and Replace* sebagai berikut:

Hasilnya

NR	NamaKaryawan	Alamat.eng	Handphone	JenisKel	Agama	TempatLah	TanggalLah	Pa	TanggalGabi	Golongan	Skala	Departem
K01	Akhmad Ali Rizaldi	Jl. Siringang 02	878143666	L	Islam	Surabaya	28/02/1990	S1	29/01/2000	IIA	1	1
K02	Susanto Angraeni	Jl. Kebon Jenu	982319111	P	Islam	Barduna	13/01/1990	S1	01/01/2012	IIA	1	2
K03	I Made Itha Nuantara	Jl. Kebon Sirih	992871722	P	Hindu	Nusa Dua Bali	06/06/1989	S1	08/08/2010	IIA	1	1
K04	Albert Sitohang	Jl. Sitawangi Jai	828219189	L	Kristen	Medan	09/09/1988	S1	08/08/2010	IIA	1	2
K05	I Nyoman Windu Asmara	Jl. Diklina 3 Jak	928282827	L	Hindu	Nusa Dua Bali	09/08/1983	S1	09/01/2007	IIA	2	1
K06	Lee Xing Wee	Jl. Gatot Subro	918181818	L	Budha	Korea	16/08/1987	S1	09/01/2008	IIA	2	3

Gambar 5.27 Hasil Operasi Mencari dan Mengganti Data

5.6.6 Mengurutkan Data dalam Tabel

Untuk mengurutkan data dalam tabel dapat Anda lakukan dengan langkah-langkah sebagai berikut:

1. Aktif pada jendela **Datasheet View** sebuah tabel.
2. Pilih **kolom** yang akan dijadikan acuan pengurutan data.
3. Kemudian klik **kanan mouse** pada kolom terpilih, dan pilih **Sort A to Z**, untuk mengurutkan data secara menaik, atau **Sort Z to A** untuk mengurutkan secara menurun.



Selain itu Anda juga dapat mengurutkan data dengan cara pilih **Tab Home**, kemudian klik tombol **Sort A to Z** atau **Sort Z to A** pada grup **Sort and Filter**.

KodeBuku	JudulBuku	Pengarang	Penerbit	TahunTerbit
BDK-001-15	Belajar dan Pembelajaran	Diyati	Rineka Cipta	2013
B71-001-15	Pemrograman C++	Satriyanto	Andi Publisher	2008
B71-002-15	Pemrograman Visual	nel Delarose, Ph.D.	Informatika	2006
B71-003-15	Konsep Pemrograman	Suarga, M.Sc, M.Match	Andi Publisher	2012
B71-004-15	Panduan Lengkap MS Access 2013	er Talib	PT. Elex Media Komputindo	2014
B71-005-15	Praktis Belajar Database dengan MS A	Jul Kadir	Andi Publisher	2002
B71-006-15	Web Programming	hammad Sadeli	Maxikom	2009
B71-007-15	Access 2013 untuk Orang Awam	hammad Sadeli	Maxikom	2013
B71-008-15	Image Processing	hammad Sadeli	Maxikom	2012
B71-011-15	Organisasi dan Arsitektur Komputer E	liam Stallings	PT. Indeks Kelompok Gramedia	2003
B71-012-15	Sistem Basis Data	Nugroho	Andi Publisher	2011

Gambar 5.28 Contoh Proses Mengurutkan Data pada Kolom secara *Ascending (Sort A to Z)*

KodeBuku	JudulBuku	Eksemplar	Pengarang	Penerbit	TahunTerbit
B71-007-15	Access 2013 untuk Orang Awam	35	Muhammad Sadeli	Maxikom	2013
BDK-001-15	Belajar dan Pembelajaran	45	Dr. Diyati	Rineka Cipta	2013
B71-008-15	Image Processing	20	Muhammad Sadeli	Maxikom	2012
B71-003-15	Konsep Pemrograman	40	Dr. Suarga, M.Sc, M.Match	Andi Publisher	2012
B71-011-15	Organisasi dan Arsitektur Komputer Ed. 5	36	William Stallings	PT. Indeks Kelompok Gramedia	2003
B71-004-15	Panduan Lengkap MS Access 2013	30	Haer Talib	PT. Elex Media Komputindo	2014
B71-001-15	Pemrograman C++	20	Edi Satriyanto	Andi Publisher	2008
B71-002-15	Pemrograman Visual	30	Karnel Delarose, Ph.D.	Informatika	2006
B71-005-15	Praktis Belajar Database dengan MS Access	10	Abdul Kadir	Andi Publisher	2002
B71-012-15	Sistem Basis Data	29	Adi Nugroho	Andi Publisher	2011
B71-006-15	Web Programming	28	Muhammad Sadeli	Maxikom	2009

Gambar 5.29 Contoh Hasil Mengurutkan Data pada Kolom secara *Ascending (Sort A to Z)*

5.6.7 Menyaring Data Menggunakan *Filter*

Untuk menyaring data menggunakan tombol *Filter*, dapat Anda lakukan dengan langkah sebagai berikut:

1. Aktif pada jendela *Datasheet View* sebuah tabel.



2. Aktif pada salah satu *field* yang akan dijadikan kriteria penyaringan.
3. Klik tombol **Filter** dalam **Tab Home** pada **Grup Sort and Filter**, sehingga pada *field* aktif akan tampil daftar penyaringan.
4. Tentukan salah satu pilihan berikut:
 - a. **Sort A to Z** untuk mengurutkan data secara **Ascending** (menaik).
 - b. **Sort Z to A** untuk mengurutkan data secara **Descending** (menurun).
 - c. **Clear filter from...** (nama *field*), untuk menghilangkan pengaruh penyaringan pada *field* tersebut.
 - d. **Text Filters** untuk memilih operator penyaringan pada data bertipe **Text** sebagai berikut:
 - **Equals** fungsi sama dengan.
 - **Does Not Equals** fungsi tidak sama dengan.
 - **Begins With** fungsi diawali dengan.
 - **Does Not Begin With** fungsi tidak diawali dengan.
 - **Contains** fungsi berisi teks tertentu.
 - **Does Not Contain** fungsi tidak berisi teks tertentu.
 - **Ends With** fungsi diakhiri dengan.
 - **Does Not End With** tidak diakhiri dengan.
 - e. **Number Filters** untuk memilih operator penyaringan pada data bertipe **Number**.
 - **Equals** fungsi sama dengan.
 - **Does Not Equals** fungsi tidak sama dengan.
 - **Less Than** fungsi lebih kecil atau sama dengan.
 - **Greater Than** fungsi lebih besar atau sama dengan.
 - **Between** fungsi antara.

Latihan Praktik

1. Buatlah *database* dengan nama **Perpustakaan** dan buatlah tabel dengan nama **Tbl_Buku** dengan struktur data sebagai berikut:

<i>Field Name</i>	<i>Data Type</i>	<i>Description</i>
-------------------	------------------	--------------------



KodeBuku	Short Text	Kode Buku [10]
Judul	Short Text	Judul Buku [255]
Eksemplar	Short Text	Eksemplar Buku [25]
Pengarang	Short Text	Nama Pengarang Buku [255]
Penerbit	Short Text	Nama Penerbit Buku [255]
TahunTerbit	Number	Tahun Terbit Buku [4]

2. Untuk *field name* **KodeBuku** jadikan sebagai **Primary Key**.
3. Pada *field name* **Judul** lakukan *edit field name* menjadi **JudulBuku**.
4. Pada *data type* **Eksemplar** adalah **Short Text** ganti dengan **Number**.
5. Isikan data buku ke dalam tabel **Tbl_Buku** sebagai berikut:

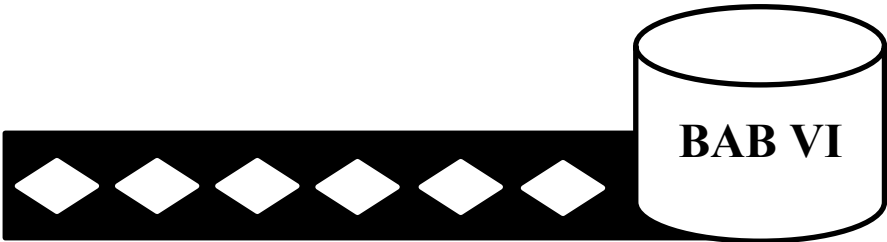
KodeBuku	JudulBuku	Eksemplar	Pengarang	Penerbit	TahunTerbit
KB001	Pemrograman C++	20	Edi Satryanto	Andi Publisher	2008
KB002	Pemrograman Visual	30	Karnel Delarose, Ph.D.	Informatika	2006
KB003	Konsep Pemrograman	40	Dr. Suarga, M.Sc, M.Match	Andi Publisher	2012
KB004	Dasar-Dasar Kependidikan	25	Dr. Sriono, M.Ed	Andi Publisher	1999
KB005	Panduan Lengkap MS Access 2013	30	Haer Talib	PT. Elex Media Komputindo	2014
KB006	Praktis Belajar Database dengan MS Access	10	Abdul Kadir	Andi Publisher	2002
KB007	Web Programming	28	Muhammad Sadeli	Maxikom	2009
KB008	Belajar dan Pembelajaran	45	Dr. Diyati	Rineka Cipta	2013
KB009	Access 2013 untuk Orang Awam	35	Muhammad Sadeli	Maxikom	2013
KB010	Image Processing	20	Muhammad Sadeli	Maxikom	2012
KB011	Organisasi dan Arsitektur Komputer Ed. 6	36	William Stallings	PT. Indeks Kelompok Gramedia	2003
KB012	Sistem Basis Data	29	Adi Nugroho	Andi Publisher	2011

6. Lakukan *edit* data pada *record* ke-3 untuk data **Konsep Pemrograman** ganti dengan **Algoritma dan Pemrograman**.
7. Lakukan *edit* data pada *record* ke-8 untuk data **Dr. Diyati** diganti dengan **Dr. Dimiyati**.
8. Lakukan pencarian data untuk pengarang **Muhammad Sadeli**, buku apa saja yang dikarang.
9. Lakukan pencarian data untuk penerbit **Andi Publisher**, buku apa saja yang diterbitkan.
10. Lakukan pencarian kemudian ganti data, cari data **Image Processing** dan ganti menjadi **Pengolahan Citra**.
11. Hapus data pada *record* ke-4.



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

12. Lakukan penyaringan pada *field JudulBuku* secara *Ascending*.
13. Lakukan penyaringan pada *field JudulBuku* secara *Descending*.



MENGGUNAKAN FIELD PROPERTIES PADA DESIGN TABLE



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

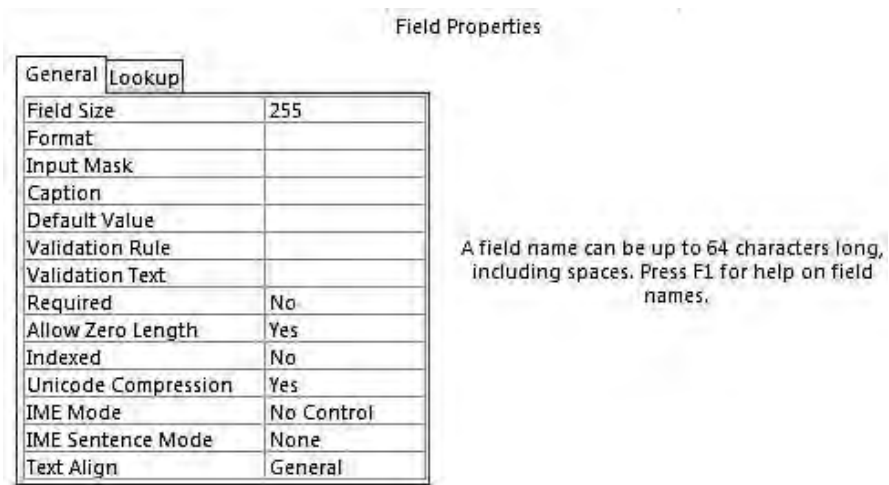
Tujuan Instruksional Khusus

1. Mahasiswa mampu mengatur format masukan data
2. Mahasiswa mampu menentukan *input mask*
3. Mahasiswa mampu menentukan nilai otomatis (*default value*)
4. Mahasiswa mampu menentukan validasi masukan data (*validation rule*)
5. Mahasiswa mampu menentukan pesan kesalahan pada masukan data (*validation text*)
6. Mahasiswa mampu menggunakan fasilitas *lookup*



6.1 Field Properties

Field Properties (di bagian bawah **Tabel Design**) digunakan untuk mengatur *field* lebih lanjut. Properti yang umumnya diatur adalah ukuran *field* (**Field Size**) dan formatnya (**Format**). Pengaturan *field properties* ini bersifat opsional, namun jika Anda mengaturnya pada sebuah desain Tabel, pengaturan ini akan otomatis dibawa ketika Anda menggunakan *field* tersebut dalam *query*, *form*, maupun *report*.



Gambar 6.1 Area Field Properties

Perlu diperhatikan bahwa pada kotak *Field Properties* terdapat beberapa pilihan sebagai berikut:

Tabel 6.1 Field Properties

Perintah	Fungsi
<i>Field Size</i>	Menentukan jumlah maksimum karakter yang diisikan pada <i>field</i> tersebut.
<i>Format</i>	Menentukan format tampilan data.
<i>Input Mask</i>	Menentukan standarisasi tampilan pada saat memasukkan data di layar.
<i>Decimal Place</i>	Menentukan jumlah angka desimal yang diinginkan.



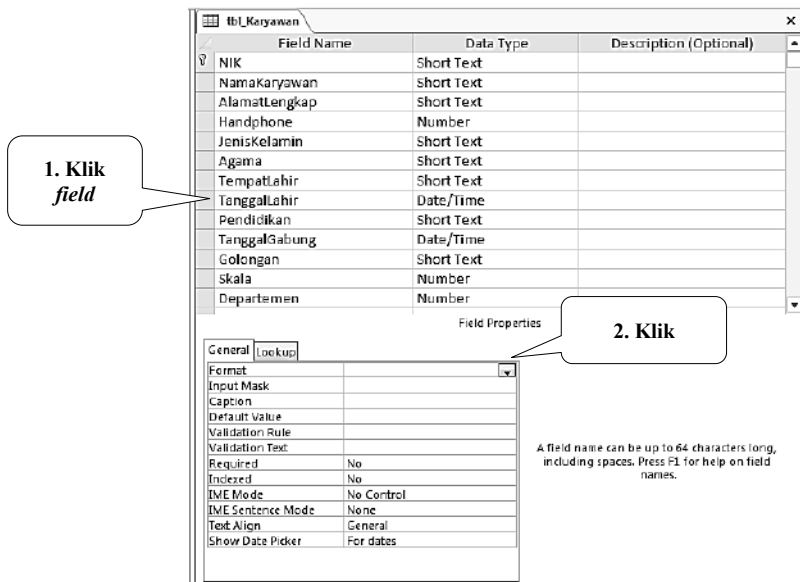
Perintah	Fungsi
<i>Caption</i>	Menempatkan keterangan sebagai judul, kolom, formulir, atau laporan.
<i>Default Value</i>	Menempatkan data yang sering digunakan atau data yang sama pada <i>field</i> tertentu agar selalu ditampilkan kembali.
<i>Validation Rule</i>	Mengontrol atau membatasi sampai sejauh mana data yang Anda masukkan dalam sebuah <i>database</i> .
<i>Validation Text</i>	Menampilkan keterangan atau pesan apabila data yang dimasukkan tidak sesuai dengan batasan yang ada di <i>validation rule</i> .
<i>Required</i>	Mengatur apakah <i>field</i> ini boleh dikosongkan atau tidak pada saat pengisian <i>record</i> .
<i>Allow Zero Length</i>	Mendefinisikan nilai <i>blank</i> ("") yang membedakannya dengan nilai <i>null</i> .
<i>Indexed</i>	Membuat <i>index</i> /urutan data pada <i>field</i> yang disorot. <i>Index</i> dapat mempercepat proses pencarian, pengurutan, dan pengelompokan.
<i>Unicode Compression</i>	Properti ini berlaku pada <i>field</i> bertipe <i>Short Text</i> , <i>Long Text</i> , dan <i>Hyperlink</i> . Jika diisi <i>Yes</i> , maka setiap karakter Latin akan di <i>compress</i> ketika disimpan dan di- <i>uncompress</i> kembali ketika diambil. <i>Unicode</i> adalah skema pengodean karakter standar yang memungkinkan hampir semua bahasa tulisan di dunia diwakili oleh sebuah <i>character set</i> .
<i>IME Mode dan IME Sentence Mode</i>	Kedua properti ini digunakan jika <i>daTabase</i> Anda berkaitan dengan bahasa Kanji (Jepang)
<i>New Values</i>	Properti ini terdapat pada tipe data <i>AutoNumber</i> . Di sini tersedia dua pilihan yaitu <i>Increment</i> dan <i>Random</i> . Umumnya properti ini diisi dengan pilihan <i>Increment</i> , dimana nilai baru selalu meningkat tetapi Anda bisa memilih <i>Random</i> di mana nilai baru diperoleh secara acak namun tetap bersifat unik.
<i>Append Only</i>	Properti ini hanya tersedia pada tipe data <i>Hyperlink</i> dan <i>Long Text</i> , di mana jika diisi <i>Yes</i> , Microsoft Office Access 2013 akan menyimpan riwayat nilai yang pernah dimasukkan pada <i>field</i> (nilai yang ditampilkan selalu nilai baru).
<i>Text Format</i>	Tersedia pada tipe data <i>Long Text</i> , di mana Anda bisa memilih <i>Rich Text</i> jika ingin menyimpan teks berformat. Secara <i>default</i> properti ini berisi <i>Plain Text</i> .
<i>TextAlign</i>	Menentukan perataan <i>default</i> teks dalam sebuah <i>control</i> .
<i>Expression</i>	Terdapat pada tipe data <i>Calculated</i> di mana Anda bisa mengisi ekspresi atau formula/rumus yang digunakan dalam kalkulasi.



Perintah	Fungsi
<i>Result Type</i>	Terdapat pada tipe data <i>Calculated</i> di mana Anda bisa menentukan tipe data hasil kalkulasi. Di sini tersedia pilihan seperti: <i>Double</i> , <i>Single</i> , <i>Integer</i> , dan sebagainya.

6.2 Mengatur Format Masukan Data

Berikut caranya: aktifkan kembali bagian *Design View* pada Tabel (misalkan *Tbl_Karyawan*) → klik bagian *field name TanggalLahir* → perhatikan pada bagian *Field Properties* dari *TanggalLahir*.



Gambar 6.2 *Field Properties* pada *Field TanggalLahir* Bertipe Data *Date/Time*

Pada kotak dialog *Field Properties Tab General*, klik kotak *Format* → klik tombol panah bawah (*drop-down*), sehingga muncul beberapa pilihan format tanggal yang diinginkan, misalnya memilih *Medium Date*.



General		Lookup
Format	▼	
Input Mask	General Date	12/11/2015 17:34:23
Caption	Long Date	12 Nopember 2015
Default Value	Medium Date	12-Nop-15
Validation Rule	Short Date	12/11/2015
Validation Text	Long Time	17:34:23
Required	Medium Time	5:34
Indexed	Short Time	17:34
IME Mode	No Control	
IME Sentence Mode	None	
Text Align	General	
Show Date Picker	For dates	

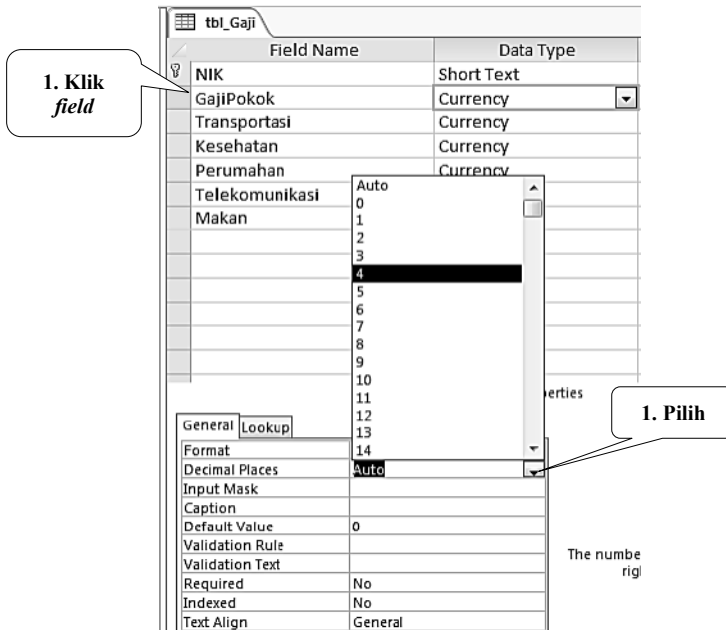
Gambar 6.3 Properties Format pada Data Type Date/Time

Namun jika yang diformat adalah *data type Number*, maka yang muncul sebagai berikut:

General		Lookup
Field Size	Long Integer	
Format	▼	
Decimal Places	General Number	3456,789
Input Mask	Currency	Rp3.457
Caption	Euro	€3.456,79
Default Value	Fixed	3456,79
Validation Rule	Standard	3.456,79
Validation Text	Percent	123,00%
Required	Scientific	3,46E+03
Indexed	No	
Text Align	General	

Gambar 6.4 Properties Format pada Data Type Number

Jika yang diformat adalah *data type Number/Currency*, dan Anda ingin mengatur **Decimal Places**, caranya: klik *field* (misalnya **GajiPokok**) bertipe data **Currency** → pada bagian **Field Properties Tab General** klik **Decimal Places** → klik **tombol panah bawah (drop-down)** sehingga muncul beberapa pilihan format *decimal* yang diinginkan, lalu pilih berapa *digit desimal* yang ingin ditampilkan.



Gambar 6.5 Properties Decimal Places pada Data Type Currency

6.3 Menentukan *Input Mask*

Digunakan untuk mengontrol data yang dimasukkan oleh pemakai. *Input Mask* berfungsi seperti *template*, di mana pemakai hanya bisa memasukkan data sesuai aturan pada *Input Mask*. Properti *Input Mask* bisa terdiri dari tiga bagian, yaitu sebagai berikut:

Tabel 6.2 Properti *Input Mask*

Bagian pertama	Merupakan <i>Input Mask</i> itu sendiri, misalnya: !(999) 999-9999.
Bagian kedua	Diisi dengan 0 jika data disimpan beserta formatnya, atau 1 (atau kosong) jika hanya data saja yang disimpan ke dalam <i>field</i> .
Bagian ketiga	Diisi dengan karakter yang ingin digunakan sebagai <i>template</i> (tempat pemakai mengetikkan huruf/angka), misalnya “_”, “-”, “*”, dan lain-lain sebagainya (bisa semua karakter). Jika tidak ingin menggunakan karakter (hanya kosong saja), gunakan spasi yang diapit tanda kutip (“ ”).



Berikut simbol-simbol yang digunakan dalam *Input Mask*:

Tabel 6.3 Simbol-Simbol pada *Input Mask*

Simbol	Keterangan
0	Untuk angka (0-9) yang harus diisi. Tanda + dan – tidak diterima.
9	Angka atau spasi, namun tidak harus diisi (bisa dibiarkan kosong/tidak diisi). Tanda + dan – tidak diterima.
#	Angka atau spasi, tidak harus diisi, dan tanda + dan – diterima.
L	Huruf (A-Z) yang harus diisi.
?	Huruf, tetapi tidak harus diisi.
A	Huruf atau angka, harus diisi.
a	Huruf atau angka, tidak harus diisi.
ε	Semua huruf dan angka termasuk spasi, harus diisi.
C	Semua huruf dan angka termasuk spasi, tidak harus diisi.
., -/	Pemisah desimal, ribuan, tanggal, dan waktu. Tanda yang akan ditampilkan disesuaikan dengan <i>Regional Settings</i> .
<	Untuk membuat semua huruf menjadi huruf kecil.
>	Untuk membuat semua huruf menjadi huruf besar.
!	Untuk perataan kanan.
\	Untuk menampilkan karakter yang mengikutinya secara apa adanya, misal untuk meampilkan huruf A: \A.

Untuk *field* yang berisi *password*, gunakan *Input Mask Password* agar setiap karakter yang dimasukkan terlihat sebagai “*”, namun isi *field* tetap seperti apa adanya. Contoh dari *Input Mask* misalnya:

Tabel 6.4 Contoh *Input Mask*

<i>Input Mask</i>	Di-input-kan Data	Yang Akan Ditampilkan di Dalam Layar
(000) 000-0000	2065550248	(206) 555-0248
(999) 999-9999	2065550248	(206) 555-0248
(000) AAA-AAAA	2065A5TERE	(206) 5A5-TERE
#999	-200	-200
SSN 000-00-000	SSN12311123	SSN 123-11-123
SSN 999-99-999	SSNA45R44BT	SSN A45-R4-4BT
>LL00-00	db1234	DB12-34

Contoh untuk melakukan penentuan *Input Mask* dalam aturan *Field Name* dengan cara sebagai berikut: Klik *field* yang akan diformat (misalnya



field NIK pada Tabel Tbl_Karyawan), pada kotak dialog **Field Properties Tab General** → klik bagian **Input Mask**, lalu ketikkan aturan: **KYW-000-00-00** dimana **KYW** itu aturan penulisan, selain **KYW** tidak bisa dimasukkan, 0 itu harus berupa angka tidak – dan + serta huruf, kemudian format – pada **KYW-000-00-00** bukanlah negatif tetapi hanya untuk membatasi angka tersebut, seperti gambar di bawah ini.

Field Name	Data Type
NIK	Short Text
NamaKaryawan	Short Text
AlamatLengkap	Short Text
Handphone	Number
JenisKelamin	Short Text
Agama	Short Text
TempatLahir	Short Text
TanggalLahir	Date/Time
Pendidikan	Short Text
TanggalGabung	Date/Time
Golongan	Short Text
Skala	Number
Departemen	Number

General	Lookup
Field Size	10
Format	
Input Mask	*KYW-000\00\00
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

Gambar 6.6 Properties Input Mask



tbl_Karyawan										
NIK	NamaKaryawan	AlamatLeng	Handphone	JenisKel	Agama	TempatLahi	TanggalLahi	Pendidikan	TanggalGabur	Golongan
KYW-001-00-11	Akmalia Ali Rizaldi	Jl. Simpang 02	878143666	L	Islam	Surabaya	28-Feb-80	S1	29/01/2000	IIIA
KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeruh	982919111	P	Islam	Bandung	19-Sep-90	S1	01/01/2012	IIIA
KYW-003-10-11	I Made Icha Nuantara	Jl. Kebon Sirih	992871722	P	Hindu	Nusa Dua Bali	06-Jun-89	S1	08/08/2010	IIIA
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jai	828919189	L	Kristen	Medan	09-Sep-88	S1	08/08/2010	IIIA
KYW-005-07-11	Njoman Windu Asmara	Jl. Didina 9 Jak	928282827	L	Hindu	Nusa Dua Bali	09-Agust-89	S1	09/01/2007	IIIA
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subro	918181818	L	Budha	Korea	16-Agust-87	S1	09/01/2008	IIIA

Gambar 6.7 Hasil Aturan *Input Mask* dalam *Field* NIK Tabel `tbl_Karyawan`

6.4 Menentukan Nilai Otomatis (*Default Value*)

Properti ini digunakan untuk memberi nilai otomatis pada *field*, yaitu nilai atau data yang otomatis dimasukkan ke *field* ketika *record* baru diciptakan. Dengan cara: klik *field* yang akan diformat (misalnya *field* `NamaKaryawan` pada Tabel `Tbl_Karyawan`), pada kotak dialog ***Field Properties Tab General*** → klik bagian ***Default Value***, lalu ketikkan nilai yang diinginkan misalnya **50** dan lihat hasilnya pada data di dalam Tabel `tbl_Karyawan`. Nama karyawan yang melebihi 50 karakter tidak akan bisa *ter-input* karena sudah dibatasi oleh *Default Value* ini.



1. Klik *field*

Field Name	Data Type
NIK	Short Text
NamaKaryawan	Short Text
AlamatLengkap	Short Text
Handphone	Number
JenisKelamin	Short Text
Agama	Short Text
TempatLahir	Short Text
TanggalLahir	Date/Time
Pendidikan	Short Text
TanggalGabung	Date/Time
Golongan	Short Text
Skala	Number
Departemen	Number

Field Properties

General	Lookup
Field Size	50
Format	
Input Mask	
Caption	
Default Value	50
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

2. Isikan

A value th

Gambar 6.8 Properties Default Value

6.5 Menentukan Validasi Masukan Data (*Validation Rule*)

Selain menentukan *Input Mask* dan *Default Value*, Anda juga dapat menentukan validasi masukan data (*Validation Rule*) aturan data yang harus dimasukkan ke dalam *field*. Dengan panjang maksimum adalah 2.048 karakter dan tidak boleh mengandung UDF (*User-Defined Function*/fungsi yang dibuat sendiri). Misalnya Anda ingin pada data *field Pendidikan* pada **tbl_Karyawan** hanya bisa diterima jika dia sudah Sarjana 1, 2, dan 3 dengan aturan S-?. Dimana S itu singkatan sarjana dan ? itu adalah berupa angka apakah 1, 2, atau 3, ketika dia mengetikkan SMA/SMP data tidak akan diterima dan muncul pesan kesalahan masukan data.

Dengan cara: klik *field* yang akan diformat (misalnya *field Pendidikan* pada tabel **tbl_Karyawan**), pada kotak dialog *Field*



Hasilnya

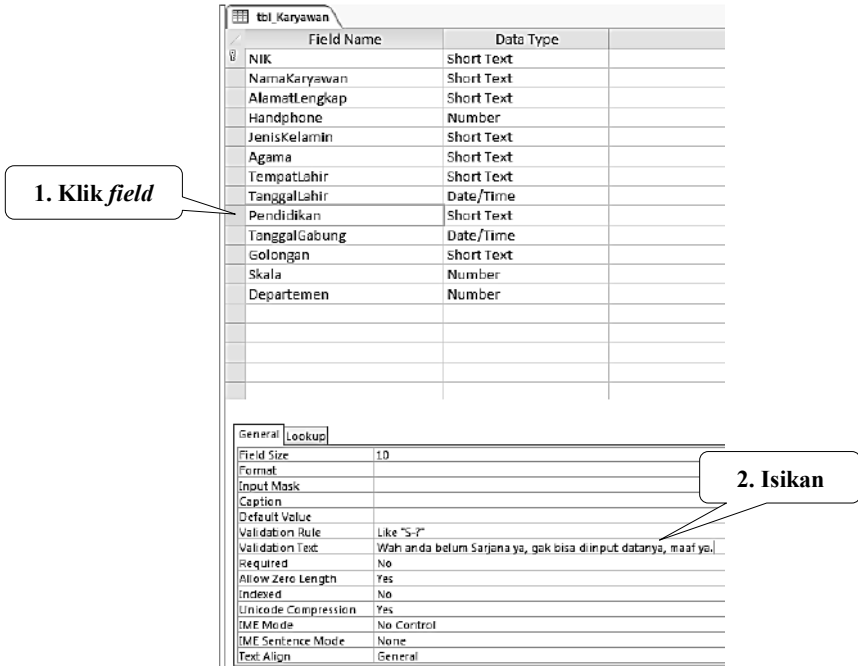
NIK	NamaKaryawan	AlamatLeng	Handphone	JenisKel	Agama	TempatLahi	Tanggal	Pendidikan	TanggalGabur
KYW-001-00-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143666	L	Islam	Surabaya	28-Feb-80	S-1	25/01/2000
KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeruk	982919111	P	Islam	Bandung	19-Sep-90	S-1	01/01/2012
KYW-003-10-11	I Made Icha Nuantara	Jl. Kebon Sirih	992871722	P	Hindu	Nusa Dua Bali	06-Jun-89	S-1	08/08/2010
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jai	828919189	L	Kristen	Medan	09-Sep-88	S-1	08/08/2010
KYW-005-07-11	I Nyoman Windu Asmara	Jl. Didina 9 Jak	928282827	L	Hindu	Nusa Dua Bali	09-Agust-89	S-1	09/01/2007
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subro	918181818	L	Budha	Korea	15-Agust-87	S-1	09/01/2008

Gambar 6.10 Hasil Aturan *Validation Rule* dalam *field* Pendidikan tabel *tbl_Karyawan*

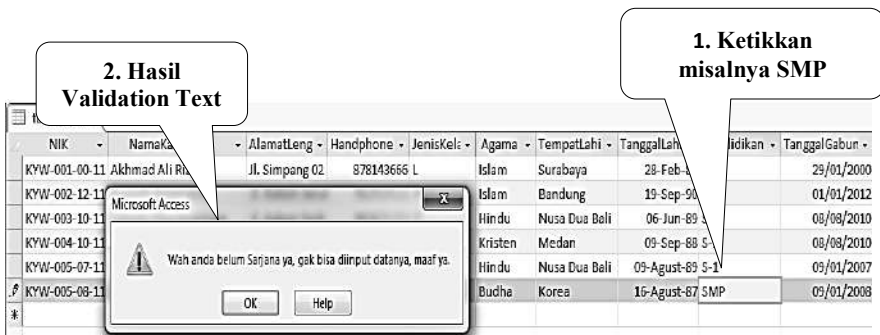
6.6 Menentukan Pesan Kesalahan pada Masukan Data (*Validation Text*)

Setelah kita menentukan *Validation Rule* pada sebuah *field* maka agar lebih baik lagi maka kita akan memberikan pesan kesalahan ketika terjadi kesalahan input data dari aturan pada *Validation Rule*. Panjang maksimum dari *Validation Text* adalah 255 karakter. Misalnya kita ingin memasukkan pesan kesalahan ketika akan meng-*input* data misalnya mengetikkan SMA pada *field Pendidikan tbl_Karyawan* berupa kata berikut: **Wah Anda belum Sarjana ya, gak bisa diinput datanya, maaf ya.**

Dengan cara: Klik *field* yang akan diformat (misalnya *field Pendidikan* pada tabel *tbl_Karyawan*), pada kotak dialog **Field Properties** tab **General** → klik bagian **Validation Text**, lalu ketikkan **Wah Anda belum Sarjana ya, gak bisa diinput datanya, maaf ya.** Dan lihat hasilnya ketika Anda mengetikkan selain S-1, 2, 3 pada *field Pendidikan tbl_Karyawan*.



Gambar 6.11 Field Properties Validation Text



Gambar 6.12 Hasil Aturan Validation Text



6.7 Menggunakan Fasilitas *Lookup*

Pada bidang **Field Properties** terdapat **Tab Lookup**. Fasilitas ini digunakan untuk memilih jenis *control* untuk menampilkan *field* dan membuat *lookup* yang berupa pilihan data untuk mengisi *field*. Keperluan untuk menyediakan pilihan data antara lain:

- Agar pemakai mudah memasukkan data, yaitu tinggal memilih dari data yang tersedia tanpa mengetikkan lagi.
- Mengurangi kesalahan dalam pemasukan data.
- Memudahkan pemakai memasukkan data yang berupa kode, karena pemakai tidak perlu mengingat atau mengetahui kode yang diperlukan.

Sebagai contoh buatlah dulu tabel dengan nama **tbl_Departemen** seperti di bawah ini dan isikan datanya.

Field Name	Data Type
KodeDepartemen	Number
NamaDepartemen	Short Text
LokasiKerja	Short Text

Gambar 6.13 Struktur Data **tbl_Departemen**

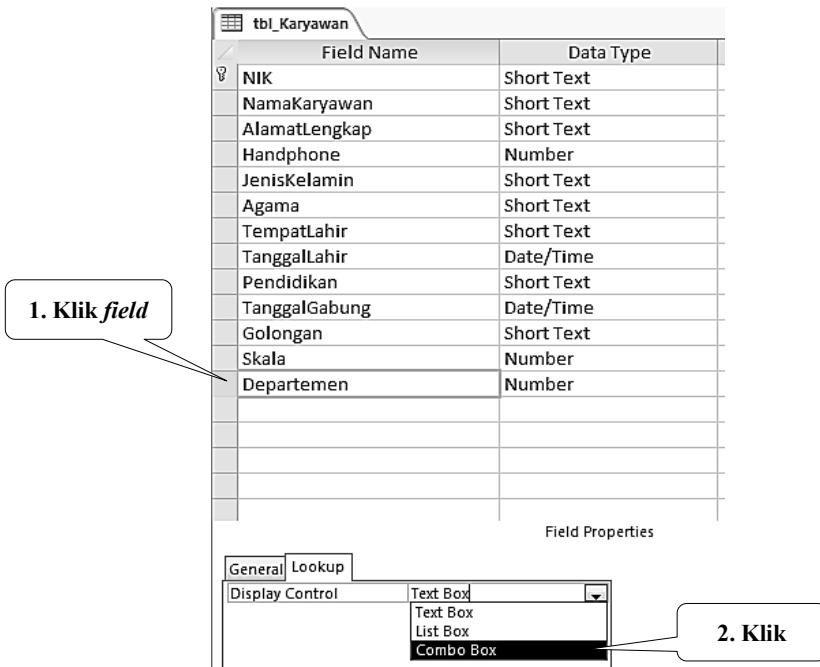
KodeDepart	NamaDeepar	LokasiKerja
1	Produksi	Cikarang
2	Pemasaran	Jakarta
3	Personalia	Jakarta
4	Keuangan	Jakarta

Gambar 6.14 Data Tabel **tbl_Departemen**



Sekarang pada *field Departemen* tabel **tbl_Karyawan**, *field* ini bertipe data *Number* dengan ukuran *Long Integer*, dimana untuk diisi dengan angka yang berasal dari **tbl_Departemen** (*field KodeDepartemen*). Tentu memakai *database* akan sulit mengingat angka *KodeDepartemen*, apalagi kode ini tidak dipublikasikan. Untuk itu kita akan membuat *Lookup* yang akan membaca **KodeDepartemen** dari tabel **tbl_Departemen** secara otomatis.

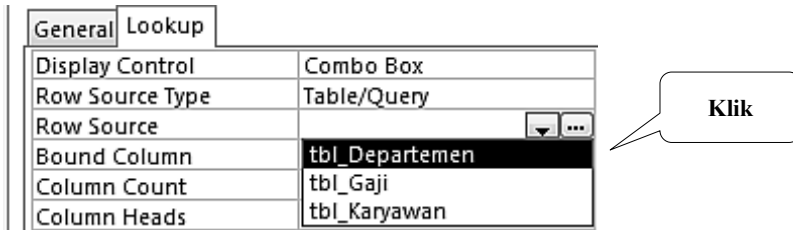
Berikut caranya: bukalah tabel **tbl_Karyawan** ke dalam tabel *Design View* → klik *field Departemen* → pada bagian *Field Properties* klik *Tab Lookup*. Pada properti *Display Control* ubah menjadi *Combo Box*.



Gambar 6.15 Mengatur *Display Control* pada *Tab Lookup* pada *Field Departemen* **tbl_Karyawan**

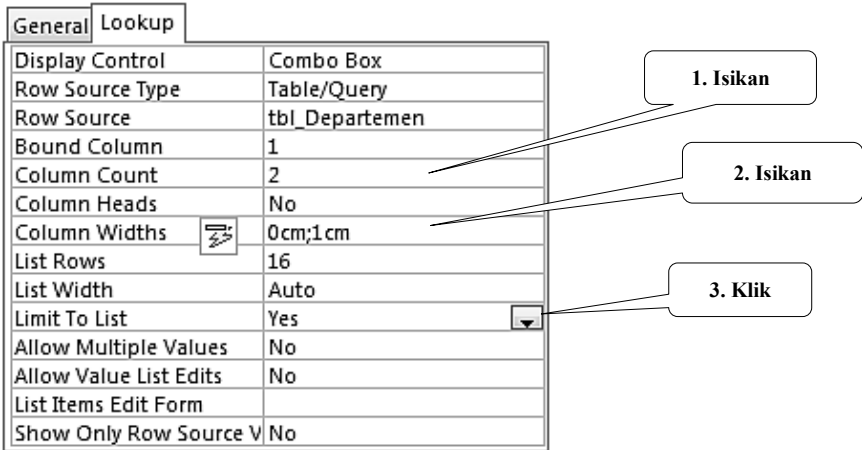


Muncul properti tambahan untuk mengatur Combo Box. Klik properti **Row Source** → pilih **tbl_Departemen** dari data yang tersedia.



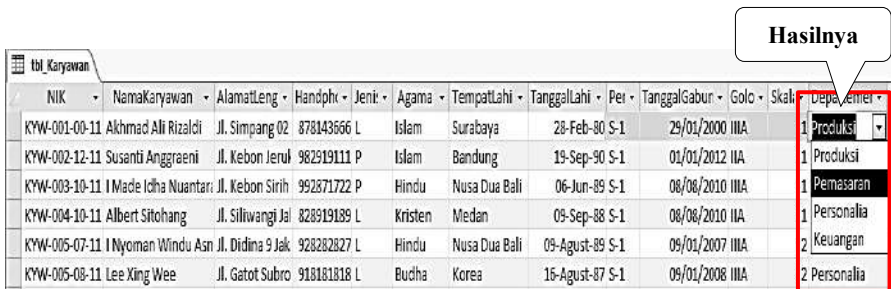
Gambar 6.16 Pilihan pada Properti *Row Source*

Pada properti *Column Count* ubah menjadi nilai **2**. Pada properti *Column Widths* masukkan **0;1** (tanda ; adalah *list separator* jika Anda menggunakan *Region* = Indonesia, jika Anda tidak mengubah *Region Settings*, cobalah menggunakan tanda koma sebagai *list separator*). Perhatikan bahwa angka **0;1** yang dimasukkan tersebut akan otomatis diubah oleh Microsoft Office Access 2013, misalnya menjadi: **0cm; 1cm**. Dan juga perhatikan untuk properti *Limit To List* ubah menjadi **Yes**, yang berarti pemakai hanya boleh memasukkan data sesuai dengan data yang tersedia, atau tidak bisa menambahkan data baru (untuk penambahan data departemen harus dilakukan pada **tbl_Departemen**).



Gambar 6.17 Pilihan pada Properti *Column Count*, *Column Widths*, *Limit To List*

Kemudian simpan tabel **tbl_Karyawan**, lalu klik *double* lagi **tbl_Karyawan** → klik *field* **Departemen**, maka akan muncul pilihan data sebagai berikut dan itulah yang dinamakan **Lookup**.



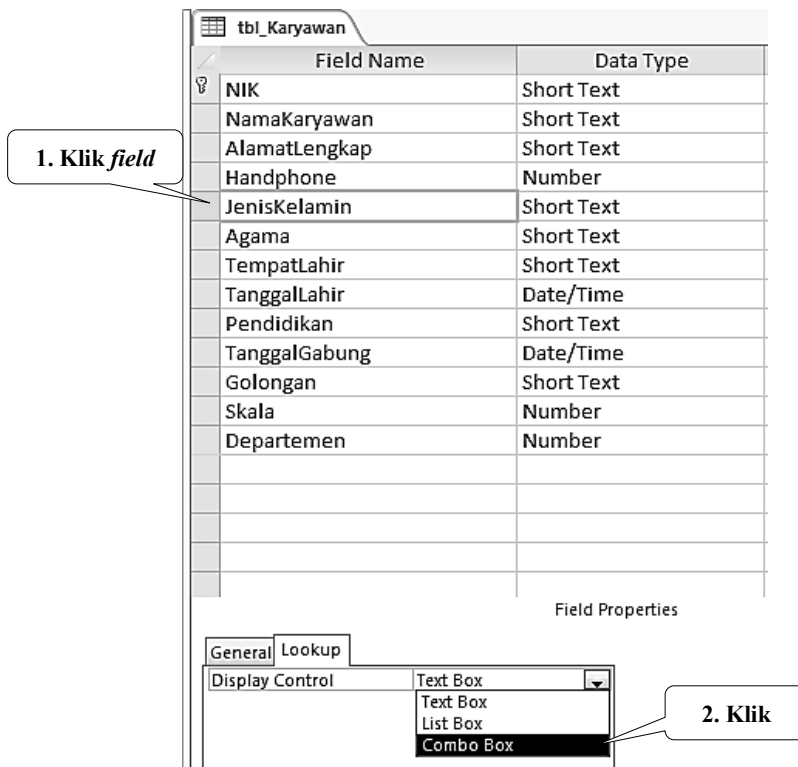
Gambar 6.18 Hasil **Lookup** pada *Field* **Departemen** **tbl_Karyawan**

Selain melakukan **Lookup** terhadap (a) tabel sebagai sumber data, **lookup** juga bisa, (b) menggunakan sumber data yang dimasukkan secara



manual, atau (c) menggunakan data yang pernah dimasukkan pada *field* itu sendiri.

Berikut ini adalah membuat **Lookup** jenis a dan b dengan cara: bukalah tabel **tbl_Karyawan** dalam bagian **Design View** sebagai contoh kita pilih *field* **JenisKelamin** → klik **Tab Lookup** → ubahlah pada bagian **Display Control** menjadi **Combo Box**.

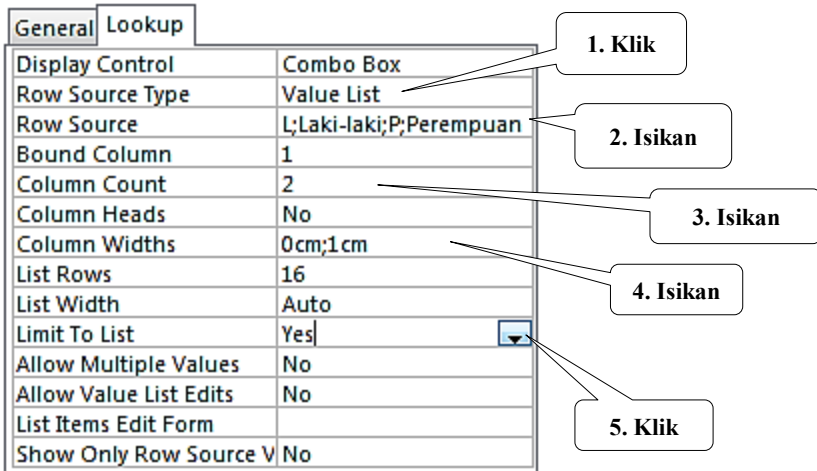


Gambar 6.19 Mengatur *Display Control* pada *Tab Lookup*

Pada properti **Row Source Type** pilih **Value List**. Pada properti **Row Source**, masukkan: **L;Laki-laki;P;Perempuan** (ingat bahwa tanda titik koma (“;”) sebagai *list separator* mungkin harus diganti koma pada



komputer Anda). Kemudian pada properti **Column Count** ubah nilainya menjadi **2**. Untuk properti **Column Widths** isi dengan: **0;1**. Dan terakhir pada properti **Limit To List** pastikan sudah pada posisi **Yes**.



Gambar 6.20 Pilihan pada Properti **Column Count**, **Column Widths**, **Limit To List**

Simpan perubahan pada **tbl_Karyawan** dan lihat hasilnya.



Gambar 6.21 Hasil *Lookup* pada *Field* **JenisKelamin** **tbl_Karyawan**



Latihan Praktik

1. Buka kembali *database Perpustakaan* pada praktikum sebelumnya.
2. Buka tabel **Tbl_Buku** kemudian edit pada *field KodeBuku* isikan aturan berikut:

Field Size : 8

Input Mask : LLL-999-99

Maksudnya adalah KodeBuku harus diawali 3 huruf yang telah ditentukan pada *validation rule*, 3 *digit* di tengah urutan masuk buku, dan 2 *digit* terakhir tahun buku di beli.

Validation Rule : **Left([KodeBuku];3)="BTI" Or Left([KodeBuku];3)="BDK"**

Maksudnya adalah *validation rule* untuk *field KodeBuku* hanya disediakan 2 buah kode buku yaitu BTI (Buku Teknologi Informasi) dan BDK (Buku Pendidikan), untuk angka 3 itu adalah format awal memasukan data dari KodeBuku tersebut.

Validation Text : **Kode yang di-input-kan salah, hanya ada buku BTI (Buku Teknologi Informasi) dan BDK (Buku Pendidikan). Harap cek dan ulangi kembali!**

Dan hasilnya sebagai berikut:

KodeBuku	JudulBuku	Eksemplar	Pengarang	Penerbit	TahunTerbit
BDK-001-15	Belajar dan Pembelajaran	45	Dr. Diyato	Rineka Cipta	2013
BTI-001-15	Pemrograman C++	20	Edi Satriyanto	Andi Publisher	2008
BTI-002-15	Pemrograman Visual	30	Karnel Delarose, Ph.D.	Informatika	2005
BTI-003-15	Konsep Pemrograman	40	Dr. Suarga, M.Sc, M.Match	Andi Publisher	2012
BTI-004-15	Panduan Lengkap MS Access 2013	30	Haer Talib	PT. Elex Media Komputindo	2014
BTI-005-15	Praktis Belajar Database dengan MS Access	10	Abdul Kadir	Andi Publisher	2002
BTI-006-15	Web Programming	28	Muhammad Sadeli	Maxikom	2009
BTI-007-15	Access 2013 untuk Orang Awam	35	Muhammad Sadeli	Maxikom	2013
BTI-008-15	Image Processing	20	Muhammad Sadeli	Maxikom	2012
BTI-011-15	Organisasi dan Arsitektur Komputer Ed. 6	35	William Stallings	PT. Indeks Kelompok Gramedia	2003
BTI-012-15	Sistem Basis Data	29	Adi Nugroho	Andi Publisher	2011



3. Buatlah tabel baru dengan nama **Tbl_Anggota** pada **Database Perpustakaan** dengan struktur data sebagai berikut:

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
IdAnggota	<i>Short Text</i>	6
NamaAnggota	<i>ShortText</i>	70
JenisKelamin	<i>ShortText</i>	1
TempatLahir	<i>ShortText</i>	150
TanggalLahir	<i>Date/Time</i>	
Alamat	<i>ShortText</i>	255
Pendidikan	<i>ShortText</i>	5
Handphone	<i>ShortText</i>	12
Email	<i>Hyperlink</i>	

4. Untuk *field* **IdAnggota** jadikan sebagai **Primary Key**.
 5. Untuk *field* **IdAnggota** masukkan **Input Mask: AG-1-000**.
 6. Untuk *field* **Nama Anggota** masukkan **Default Value: 70**.
 7. Untuk *field* **JenisKelamin** masukkan **Lookup** manual dengan aturan pada **Lookup** sebagai berikut:

Display Control : Combo Box

Row Source Type : Value List

Column Count : 1

Column Widths : 1;2

Limit To List : Yes

Sebagai pilihan data (*lookup*) masukkan **Row Source:**

a. **L** sebagai **Laki-laki**

b. **P** sebagai **Perempuan**.

8. Untuk *field* **TanggalLahir** masukkan **Format: Long Date**.
 9. Untuk *field* **Pendidikan** masukkan **Lookup** manual dengan aturan pada **Lookup** sebagai berikut:

Display Control : Combo Box

Row Source Type : Value List

Column Count : 1

Column Widths : 2

Limit To List : Yes



Sebagai pilihan data (*lookup*) masukkan **Row Source**:

- a. **TK**
- b. **SD**
- c. **SMP**
- d. **SMA**
- e. **D3**
- f. **S1**
- g. **S2**
- h. **S3**
- i. **Lain-lain**

10. Isikan data sebagai berikut:

IdAnggot.	NamaAnggota	JenisKelami	TempatLahir	TanggalLahir	Alamat	Pendidikan	Handphone	Email
AG-1-001	Muhammad William	L	Surabaya	16 Nopember 1994	Surabaya	S2	08987788900	william.muhammad@gmail.com
AG-1-002	Noor Chaterine	P	Malang	16 Agustus 1994	Surabaya	S2	08987272720	chaterine.noor@gmail.com
AG-1-003	Abdul Muthalib	L	Surabaya	13 September 1977	Surabaya	S1	09281233221	muthalib@yahoo.com
AG-1-004	Rikas Harsa	L	Jakarta	29 Juli 1982	Surabaya	S2	081325611771	rikasmen@gmail.com
AG-1-005	Richard Edward	L	London	09 September 1985	Surabaya	S3	099999181111	rich.edward@gmail.com
AG-1-006	Firman Utina	L	Lombok	09 September 1986	Surabaya	SMA	092881827881	firnut@yahoo.co.id
AG-1-007	Sri Ummi Wesi Tegai	P	Tegai	09 Maret 1978	Surabaya	D3	092828812344	sritegai@gmail.com



RELATIONSHIP



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu mengatur hubungan antar *table* dengan *relationship*
2. Mahasiswa mampu mengetahui *relationship type*
3. Mahasiswa mampu mengetahui *join type*



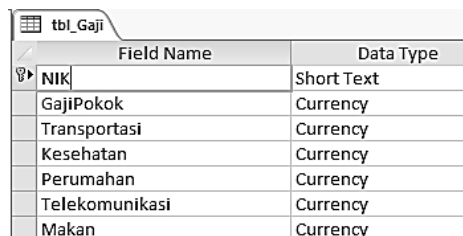
7.1 Relationship

Database dengan kapasitas kecil cukup menggunakan sebuah tabel saja. *Database* seperti ini disebut dengan istilah **Flat-File Database**. Daftar pada suatu aplikasi *spreadsheet* seperti Microsoft Office Excel dapat dipandang sebagai suatu *Flat-File Database*. Sedangkan untuk suatu sistem *database* yang baik dan andal, organisasi data harus melibatkan lebih dari satu tabel, sebuah tabel utama yang merupakan tabel induk dan beberapa tabel pendukung. Karena antar tabel yang satu dengan tabel yang lainnya berkaitan, maka secara fisik harus dibentuk suatu hubungan di antaranya.

Hubungan yang terjadi pada suatu tabel dengan tabel yang lainnya disebut dengan **Relationship/Relasi** yang berfungsi untuk mengatur operasi dalam suatu *database*. Hubungan yang terbentuk bisa bermacam-macam yaitu **one-to-one**, **one-to-many**, **many-to-many**. Relasi ini tidak harus dibuat, namun jika dibuat akan memberikan manfaat sebagai berikut:

1. Memudahkan dalam pembuatan *query*, dimana relasi antar tabel otomatis mengikuti definisi *relationship*.
2. Memungkinkan kita menerapkan **Referential Integrity**, yaitu aturan referensi untuk menjaga integritas/kesatuan data.
3. Memudahkan bagi kita untuk mengenal atau mengingat kunci-kunci yang menghubungkan data antar tabel, karena definisi *relationship* bisa dilihat secara visual dan tersimpan.

Sebelum membuat *relationship*, sebagai contoh buka *database* Karyawan kemudian buat satu tabel lagi dengan nama **tbl_Gaji** dengan struktur data sebagai berikut:



Field Name	Data Type
NIK	Short Text
GajiPokok	Currency
Transportasi	Currency
Kesehatan	Currency
Perumahan	Currency
Telekomunikasi	Currency
Makan	Currency



General		Lookup
Field Size	10	
Format		
Input Mask	"KYW-000\00\00	
Caption		
Default Value		
Validation Rule		
Validation Text		
Required	Yes	
Allow Zero Length	Yes	
Indexed	Yes (No Duplicates)	
Unicode Compression	Yes	
IME Mode	No Control	
IME Sentence Mode	None	
Text Align	General	

Gambar 7.1 Struktur Data Tabel tbl_Gaji

Isikan datanya sebagai berikut:

tbl_Gaji	NIK	GajiPokok	Transportas	Kesehatan	Perumahan	Telekomuni	Makan
	KYW-001-00-11	Rp3.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000
	KYW-002-12-11	Rp1.200.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000
	KYW-003-10-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000
	KYW-004-10-11	Rp1.200.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000
	KYW-005-07-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000
	KYW-005-08-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000

Gambar 7.2 Data Tabel tbl_Gaji

Maka kita sudah memiliki 3 buah tabel dalam *database* Karyawan.

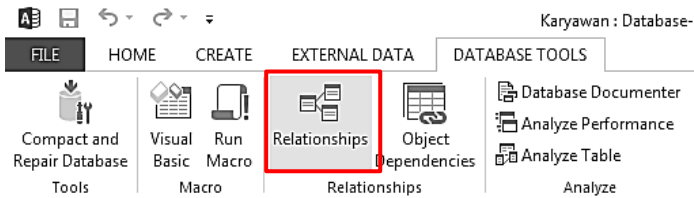


Gambar 7.3 Menu Tables



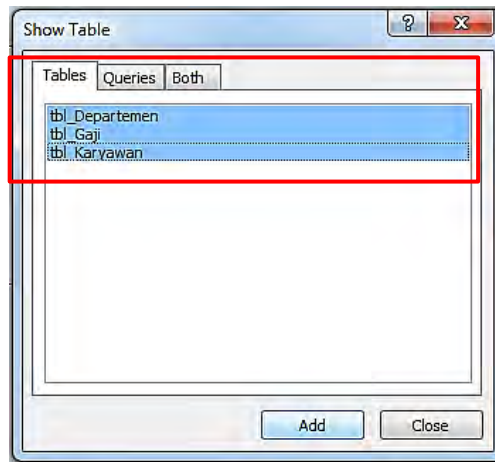
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Sekarang kita akan membuat **Relationship** pastikan semua tabel telah tidak aktif/tertutup → klik **Ribbon Tab Database Tools** → klik **Relationship**.



Gambar 7.4 Membuat **Relationship**

Muncul kotak dialog **Show Table** klik **Tab Tables** → pilih tabel-tabel yang direlasikan → klik **Add**.

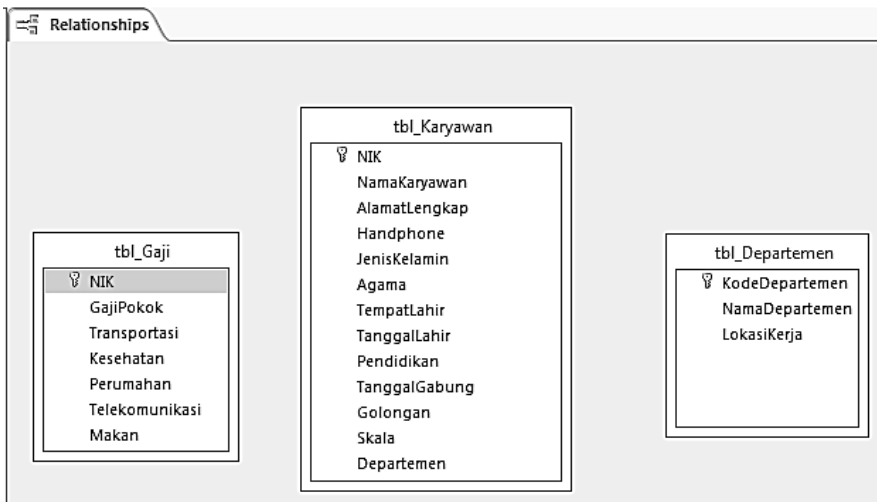


Gambar 7.5 Menambahkan Tabel ke dalam **Relationship** Melalui **Show Table**

Tips: anda bisa mengklik ganda tabel dalam kotak **Show Table** untuk langsung memasukkannya ke dalam area **relationship**, atau memilih



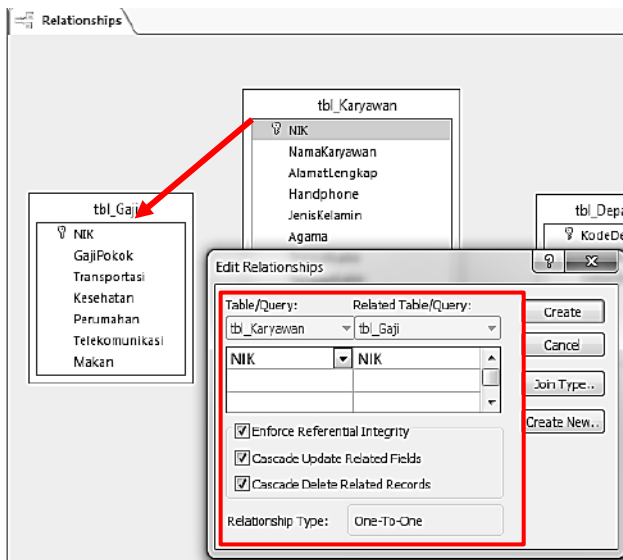
tabel dan mengklik *Add* untuk menambahkannya. Cara lain yang lebih mudah adalah dengan menyeret tabel langsung dari *Navigation Pane* ke dalam area *relationship*.



Gambar 7.6 Tampilan Area *Relationship* Setelah Penambahan Tabel

Untuk membuat relasi antar tabel, **caranya:** klik dan seret (*drag*) *field* yang ingin direlasikan kemudian jatuhkan (*drop*) di atas *field* yang ditujukan. Misalnya di sini klik dan seret *field* NIK dari **tbl_Karyawan** lalu lepaskan di atas *field* NIK dari **tbl_Gaji**. Kemudian muncul kotak dialog *Edit Relationship*.

Pilih dan centang *Enforce Referensial Integrity*, *Cascade Update Related Fields*, dan *Cascade Delete Related Records* → klik *Join Type*.



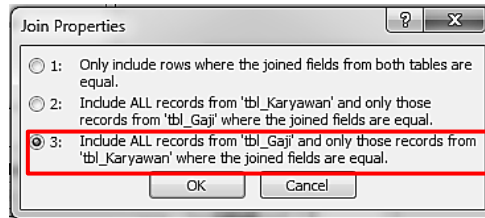
Gambar 7.7 Area Relationship dan Kotak Edit Relationship

Perlu diketahui:

1. **Enforce Referential Integrity** berfungsi untuk menjaga integritas data, memaksakan adanya suatu hubungan keterkaitan antar dua buah tabel.
2. **Cascade Update Related Fields** berfungsi agar Microsoft Office Access 2013 secara otomatis mengubah nilai/data pada *field* yang terkait jika nilai/data *field* pada **tabel master berubah**, tabel master pada contoh ini adalah tabel **tbl_Karyawan**. Pada contoh di atas misalnya jika nilai/data **NIK** pada **tbl_Karyawan berubah** maka nilai/data **NIK** pada **tbl_Gaji** juga ikut berubah secara otomatis, sehingga menghasilkan kesesuaian data tetap terjamin.
3. **Cascade Delete Related Records** berfungsi jika kita menghapus *record* yang berada pada **tbl_Karyawan**, maka *record* yang terkait yang berada pada **tbl_Gaji** juga ikut terhapus. Hal ini perlu agar tidak ada *record* pada **tbl_Gaji** yang terisolasi.



Muncul kotak dialog *Join Properties* dan pilih/klik opsi 3.



Gambar 7.8 Kotak Dialog *Join Properties*

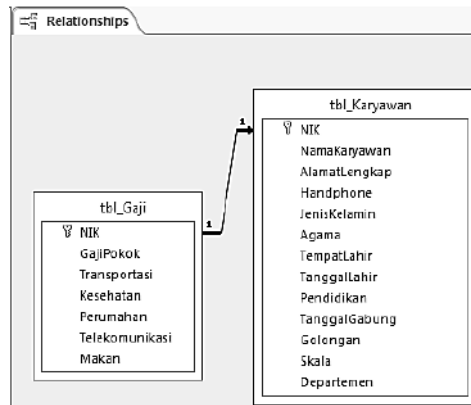
Perlu diperhatikan:

Pada *Join Properties*, opsi sebagai berikut:

1. Untuk membuat *inner join*, yaitu menampilkan *record* yang cocok pada kedua tabel.
2. Untuk menampilkan semua *record* pada **tbl_Karyawan** dan hanya *record* yang cocok pada **tbl_Gaji**.
3. Untuk menampilkan semua *record* pada **tbl_Gaji** dan hanya *record* yang cocok pada **tbl_Karyawan**.

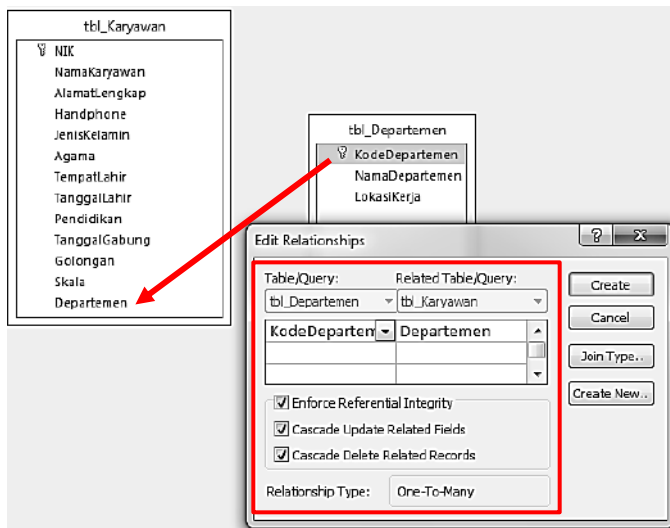
Join Type adalah penggabungan dua buah *table/query*, baik dengan menghubungkannya ataupun tidak menghubungkannya. Berikut tipe-tipe *join* yang disediakan di dalam Microsoft Office Access 2013 sebagai berikut:

1. **Inner Join** yaitu *join* yang menghubungkan dua *table/query* berdasarkan kunci (*field*) tertentu di mana dihasilkan *record-record* yang cocok dengan keduanya. *Type join* ini merupakan *type join default* yang sering digunakan.
2. **Outer Join** yaitu penggabungan yang menampilkan semua *record* pada satu sisi dan hanya menampilkan *record* yang cocok dari sisi lain.
3. **Cross Join** yaitu tidak menghubungkan kedua tabel, sehingga hasilnya adalah kombinasi dari semua *record* pada tabel pertama dengan semua *record* pada tabel kedua.

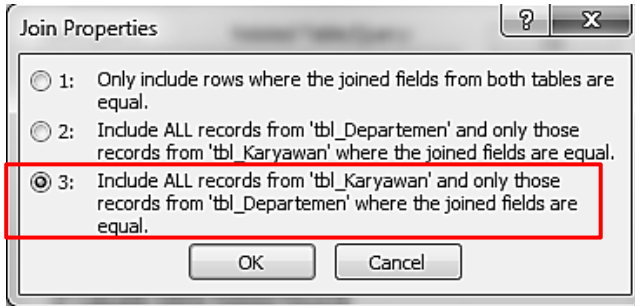


Gambar 7.9 Hasil Relasi *one-to-one* Antara **tbl_Gaji** – **tbl_Karyawan**

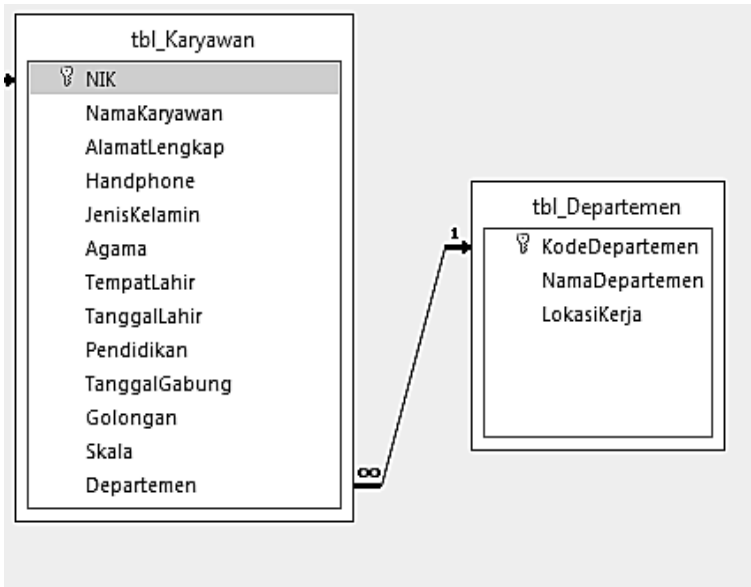
Lakukan relasi juga dengan cara yang sama pada *field* **KodeDepartemen** pada **tbl_Departemen** dengan *field* **Departemen** pada **tbl_Karyawan**. Agar hasilnya sebagai berikut:



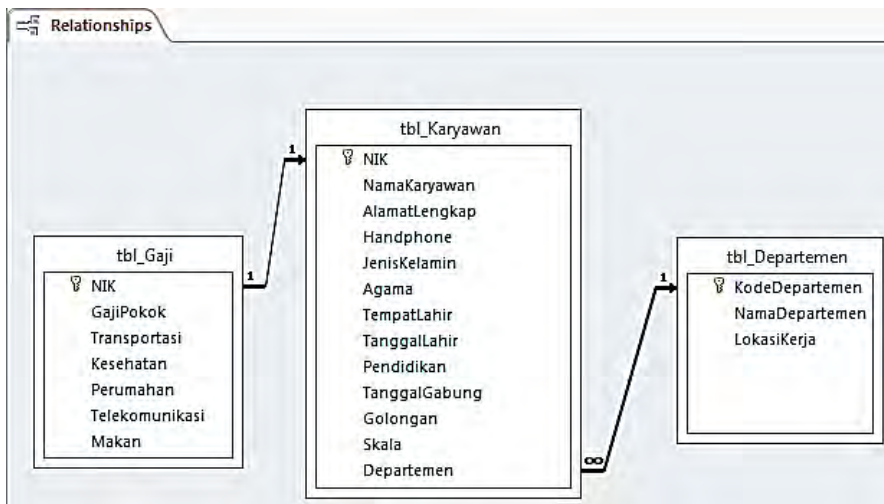
Gambar 7.10 Area Relationship dan Kotak Edit Relationship



Gambar 7.11 Kotak Dialog *Join Properties*



Gambar 7.12 Hasil Relasi *one-to-many* Antara *tbl_Karyawan* – *tbl_Departemen*



Gambar 7.13 Relationship

Kemudian tekan **Ctrl+S** untuk menyimpan hasil *relationship*, kemudian kita akan mencoba hasilnya. Misalnya aktifkan **tbl_Karyawan** → ganti data *field* **NIK** pada *record* ke-1 menjadi **KYW-001-10-11** → tekan **Ctrl+S** → klik *Close*.

	NIK	NamaKaryawan	AlamatLeng	Handphc
+	KYW-001-10-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143666
+	KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeruk	982919111
+	KYW-003-10-11	I Made Idha Nuantara	Jl. Kebon Sirih	992871722
+	KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jal	828919189
+	KYW-005-07-11	I Nyoman Windu Asn	Jl. Didina 9 Jak	928282827
+	KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subro	918181818

Gambar 7.14 Datasheet View Tabel **tbl_Karyawan**

Kemudian cek dan aktifkan **tbl_Gaji** dan cek apakah telah terganti data *field* **NIK** yang tadinya **KYW-001-00-11** menjadi **KYW-001-10-11**.



tbl_Gaji			
	NIK	GajiPokok	Transportas
+	KYW-001-10-11	Rp3.000.000	Rp500.000
+	KYW-002-12-11	Rp1.200.000	Rp500.000
+	KYW-003-10-11	Rp2.000.000	Rp500.000
+	KYW-004-10-11	Rp1.200.000	Rp500.000
+	KYW-005-07-11	Rp2.000.000	Rp500.000
+	KYW-005-08-11	Rp2.000.000	Rp500.000

Gambar 7.15 Datasheet View Tabel tbl_Gaji

Cek lagi untuk **tbl_Departemen** → ganti data *field* NamaDepartemen pada *record* ke-1 menjadi **Produksi Saja** → tekan **Ctrl+S** → klik **Close**.

tbl_Departemen			
	KodeDepart	NamaDepar	LokasiKerja
+	1	Produksi Saja	Cikarang
+	2	Pemasaran	Jakarta
+	3	Personalia	Jakarta
+	4	Keuangan	Jakarta

Gambar 7.16 Datasheet View Tabel tbl_Departemen

Kemudian cek dan aktifkan **tbl_Karyawan** dan cek apakah telah terganti data dari *field* Departemen yang tadinya **Produksi** menjadi **Produksi Saja**.

tbl_Karyawan													
	NIK	NamaKaryawan	AlamatLeng	Handph	JenisKelamin	Agama	TempatLahi	TanggalLahi	Pei	TanggalGabun	Golo	Skal	Departemen
+	KYW-001-10-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143666	Laki-laki	Islam	Surabaya	28-Feb-80 S-1		29/01/2000	III A		1 Produksi Saja
+	KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeru	982919111	Perempuan	Islam	Bandung	15-Sep-90 S-1		01/01/2012	III A		1 Pemasaran
+	KYW-003-10-11	I Made Idha Nuantari	Jl. Kebon Sirih	992871722	Perempuan	Hindu	Nusa Dua Bali	06-Jun-89 S-1		08/08/2010	III A		1 Produksi Saja
+	KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jal	828919189	Laki-laki	Kristen	Medan	09-Sep-88 S-1		08/08/2010	III A		1 Pemasaran
+	KYW-005-07-11	I Nyoman WIndu Asn	Jl. Didina 9 Jak	928282827	Laki-laki	Hindu	Nusa Dua Bali	09-Agust-89 S-1		09/01/2007	III A		2 Produksi Saja
+	KYW-005-08-11	Lee Xing Wee	Jl. Gator Subro	918181818	Laki-laki	Budha	Korea	16-Agust-87 S-1		09/01/2008	III A		2 Personalia

Gambar 7.17 Datasheet View Tabel tbl_Karyawan



Latihan Praktik

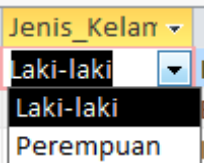
1. Buka kembali *database* **Sistem Informasi Kepegawaian** pada praktikum sebelumnya.
2. Buatlah tabel dengan nama **Jabatan** dengan struktur data sebagai berikut:

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
Kode_Jabatan	<i>Number</i>	<i>Long Integer</i>
Nama_Jabatan	<i>Short Text</i>	255
Tempat_Bekerja	<i>Short Text</i>	255

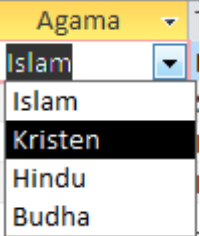
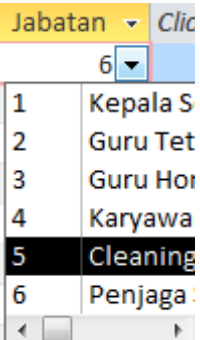
3. Jadikan *field* **Kd_Jabatan** sebagai **Primary Key**.
4. Isikan data pada tabel **Jabatan** sebagai berikut:

Kode_Jabatan	Nama_Jabatan	Tempat_Bekerja
1	Kepala Sekolah	SMKN 1 Jakarta
2	Guru Tetap	SMKN 1 Jakarta
3	Guru Honor	SMKN 1 Jakarta
4	Karyawan	SMKN 1 Jakarta
5	Cleaning Service	SMKN 1 Jakarta
6	Penjaga Sekolah	SMKN 1 Jakarta

5. Buatlah tabel relasi dengan nama **Pegawai** dengan struktur data sebagai berikut:

<i>Field Name</i>	<i>Data Type</i>	<i>Field Size</i>
NIP	<i>Short Text</i>	18
Nama_Pegawai	<i>Short Text</i>	50
Jenis_Kelamin	<i>Short Text</i>	255 Buat <i>lookup</i> dengan <i>Row Source</i> manual: Laki-laki Perempuan 
Agama	<i>Short Text</i>	255 Buat <i>lookup</i> dengan <i>Row Source</i> manual:



Field Name	Data Type	Field Size
		Islam Kristen Hindu Budha 
Tempat Lahir	Short Text	255
Tanggal Lahir	Date/Time	Format: Short Date
No HP	Short Text	12
Alamat	Short Text	255
Mulai Bekerja	Date/Time	Format: Short Date
Golongan	Short Text	4
Jabatan	Number	Long Integer Buat lookup dengan Row Source dari Tabel Jabatan 

6. Jadikan *field* NIP sebagai **Primary Key**.
7. Isikan data pada tabel **Pegawai** sebagai berikut:



NIP	Nama_Pegawai	Jenis_Kelamin	Agama	Tempat_Lahir	Tanggal_Lahir	No_HP	Alamat	Mulai_Bekerja	Golongan	Jabatan
196501021989011002	Hj. Sri Lestari, M.Pd.	Perempuan	Islam	Bandung	02/03/1985	087819192020	Jakarta	03/03/1980	IWA	2
197702041999011001	Christian Aditia, S. Kom. M.T.	Laki-laki	Kristen	Jakarta	04/02/1977	085219891011	Jakarta	01/01/1990	IIIA	2
196401041987011001	Muhammad Aznannor, M.Pd.	Laki-laki	Islam	Banjarasin	15/11/1994	08795565778	Jakarta	01/03/1985	IWA	1
198705052007011002	Noor Manwa, S.Pd.	Perempuan	Islam	Jakarta	05/05/1987	089876767889	Jakarta	01/01/2007	IIIA	3
198309072000011002	I Kadek Chaterine Windu, S.E.	Perempuan	Hindu	Bali	07/09/1983	081350557889	Jakarta	01/01/2004	IIA	4
196012101985011001	Uddin Maholgar	Laki-laki	Budha	Surabaya	10/12/1960	087899191910	Jakarta	01/01/1980	IIIA	5
195811071985021001	H. Maulidi Sofyan	Laki-laki	Islam	Banjarasin	07/10/1958	0812096091100	Jakarta	01/01/1979	IIIA	6

8. Buatlah tabel dengan nama **Gaji** dengan struktur data sebagai berikut:

Field Name	Data Type	Field Size
NIP	Short Text	18 Buat <i>lookup</i> dengan <i>Row Source</i> dari Tabel Pegawai
Gj Pokok	Currency	Format: Rp.
Tj Transportasi	Currency	Format: Rp.
Askes	Currency	Format: Rp.
Tj Perumahan	Currency	Format: Rp.
Tj Telekomunikasi	Currency	Format: Rp.
Tj Makan	Currency	Format: Rp.
Tj Jabatan	Currency	Format: Rp.

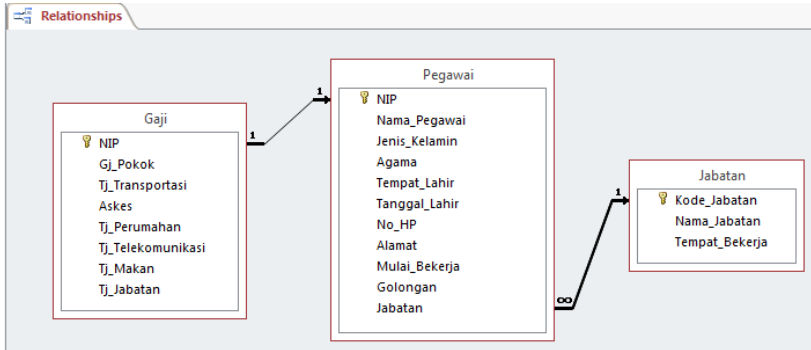
9. Jadikan *field* **NIP** sebagai **Primary Key** dan **Foreign Key**.

10. Isikan data pada tabel **Gaji** sebagai berikut:

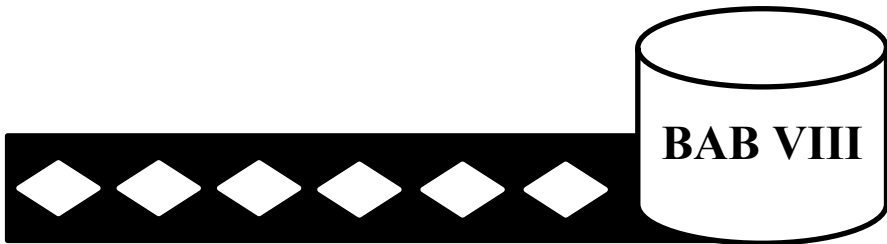


	NIP	Gj_Pokok	Tj_Transpor	Askes	Tj_Perumah	Tj_Telekom	Tj_Makan	Tj_Jabatan
☐	195811071985021001	Rp1.200.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp0
☐	196401041987011001	Rp3.000.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp500.000
☐	196501021989011002	Rp2.700.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp0
☐	197702041999011001	Rp2.000.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp0
☐	198309072000011002	Rp2.000.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp0
☐	198705052007011002	Rp1.500.000	Rp500.000	Rp230.000	Rp650.000	Rp270.000	Rp600.000	Rp0

11. Buatlah *relationship* sebagai berikut:
 - a. Dimana antara tabel **Gaji** dngan tabel **Pegawai** berelasi **One-To-One**
 - b. Dimana antara tabel **Pegawai** dengan tabel **Jabatan** berelasi **One-To-Many**



12. Ganti data pada *field* **NIP** atas nama **Noor Mazwa, S.Pd.** pada tabel **Pegawai** ganti dengan **198705052007011003**, kemudian cek hasilnya pada *field* **NIP** di tabel **Gaji** apakah ikut terganti.
13. Ganti data pada *field* **Nama_Jabatan** atas jabatan **Karyawan** pada tabel **Jabatan** dengan **Karyawan TU**, kemudian cek hasilnya pada *field* **Jabatan** di tabel **Pegawai** apakah ikut terganti.
14. Hapuslah *record* atas nama **Uddin Maholgar** pada tabel **Pegawai**, kemudian cek di tabel **Gaji** apakah masih ada data atas nama **Uddin Maholgar**.



MEMBUAT *QUERY*



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu membuat *query*
2. Mahasiswa mampu menjalankan *query*
3. Mahasiswa mampu memberikan *criteria* pada *query*



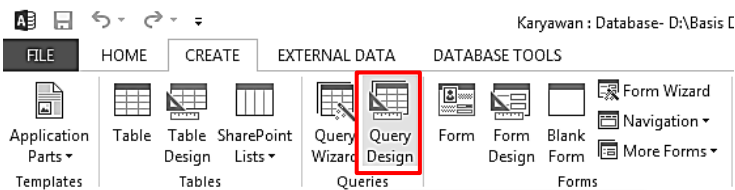
8.1 Query

Query adalah sarana untuk mengatur data yang disimpan dalam tabel, sehingga hanya data-data tertentu yang akan dimunculkan dalam tabel. Secara fisik *query* berupa tabel juga, jadi *query* adalah tabel yang dibuat dari beberapa tabel lainnya dengan tujuan untuk mengelompokkan data baru berupa tabel, dengan sumber dari tabel-tabel yang sudah ada. Dengan bantuan *query*, kita dapat menyortir data atau untuk menampilkan data-data yang memenuhi kriteria tertentu, atau juga dapat berupa suatu ekspresi.

Sekarang kita akan membuat *query* dengan **Query Design**, adalah alat untuk mendesain *query* secara visual. Dengan **Query Design** kita leluasa dalam mendesain *query*, tidak terbatas oleh langkah-langkah yang disediakan **Query Wizard**. Secara singkat pekerjaan membuat *query* dengan **Query Design** sebagai berikut:

1. Tambahkan *table* atau *query* yang akan dijadikan dasar atau sumber data.
2. Tentukan *join type* (jika menggabungkan beberapa *table* atau *query*).
3. Pilih *field* yang ingin ditampilkan dalam *query*.

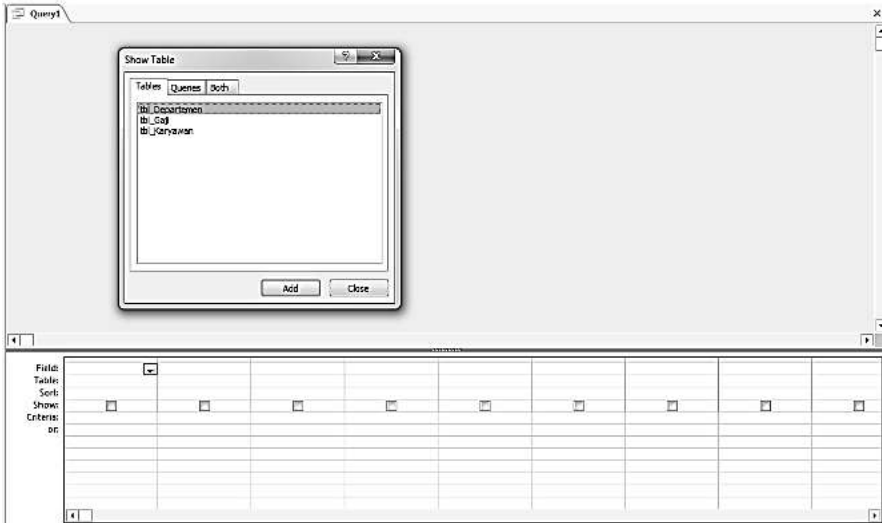
Pembuatan *query* sebenarnya tergantung pada jenis *query* yang ingin dibuat. Setiap jenis *query* mempunyai langkah-langkah pembuatannya tersendiri. Oleh karena itu, di dalam pembuatan *query* Anda perlu mengenal dengan baik jenis *query* dan karakteristiknya. Membuat *query* dengan **Query Design**, dengan cara: Klik **Ribbon Tab CREATE** → klik **Query Design**.



Gambar 8.1 Query Design pada Ribbon Tab CREATE

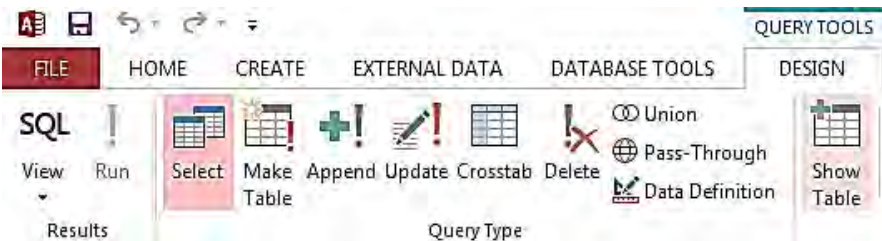


Muncul kotak dialog *Show Table* dari *Query Design*.



Gambar 8.2 Area *Query* dan Kotak Dialog *Show Table*

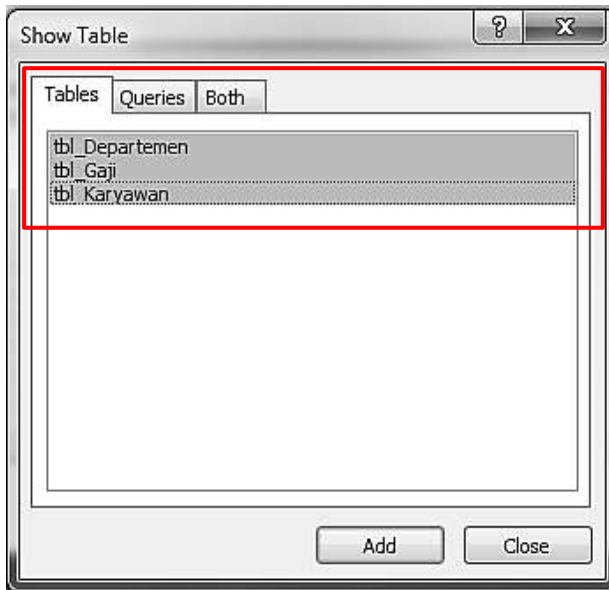
Jika kotak dialog *Show Table* tidak muncul dan ingin memunculkan kembali, pastikan masih berada dalam area kerja pembuatan *query*, klik *Ribbon Tab Design* → klik *Show Table*.



Gambar 8.3 Show Table pada Ribbon DESIGN

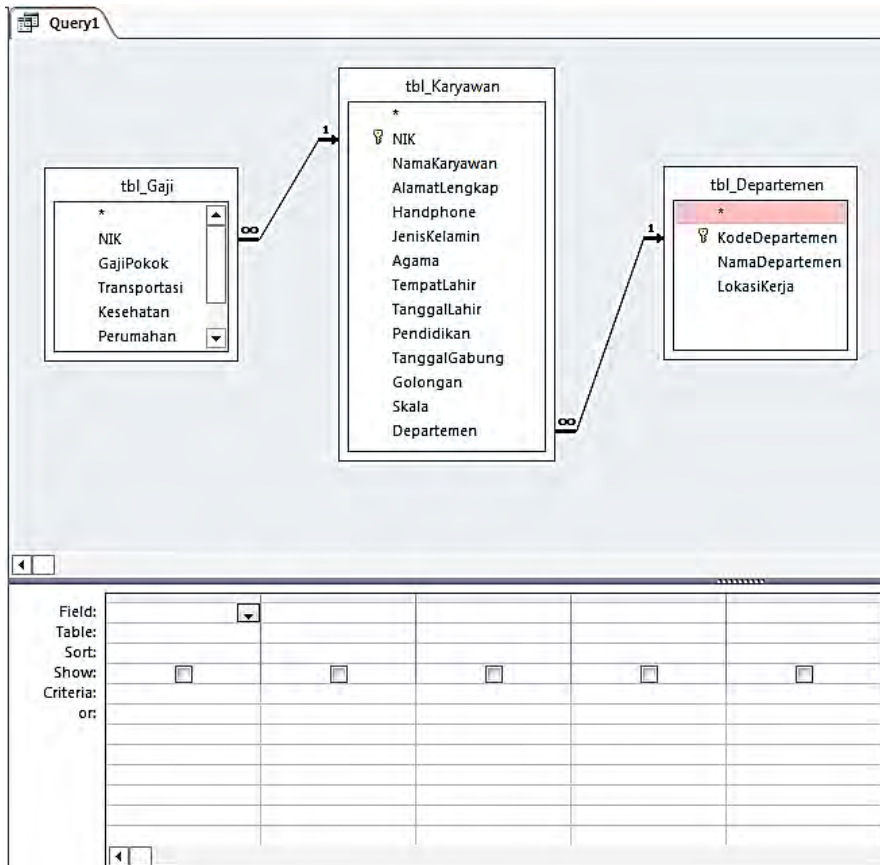


Pada kotak **Show Table** pilih *table* atau *query* yang ingin digunakan sebagai dasar *query*. Pada intinya membuat *query* ini tergantung dari pemakai sendiri apakah akan menyortir data atau menampilkan semua data dari *table* atau *query*. Sebagai contoh kita menampilkan semua data dari 3 buah tabel pada **Tab Tables** klik **tbl_Departemen**, **tbl_Gaji**, dan **tbl_Karyawan** → klik **Add**.



Gambar 8.4 Kotak Dialog **Show Table**

Kemudian secara otomatis akan muncul 3 buah tabel sekaligus relasi antar tabelnya, karena sudah diatur pada *relationship* sebelumnya.



Gambar 8.5 Area Query

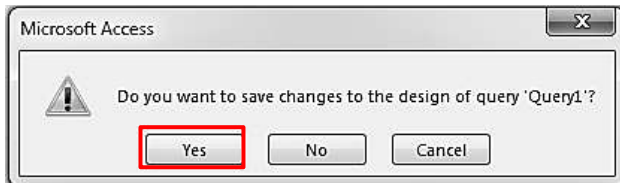
Cara memasukkan *field* ke dalam *query*: klik *field* misalnya *field* **NIK** pada **tbl_Gaji** → **Enter**. Untuk *field* yang dipilih selanjutnya lakukanlah tahapan seperti tadi. Dalam hal ini semua *field* kita masukkan ke dalam *query*.



tbl_Gaji_NIK	GajiPokok	Transportasi	Kesehatan	Perumahan	Telekomuni	Makan	tbl_Karyawan	NamaKaryawan	AlamatJeng	Handph	JenisJend
KYW-001-00-11	Rp3.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-001-00-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143665	Laki
KYW-002-12-11	Rp1.200.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeruk	902819111	Perempuan
KYW-003-10-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-003-10-11	I Made Ktha Muartari	Jl. Kebon Sirih	902817122	Perempuan
KYW-004-10-11	Rp1.200.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jal	828919189	Laki-laki
KYW-005-07-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-005-07-11	Nyoman Windu Asri	Jl. Didina 9 Jak	928282827	Laki-laki
KYW-005-06-11	Rp2.000.000	Rp500.000	Rp150.000	Rp500.000	Rp400.000	Rp1.000.000	KYW-005-06-11	Lee Xing Wee	Jl. Gatot Subro	918181818	Laki-laki

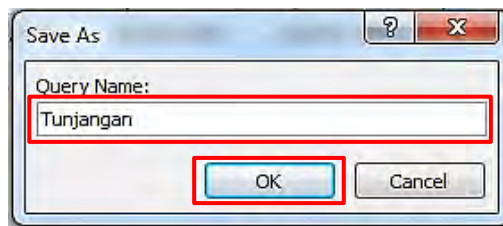
Gambar 8.8 Hasil Query

Setelah diklik *close* kemudian muncul kotak dialog dari Microsoft Office Access 2013 dimana untuk mengonfirmasi apakah *query* ini akan disimpan? → klik *Yes*.



Gambar 8.9 Kotak Dialog Konfirmasi Penyimpanan query

Kemudian muncul lagi kotak dialog *Save As* berikan nama *query* dengan **Tunjangan** → klik *OK*.



Gambar 8.10 Kotak Dialog *Save As*



Latihan Praktik

1. Buka kembali *database* **Sistem Informasi Kepegawaian**.
2. Buatlah *Query* dengan nama **Semua** → masukkan semua *field* yang ada pada tabel **Pegawai, Gaji, Jabatan** → tampilkan hasilnya.
3. Buatlah *Query* dengan nama **Awalan Nama Huruf H** → masukkan *field* **Nama_Pegawai** → masukkan *criteria* **Like H*** → tampilkan hasilnya.



4. Buatlah *Query* dengan nama **Akhiran Gelar M.Pd.** → masukkan *field* **Nama_Pegawai** → masukkan *criteria* **Like *M.Pd.** → tampilkan hasilnya.



5. Buatlah *Query* dengan nama **Hanya Perempuan** → masukkan *field* **Nama_Pegawai** dan **Jenis_Kelamin** → pada *field* **Jenis_Kelamin** masukkan *criteria* **Not Like Laki-laki** → tampilkan hasilnya.



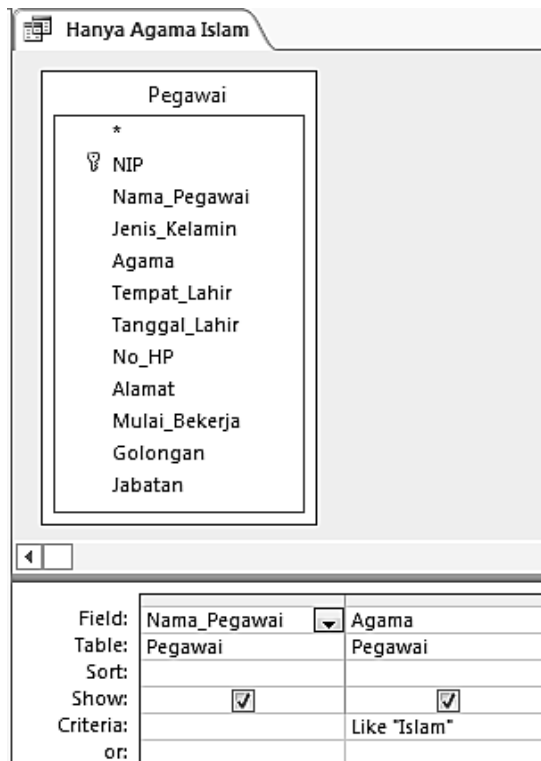
Hanya Perempuan

Pegawai

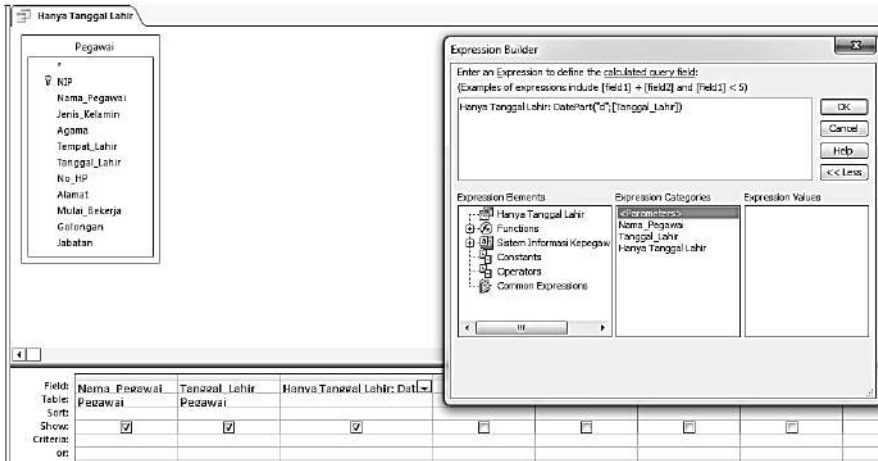
- *
 - NIP
 - Nama_Pegawai
 - Jenis_Kelamin
 - Agama
 - Tempat_Lahir
 - Tanggal_Lahir
 - No_HP
 - Alamat
 - Mulai_Bekerja
 - Golongan
 - Jabatan

Field:	Nama_Pegawai	Jenis_Kelamin
Table:	Pegawai	Pegawai
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Not Like "Laki-laki"
or:		

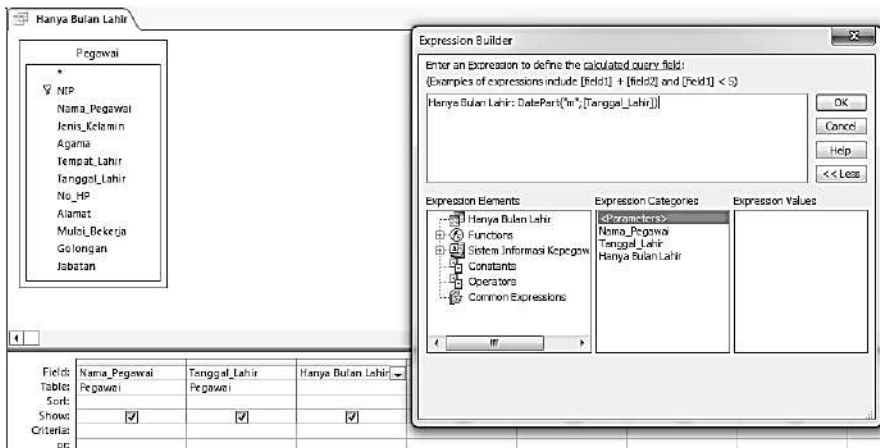
- Buatlah *Query* dengan nama **Hanya Agama Islam** → masukkan *field* **Nama_Pegawai** dan **Agama** → pada *field* **Agama** masukkan *criteria* **Like Islam** → tampilkan hasilnya.



7. Buatlah *Query* dengan nama **Hanya Tanggal Lahir** → masukkan *field* **Nama_Pegawai** dan **Tanggal Lahir** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Hanya Tanggal Lahir: DatePart("d";[Tanggal_Lahir])** → OK → tampilkan hasilnya.

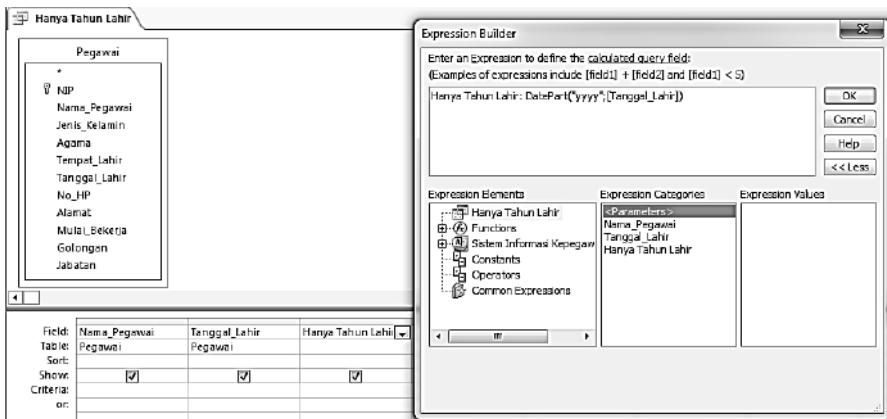


8. Buatlah *Query* dengan nama **Hanya Bulan Lahir** → masukkan *field* **Nama_Pegawai** dan **Tanggal Lahir** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Hanya Bulan Lahir: DatePart("m";[Tanggal_Lahir])** → OK → tampilkan hasilnya.

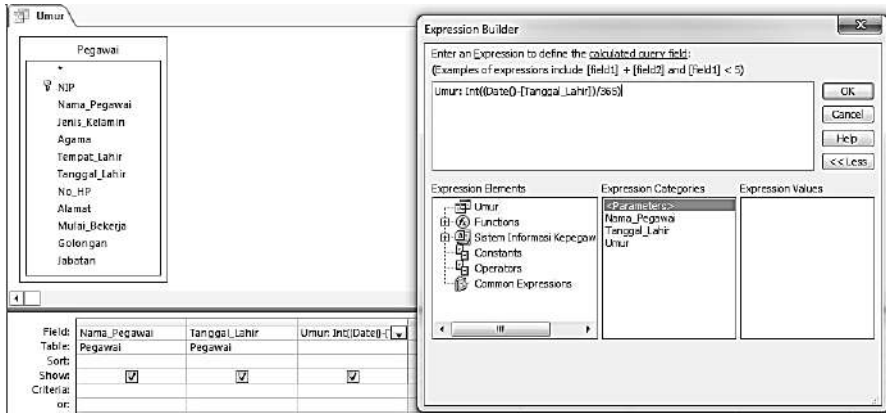




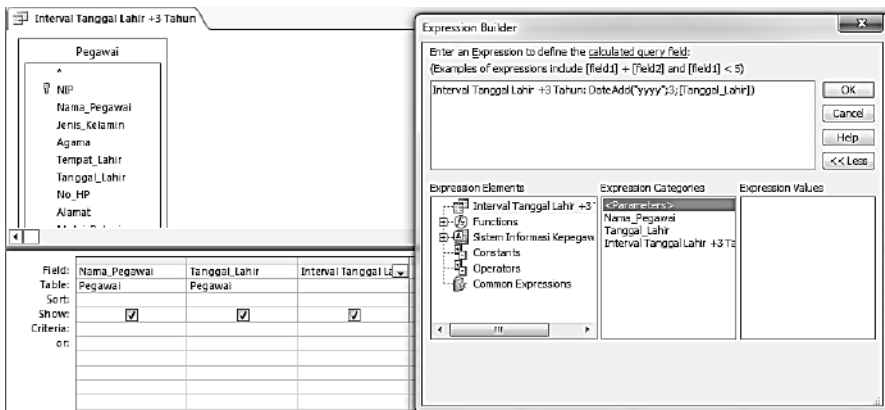
9. Buatlah *Query* dengan nama **Hanya Tahun Lahir** → masukkan *field* **Nama_Pegawai** dan **Tanggal Lahir** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Hanya Tahun Lahir: DatePart("yyyy";[Tanggal_Lahir])** → **OK** → tampilkan hasilnya.



10. Buatlah *Query* dengan nama **Umur** → masukkan *field* **Nama_Pegawai** dan **Tanggal Lahir** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Umur: Int((Date()-[Tanggal_Lahir])/365)** → **OK** → tampilkan hasilnya.



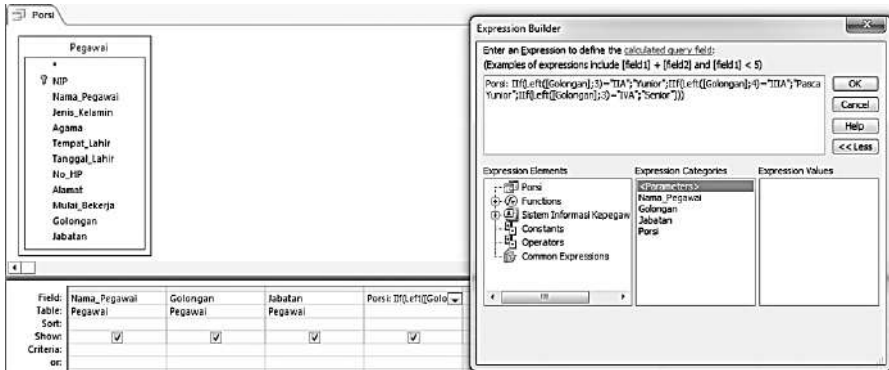
11. Buatlah *Query* dengan nama **Interval Tanggal Lahir +3 Tahun** → masukkan *field* **Nama_Pegawai** dan **Tanggal Lahir** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Interval Tanggal Lahir +3 Tahun: DateAdd("yyyy";3;[Tanggal_Lahir])** → OK → tampilkan hasilnya.



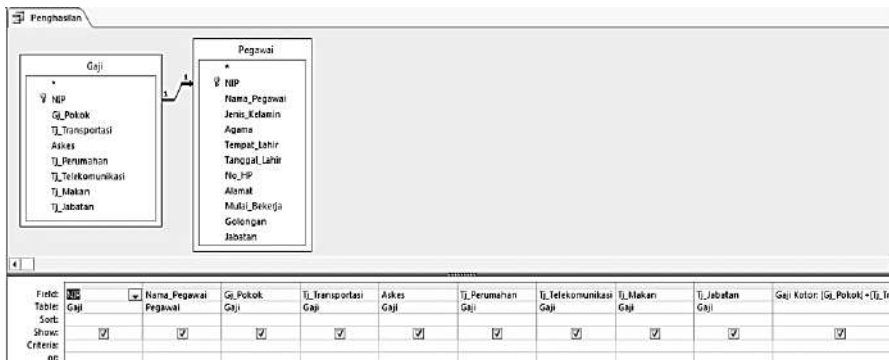
12. Buatlah *Query* dengan nama **Descending** → masukkan *field* **Nama_Pegawai** → kemudian pada bagian *Sort* pilih **Descending** → tampilkan hasilnya.

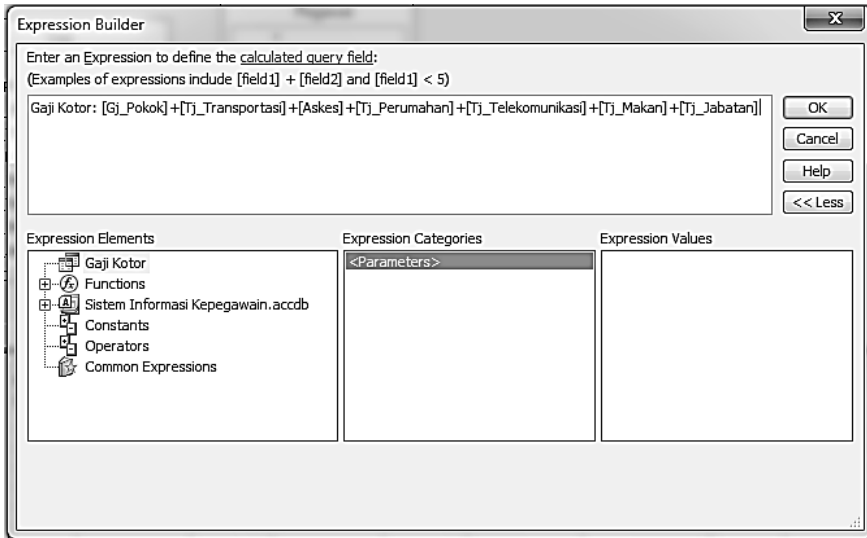


13. Buatlah *Query* dengan nama **Porsi** → masukkan *field* **Nama_Pegawai**, **Golongan**, dan **Jabatan** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Porsi: IIf(Left([Golongan];3)="IIA";"Yunior";IIf(Left([Golongan];4)="III A";"Pasca Yunior";IIf(Left([Golongan];3)="IVA";"Senior")))** → **OK** → tampilkan hasilnya.

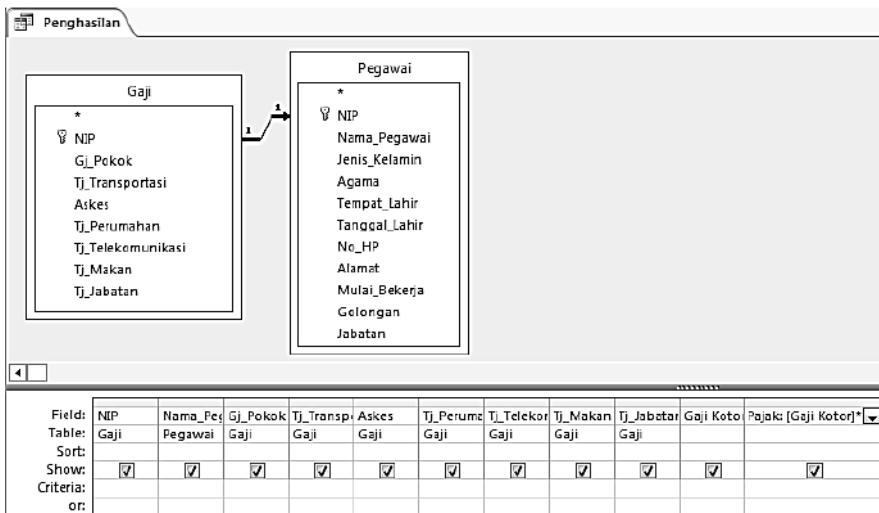


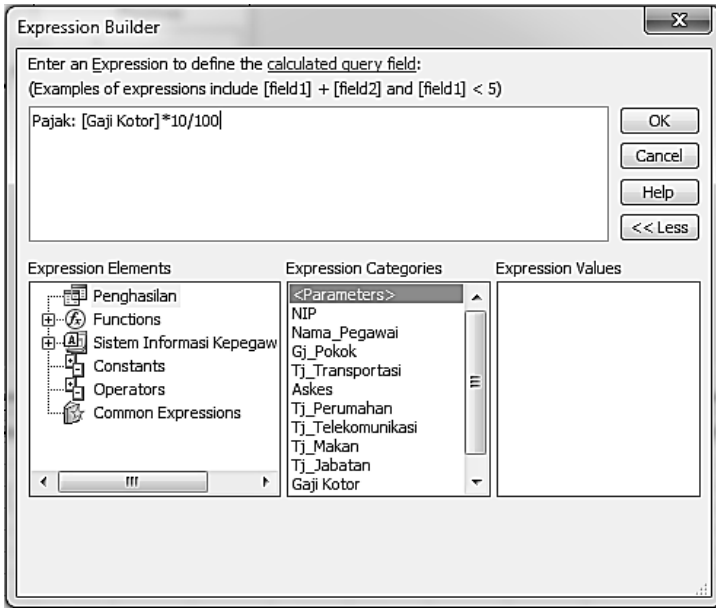
14. Buatlah *Query* dengan nama **Penghasilan** → masukkan *field* **NIP** dari **tbl_Gaji**, **Nama_Pegawai** dari **tbl_Pegawai**, **Gj_Pokok** dari **tbl_Gaji**, **Tj_Transportasi** dari **tbl_Gaji**, **Askes** dari **tbl_Gaji**, **Tj_Perumahan** dari **tbl_Gaji**, **Tj_Telekomunikasi** dari **tbl_Gaji**, **Tj_Makan** dari **tbl_Gaji**, dan **Tj_Jabatan** dari **tbl_Gaji** → kemudian klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Gaji Kotor**: **[Gj_Pokok]+[Tj_Transportasi]+[Askes]+[Tj_Perumahan]+[Tj_Telekomunikasi]+[Tj_Makan]+[Tj_Jabatan]** → **OK** → tampilkan hasilnya.



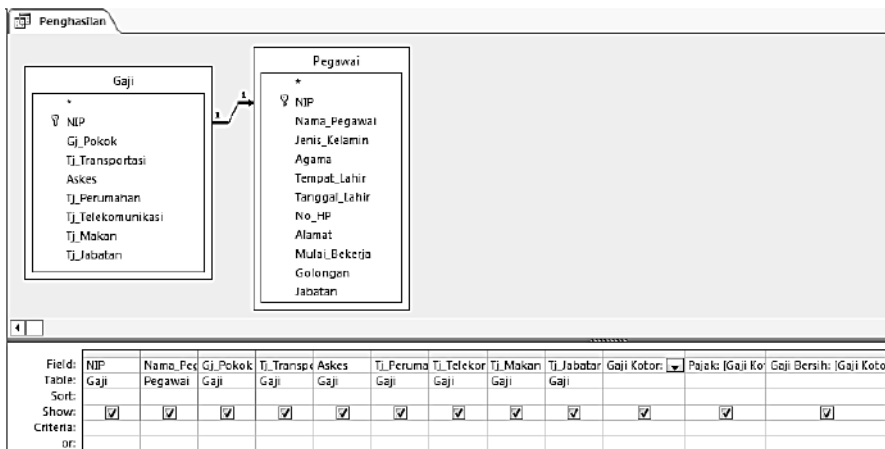


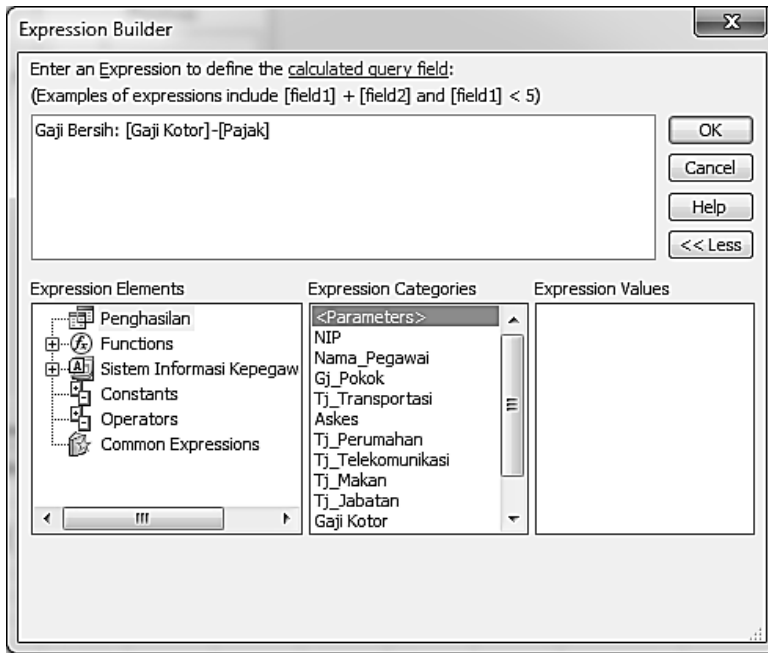
15. Masih pada *Query Penghasilan* → masuk ke dalam area *Design View* → kemudian di samping *field Gaji Kotor* klik *field* kosong dan klik kanan *field* kosong tersebut → pilih *Build* → masukkan rumus **Pajak: [Gaji Kotor]*10/100** → OK → tampilkan hasilnya.

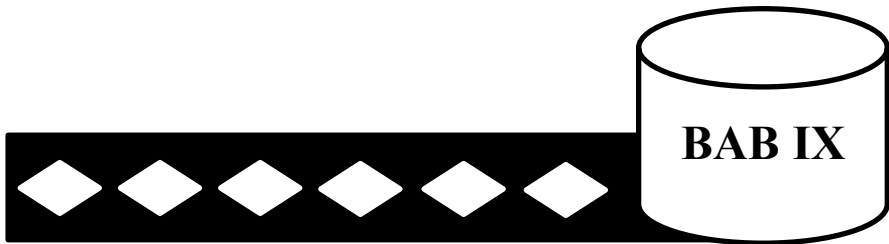




16. Masih pada *Query Penghasilan* → masuk ke dalam area **Design View** → kemudian di samping *field Pajak* klik *field* kosong dan klik kanan *field* kosong tersebut → pilih **Build** → masukkan rumus **Gaji Bersih: [Gaji Kotor]-[Pajak]** → OK → tampilkan hasilnya.







MENGIMPLEMENTASIKAN STUDI KASUS ERD MENJADI SEBUAH *DATABASE*



Tujuan Instruksional Umum :

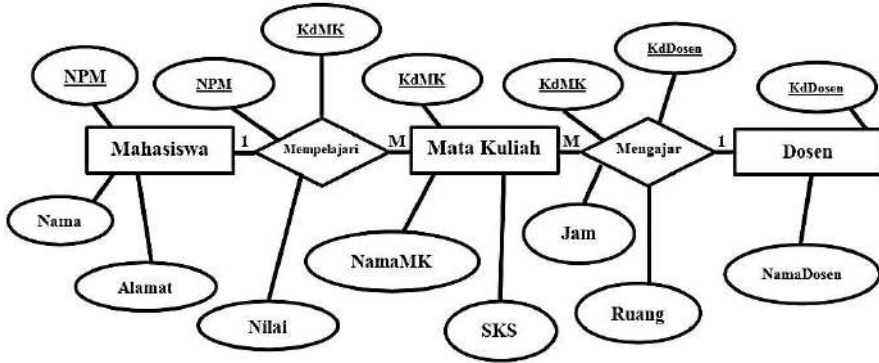
Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu memahami dan membuat ERD
2. Mahasiswa mampu memahami dan membuat struktur data secara konseptual dari ERD
3. Mahasiswa mampu memahami dan membuat *table* pada *database* Microsoft Office Access 2013 berdasarkan ERD
4. Mahasiswa mampu membuat *relationship* pada *database* Microsoft Office Access 2013 berdasarkan ERD
5. Mahasiswa mampu membuat *query* pada *database* Microsoft Office Access 2013 berdasarkan ERD



Dalam praktiknya ERD sebagai sebuah konseptual sebelum dijadikan sebuah Database di dalam Microsoft Office Access 2013. Berikut contohnya yaitu Database Akademik:



9.1 Memahami ERD

Langkah-langkah untuk memahami ERD sebagai berikut:

1. Mengetahui ada berapa buah entitas dan atribut
 - Entitas Mahasiswa = {NPM, Nama, Alamat}
 - Entitas Mata Kuliah = {KdMK, NamaMK, SKS}
 - Entitas Dosen = {KdDosen, NamaDosen}
2. Mengetahui ada berapa buah relasi dan atribut
 - Mempelajari = {NPM, KdMK, Nilai}
 - Mengajar = {KdDosen, KdMK, Jam, Ruang}

9.2 Membuat Struktur Data secara Konseptual dari ERD

Dari ERD dapat diperoleh informasi sebagai berikut:

1. Tabel Acuan Mahasiswa

Field Name	Keterangan	Data Type	Field Size
NPM	Primary Key (PK)	Short Text	10
Nama		Short Text	255
Alamat		Short Text	255



2. Tabel Acuan Mata Kuliah

Field Name	Keterangan	Data Type	Field Size
KdMK	Primary Key (PK)	Short Text	6
NamaMK		Short Text	255
SKS		Number	Long Integer

3. Tabel Acuan Dosen

Field Name	Keterangan	Data Type	Field Size
KdDosen	Primary Key (PK)	Short Text	5
NamaDosen		Short Text	255

4. Tabel Relasi Mempelajari

Field Name	Keterangan	Data Type	Field Size
NPM	Foreign Key (FK)	Short Text	10
KdMK	Foreign Key (FK)	Short Text	6
Nilai		Number	Long Integer

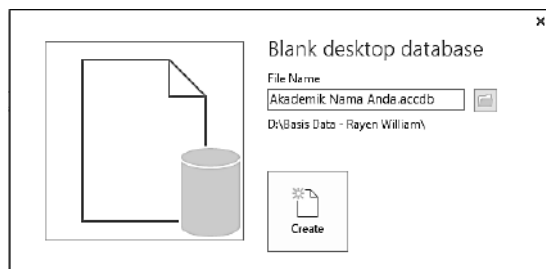
5. Tabel Relasi Mengajar

Field Name	Keterangan	Data Type	Field Size
KdDosen	Foreign Key (FK)	Short Text	5
KdMK	Foreign Key (FK)	Short Text	6
Jam		Date/Time	Format: Short Time
Ruang		Short Text	30

Latihan Praktik

9.3 Membuat Tabel pada Database Microsoft Office Access 2013 Berdasarkan ERD di Atas

1. Membuat Database dengan nama Akademik Nama Anda





2. Membuat Tabel Mahasiswa

Mahasiswa		
	Field Name	Data Type
🔑	NPM	Short Text
	Nama	Short Text
	Alamat	Short Text

3. Mengisi Tabel Mahasiswa

Mahasiswa			
	NPM	Nama	Alamat
+	3061546001	Azri Aulia Muhammad	Banjarmasin
+	3061546002	Riyanti Eka Prahesti	Banjarbaru
+	3061546003	Muhammad Yufhit	Martapura
+	3061546004	Daniel Dwi Rahardjo	Marabahan
+	3061546005	Kuanti Yesfita	Banjarmasin

4. Membuat Tabel Mata Kuliah

Mata Kuliah		
	Field Name	Data Type
🔑	KdMK	Short Text
	NamaMK	Short Text
	SKS	Number

5. Mengisi Tabel Mata Kuliah

Mata Kuliah			
	KdMK	NamaMK	SKS
+	BKDK1	Dasar-Dasar Kependidikan	3
+	BMKU1	Pendidikan Sosial dan Budaya Dasar	3
+	BTIK32	Basis Data	3
+	BTIK33	Jaringan Komputer	3
+	BTIK34	Algoritma dan Struktur Data	3



6. Membuat Tabel Dosen

Dosen	
Field Name	Data Type
KdDosen	Short Text
NamaDosen	Short Text

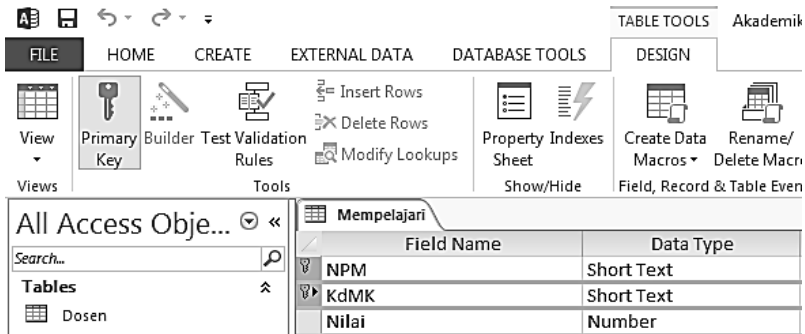
7. Mengisi Tabel Dosen

Dosen	
KdDosen	NamaDosen
+ Dos01	Subandi, S.T., M.Kom.
+ Dos02	Arifin Noor Asyikin, M.T.
+ Dos03	Kenti Yuliana, M.Kom.
+ Dos04	Drs. H. Riduan Saberan, M.Pd.
+ Dos05	Dr. Abidinsyah, M.Pd.

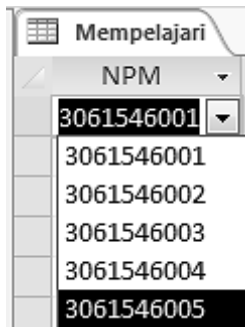
8. Membuat Tabel Mempelajari

Mempelajari	
Field Name	Data Type
NPM	Short Text
KdMK	Short Text
Nilai	Number

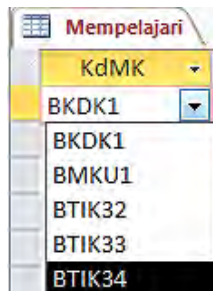
Tabel mempelajari merupakan Tabel relasi antara Tabel Mahasiswa dan Mata Kuliah, jadi ada 2 buah kunci yang sama-sama merupakan kunci tamu (*Foreign Key*). Untuk menjadikan *field* NPM dan KdMK menjadi *Foreign Key*: tekan tombol **Ctrl+Shift (jangan dilepas)** → klik *field* NPM dan KdMK → Tekan **Ribbon Primary Key** → lepaskan tombol **Ctrl+Shift**.



9. Atur NPM dengan *lookup* ke Tabel Mahasiswa, agar hasilnya seperti di bawah ini:



10. Atur KdMK dengan *lookup* ke Tabel Mata Kuliah, agar hasilnya seperti di bawah ini:



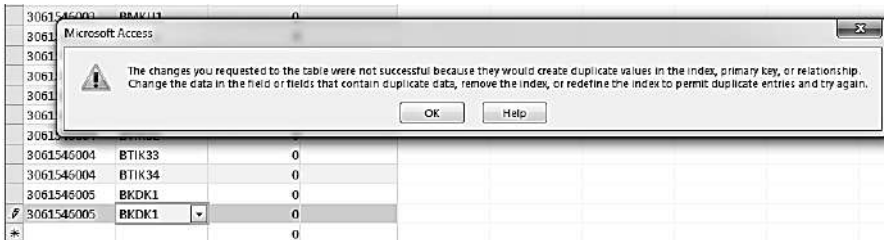


11. Mengisi Tabel Mempelajari

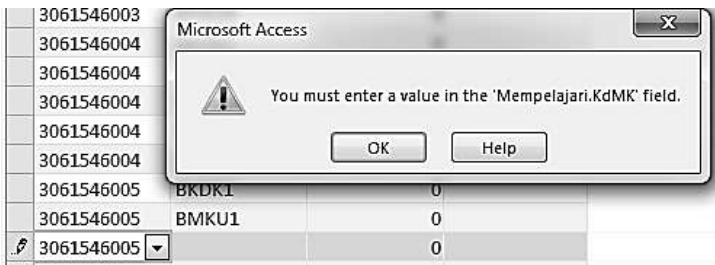
Mempelajari		
NPM	KdMK	Nilai
3061546001	BKDK1	60
3061546001	BMKU1	70
3061546001	BTIK32	80
3061546001	BTIK33	90
3061546001	BTIK34	70
3061546002	BKDK1	80
3061546002	BMKU1	79
3061546002	BTIK32	75
3061546002	BTIK33	75
3061546002	BTIK34	80
3061546003	BKDK1	80
3061546003	BMKU1	90
3061546003	BTIK32	89
3061546003	BTIK33	85
3061546003	BTIK34	80
3061546004	BKDK1	90
3061546004	BMKU1	80
3061546004	BTIK32	70
3061546004	BTIK33	70
3061546004	BTIK34	80
3061546005	BKDK1	70
3061546005	BMKU1	70
3061546005	BTIK32	65
3061546005	BTIK33	70
3061546005	BTIK34	65

Catatan:

- Ketika Anda mengisi data pada Tabel misalnya mempelajari, ketika ada data milik NPM **3061546005** dia sebelumnya **telah mengambil** Mata Kuliah dengan KdMK **BKDK1** kemudian **diambil lagi**, maka akan muncul kotak dialog **nilai data tidak boleh terduplikasi/2 kali muncul** karena sebagai **kunci** → klik **OK** → kemudian gantikan dengan KdMK yang belum dia ambil.



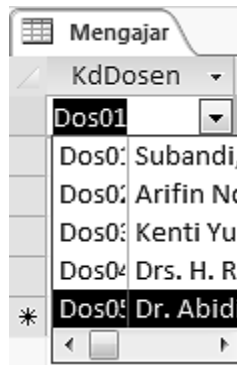
- b. Ketika Anda mengisi data pada Tabel misalnya mempelajari, ketika ada data milik NPM 3061546005 kemudian data KdMK belum diisikan dan mau beralih ke *record/field* yang lain maka akan muncul kotak dialog kamu harus mengisikan KdMK terlebih dahulu karena sebagai **nilai kunci yang harus diisi** → klik **OK** → kemudian isikan dengan KdMK yang belum dia ambil.



12. Membuat Tabel Mengajar

	Field Name	Data Type
🔑	KdDosen	Short Text
🔑	KdMK	Short Text
	Jam	Date/Time
	Ruang	Short Text

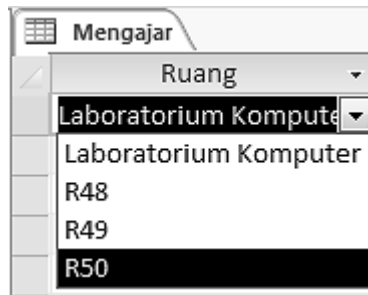
13. Atur KdDosen dengan *lookup* ke Tabel Dosen, agar hasilnya seperti di bawah ini:



14. Atur KdMK dengan *lookup* ke Tabel Mata Kuliah, agar hasilnya seperti di bawah ini:



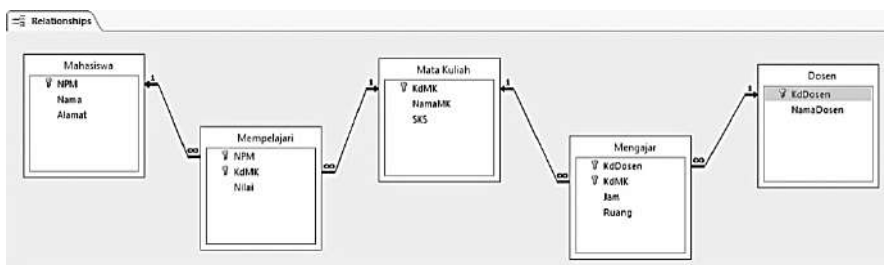
15. Atur Ruang dengan *lookup* manual isi *Row Source* dengan: Laboratorium Komputer, R48, R49, dan R50. Agar hasilnya seperti di bawah ini:



16. Mengisi Tabel Mengajar

KdDosen	KdMK	Jam	Ruang
Dos01	BTIK32	8:00	Laboratorium Komputer
Dos02	BTIK33	10:30	Laboratorium Komputer
Dos03	BTIK34	8:00	R48
Dos04	BKDK1	8:00	R49
Dos05	BMKU1	8:00	R50

17. Atur *Relationship* seperti gambar di bawah ini:



18. Buatlah *Query* dengan nama **Nilai BTIK32 >85**.

19. **Add** Tabel **Mempelajari**.

20. Isi kriteria pada *field* **KdMK** dan *field* **Nilai**.



Nilai BTIK32 >85

Mempelajari

*

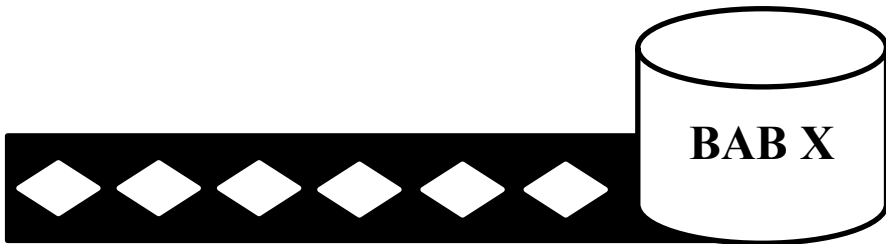
- 🔑 NPM
- 🔑 KdMK
- Nilai

Field:	NPM	KdMK	Nilai
Table:	Mempelajari	Mempelajari	Mempelajari
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Like "BTIK32"	>85
or:			

21. Tampilkan hasil *query*.

Nilai BTIK32 >85

NPM	KdMK	Nilai
3061546003	BTIK32	89



MERANCANG DAN MEMBUAT *FORM*



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu merancang dan membuat *form*
2. Mahasiswa mampu menutup dan menyimpan *form*
3. Mahasiswa mampu mengetahui bentuk *form*
4. Mahasiswa mampu menjalankan *form*
5. Mahasiswa mampu memodifikasi *form*
6. Mahasiswa mampu memasukkan data melalui *form*
7. Mahasiswa mampu memasukkan rumus ke dalam *form*



10.1 Form

Form adalah bentuk visual yang umumnya kita gunakan untuk menampilkan data dan memasukkan/*entry* data dari pemakai. *Form* juga digunakan sebagai tampilan program (visualisasi) atau juga sebagai kotak dialog untuk berinteraksi dengan pemakai (*interface/antarmuka*) dan sebagainya. Pembahasan tentang *form* berkaitan juga dengan *control*, yaitu objek yang kita tambahkan ke atas *form*, seperti **Text box**, **Combo Box**, **Button**, dan lain-lain sebagainya. Tanpa *control*, *form* hanya akan menjadi permukaan kosong. Pembahasan *form* dan *control* ini akan dipisahkan ke dalam bab masing-masing agar terorganisasi dan memudahkan pemahaman Anda.

Sebenarnya jika hanya ingin memasukkan dan mengelola data ke dalam *database*, kita tidak harus membuat *form*. Tampilan *datasheet* yang disediakan oleh Microsoft Office Access 2013 telah mencukupi untuk kebutuhan itu, baik untuk menampilkan isi tabel (secara langsung) maupun untuk menampilkan hasil *query*. *Form* berguna untuk misalnya jika kita ingin menampilkan data dari sebuah *record* yang memiliki banyak *field* secara lebih terstruktur, tidak lagi memanjang ke kanan seperti *datasheet*. Pemasukan data dan penyuntingan (*editing*) oleh pemakai juga bisa dibatasi per baris *record*, bisa diberi keterangan yang langsung tampak di layar, serta bisa menambahkan unsur-unsur yang memudahkan visualisasi data/informasi, maupun pemasukan data.

Form bisa dikaitkan dengan sumber data, bisa juga tidak memiliki data (hanya digunakan untuk tampilan visual), dan bisa juga mengelola data yang tidak dikaitkan secara langsung (dalam hal ini pengelolaan data menggunakan perintah pemrograman, baik menggunakan macro, maupun bahasa *Visual Basic Application/VBA*). Untuk menggunakan *form*, terlebih dahulu kita membuat *form* dengan **Design View** (atau bisa juga memanfaatkan *wizard* dan fasilitas lainnya) lalu menjalankannya (menampilkan dalam **Form View**).

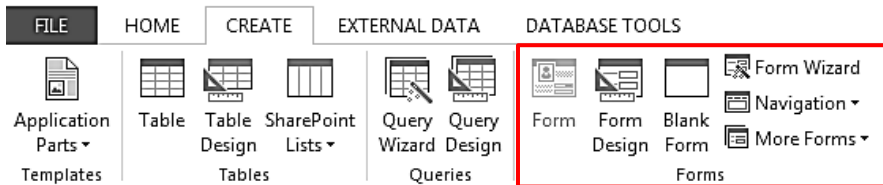
Catatan: Istilah menjalankan (*run*) *form* berarti membuka *form* dalam **Form View**. Istilah ini sama seperti membuka tabel yang berarti menampilkan data dalam **Datasheet View**, menjalankan *query* yang artinya



menampilkan hasil **SELECT** query dalam *datasheet* atau menjalankan *action query* (*action query* tidak mempunyai hasil untuk ditampilkan).

10.2 Merancang dan Membuat *Form*

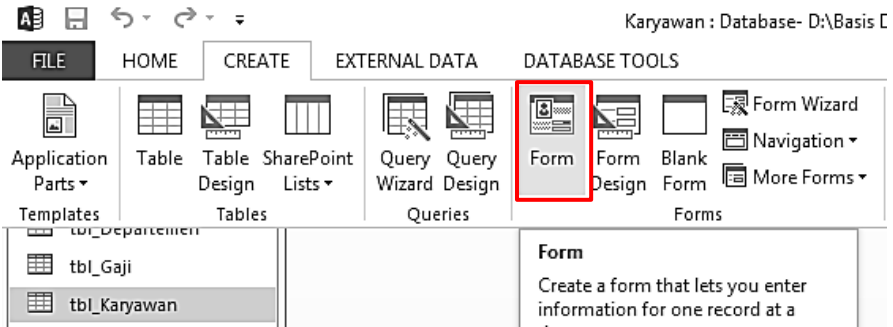
Seperti pada pembuatan *query*, untuk membuat *form*. Microsoft Office Access 2013 telah menyediakan **Form Design** (*Form Design View*). **Form Design** adalah alat utama untuk membuat *form*, atau memodifikasi *form* yang ada. Selain *form design*, Microsoft Office Access 2013 juga menyediakan beberapa cara lain untuk memudahkan atau mempercepat pembuatan *form*, seperti **Blank Form**, **Form Wizard**, **Navigasi**, dan lain-lain sebagainya. Untuk membuat *form* klik **tabel yang diinginkan** → klik **Ribbon Tab CREATE** → pada area **Forms** pilih jenis *form* yang diinginkan, seperti gambar di bawah ini.



Gambar 10.1 *Tools* untuk Membuat *Form* pada **Ribbon Tab CREATE**

10.3 Membuat *Form* dengan *Form*

Untuk membuat *form* klik **tabel yang diinginkan** misal **tbl_Karyawan** → klik **Ribbon Tab CREATE** → pada area **Forms** pilih **Form**.



Gambar 10.2 Ribbon Form

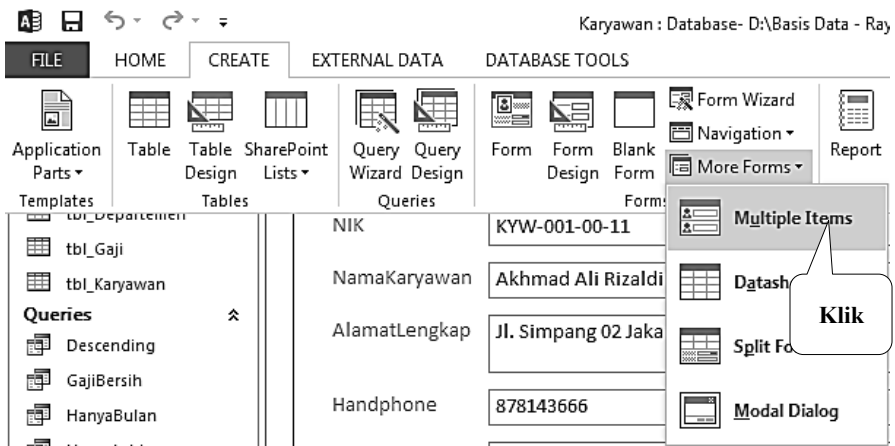
Dan hasilnya sebagai berikut:

NIK	KYW-001-00-11
NamaKaryawan	Akhmad Ali Rizaldi
AlamatLengkap	Jl. Simpang 02 Jakarta
Handphone	878143556
JenisKelamin	Laki-laki
Agama	Islam
TempatLahir	Surabaya
TanggalLahir	28-Feb-80
Pendidikan	S-1
TanggalGabung	29/01/2000
Golongan	IIIA
Skala	1
Departemen	Produksi

Gambar 10.3 Area Form View



Pembuatan *form* dengan cara di atas akan menghasilkan *single form*. Jika ingin membuat *form* berbentuk *datasheet*, *multiple items* (*continuous form*), *split form*, dan lain-lain. Dengan cara klik **More Form** (masih pada area **Ribbon Tab CREATE**) → lalu pilih bentuk *form* yang diinginkan.

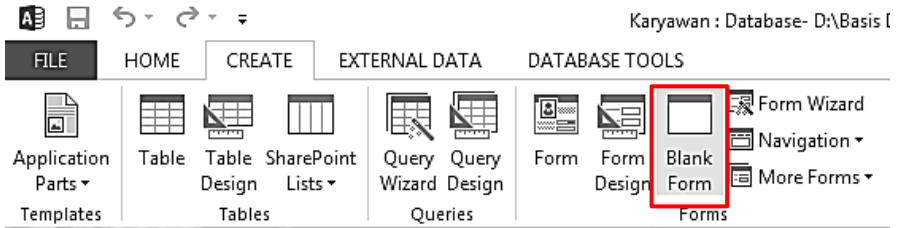


Gambar 10.4 Ribbon More Form

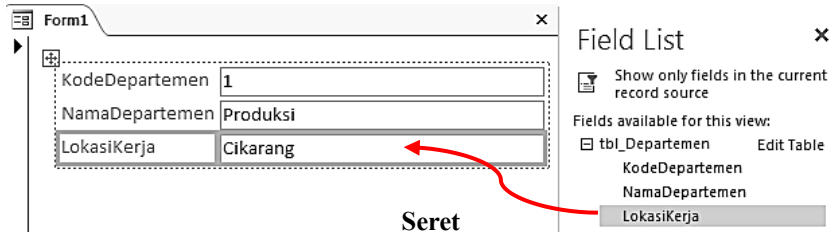
10.4 Membuat *Form* dengan *Blank Form*

Form kosong (*blank form*) adalah *form* yang tidak berisi apa-apa dan tidak mengandung data. Kita membuat *form* kosong jika ingin menambahkan sendiri *field* data, berbeda dengan pembuatan *form* dengan *Form* yang otomatis menambahkan *field* dari *table/query* yang dipilih pada *Navigation Pane*.

Untuk membuat *form* kosong, klik **Ribbon Tab CREATE** → klik **Blank Form**. Sebuah *form* kosong akan langsung ditampilkan dalam *Layout View*. Di sebelah kanan tampak daftar sumber data (*table/query*) dan *field* yang bisa kita tambahkan ke dalam *form*.



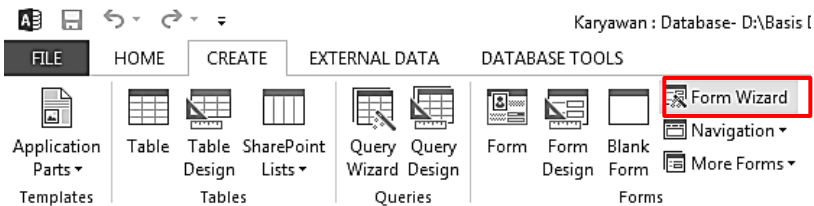
Gambar 10.5 Ribbon Blank Form



Gambar 10.6 Membuat Form dengan Blank Form

10.5 Membuat Form dengan Form Wizard

Form Wizard digunakan untuk membantu membuat form secara instan namun terdapat kesempatan bagi kita untuk menentukan bentuk form dan field yang ingin dimasukkan. Untuk membuat form dengan wizard. Dengan cara sebagai berikut: klik **Ribbon Tab CREATE** → klik **Form Wizard** → muncul **Form Wizard**.

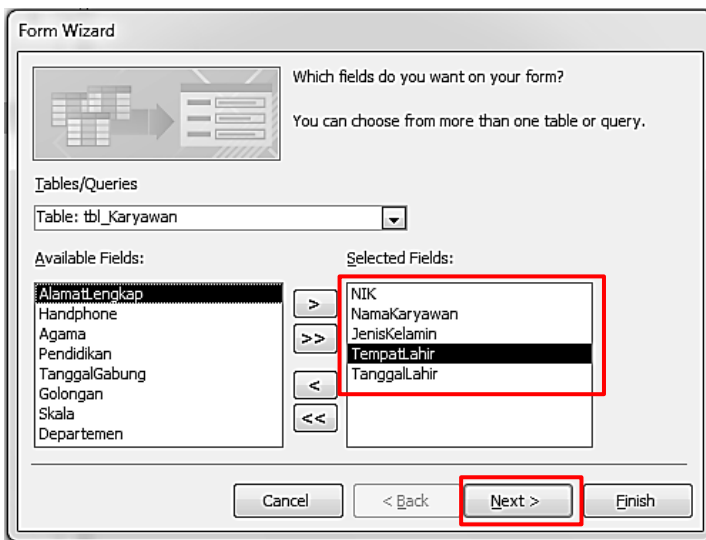


Gambar 10.7 Ribbon Form Wizard



Kemudian pilih sumber data (*table/query*). Jika ingin menggunakan data yang berasal dari gabungan *table* atau *query*, kita perlu membuat *query*-nya terlebih dahulu sebelum menjalankan *Form Wizard*.

Muncul kotak dialog **Form Wizard** pilih dan pindahkan *field* yang ingin ditampilkan di *form* → klik *Next*.

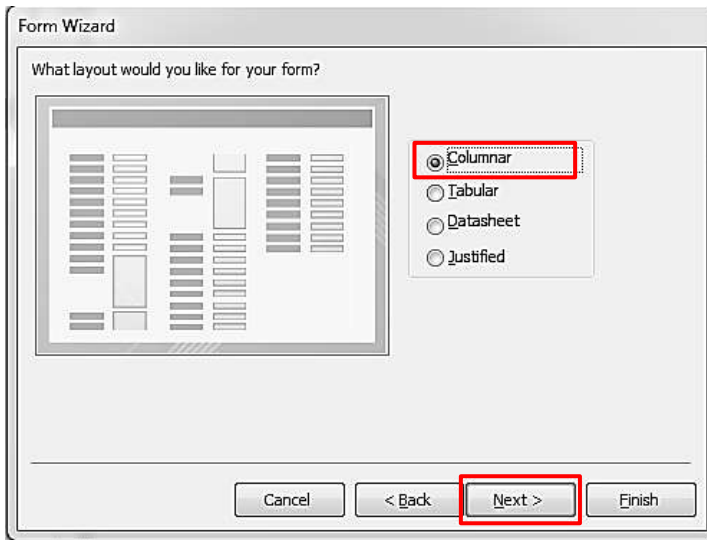


Gambar 10.8 Kotak Dialog *Form Wizard*

Perlu diperhatikan untuk fungsi tombol pada *Form Wizard*:

1. Fungsi “>” untuk memindahkan satu persatu *field* ke dalam area *Selected Fields*.
2. Fungsi “>>” untuk memindahkan semua *field* ke dalam area *Selected Fields*.
3. Fungsi “<” untuk mengembalikan satu persatu *field* ke dalam area *Available Fields*.
4. Fungsi “<<” untuk mengembalikan satu persatu *field* ke dalam area *Available Fields*.

Kemudian pilih **Columnar** → klik *Next*.



Gambar 10.9 Kotak Dialog *Form Wizard*

Perlu diperhatikan untuk fungsi tampilan pada *Form Wizard*:

1. **Columnar** yaitu berbentuk seperti formulir biasa disebut *single form*.
2. **Tabular** yaitu berbentuk seperti tabel, namun bukan *datasheet* biasa disebut *continuous form*.
3. **Datasheet** yaitu *form* yang ditampilkan dalam bentuk *datasheet*.
4. **Justified** yaitu berbentuk seperti formulir (*single form*) namun antara label dan *text box* diratakan.

Kemudian berikan nama *form* → klik pada bagian **Modify the form's design** → klik **Finish**.



Form Wizard

What title do you want for your form?
Form Daftar Karyawan Form Wizard

That's all the information the wizard needs to create your form.
Do you want to open the form or modify the form's design?

Open the form to view or enter information.
 Modify the form's design.

Cancel < Back Next > Finish

Gambar 10.10 Kotak dialog *Form Wizard*

Berikut adalah area *Design View* dan hasil tampilan dari *Form View*.

Form Daftar Karyawan Form Wizard

Form Header	
Form Daftar Karyawan Form Wizard	

Detail	
NIK	NIK
NamaKaryawan	NamaKaryawan
JenisKelamin	JenisKelamin
TempatLahir	TempatLahir
TanggalLahir	TanggalLahir

Form Footer

Gambar 10.11 Area *Design View Form*

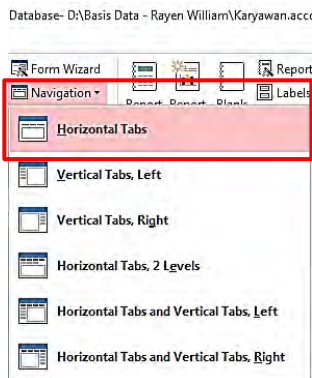


Form Daftar Karyawan Form Wizard	
NIK	KYW-001-00-11
NamaKaryawan	Akhmad Ali Rizaldi
JenisKelamin	Laki-laki
TempatLahir	Surabaya
TanggalLahir	28-Feb-80

Gambar 10.12 Contoh Bentuk *Columnar* Hasil dari *Form Wizard*

10.6 Membuat Navigation Form

Navigation Form digunakan untuk mengorganisasi *form* lain yang telah dibuat sehingga membentuk sebuah aplikasi. *Form* ini bisa dianggap pengganti *form Switchboard* pada Microsoft Office Access versi lama. Sebenarnya, *form* navigasi merupakan *form* induk yang mengandung *form* lain yang dimasukkan ke dalamnya sebagai *subform*. Oleh karena itu, *Navigation Form* sebaiknya dibuat setelah semua *form* yang ingin dimasukkan ke dalamnya tersedia. Berikut cara membuat *Navigation Form* sebagai berikut: klik **Ribbon Tab CREATE** → klik **Navigation**. Kemudian muncul pilihan bentuk *navigation form*.

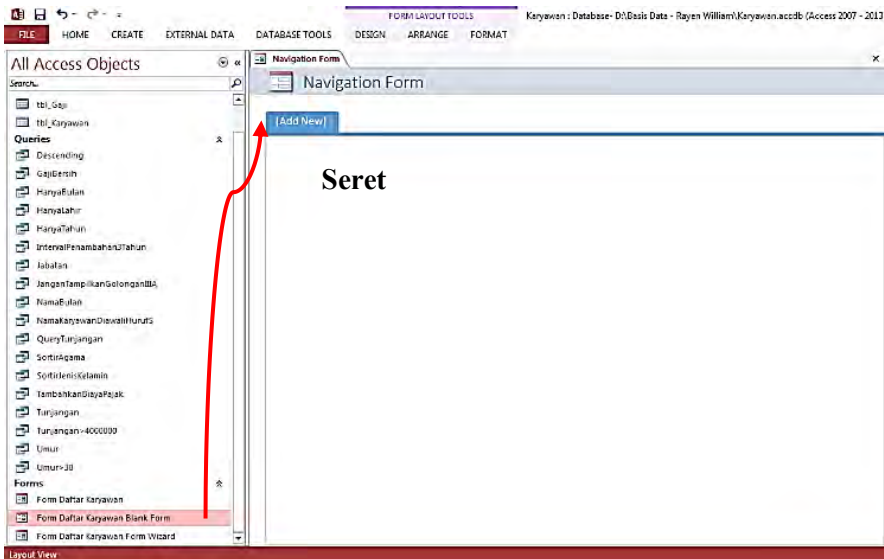


Gambar 10.13 Ribbon Navigation



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Misalnya kita memilih **Horizontal Tabs**. Muncul **Navigation Form** dalam bentuk **Layout View**. Kemudian **seret (drag) form** yang ingin ditambahkan dari **Navigation Pane** ke atas **Form Navigation**, **jatuhkan (drop)** pada baris **tab** di mana tertera **Add New**. Sebuah **tab** akan dibuat untuk menampung **form** yang ditambahkan.



Gambar 10.14 Area Layout Navigation Form

Setelah menambahkan **form**, kita sudah bisa menampilkan **navigation form** dalam **Form View**.



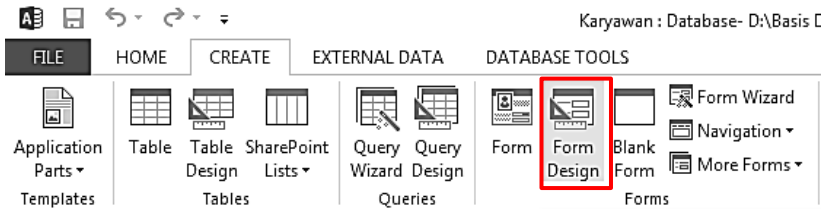
NIK	KYW-001-00-11
Nama Karyawan	Akhmad Ali Rizaldi
Alamat Lengkap	Jl. Simpan 02 Jakarta
Handphone	078143866
Jenis Kelamin	Laki-laki
Agama	Islam
Tempat Lahir	Surabaya
Tanggal Lahir	28-Feb-80
Pendidikan	S-1
Tanggal Gabung	29/01/2000
Golongan	IIIA
Skala	1

Gambar 10.15 Area *Form* Daftar Karyawan

Perlu diketahui bahwa *navigation form* yang mengandung banyak *form* cukup berat untuk dijalankan karena setiap *form* yang menjadi *subform*nya ikut dimuat juga secara sekaligus. Jika Anda ingin membuat *menu* atau *switchboard*, disarankan agar membuat sebuah *form* yang mengandung tombol-tombol menu saja untuk menjalankan setiap *form* aplikasi. Selain itu referensi kepada objek di dalam *form* berubah jika *form* tersebut dijadikan *subform*. *Form* navigasi ini lebih berguna dalam pembuatan *Web App*.

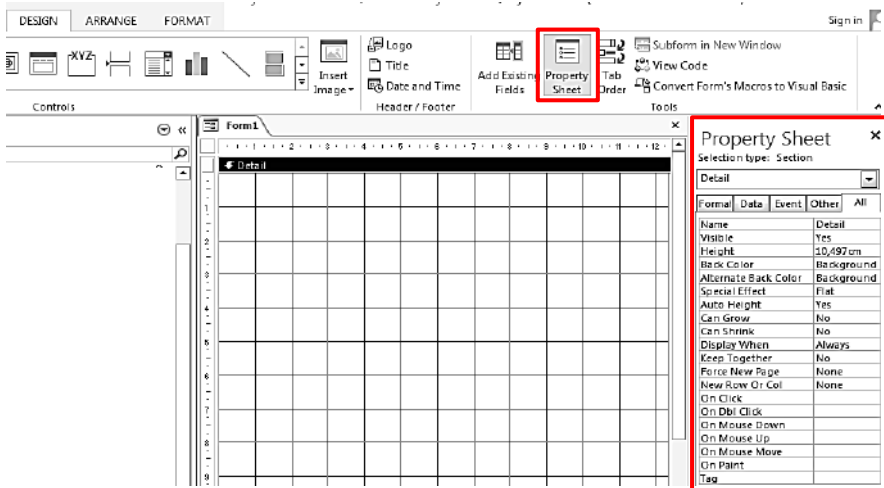
10.7 Membuat *Form* dengan *Form Design*

Form Design adalah alat utama untuk membuat *form*. Dengan *Form Design* kita bisa leluasa membuat *form*, menambahkan data, dan mengatur tata letak control pada *form*. Kita juga bisa menambahkan kemampuan *form* dan *control* secara pemrograman. Untuk membuat *form* dengan *Form Design* caranya sebagai berikut: klik **Ribbon Tab CREATE** → klik **Form Design**.



Gambar 10.16 Ribbon Form Design

Anda tidak perlu memilih *table/query* terlebih dahulu untuk dijadikan sumber data, karena pengaturan itu bisa dilakukan melalui kotak *Property Sheet*.

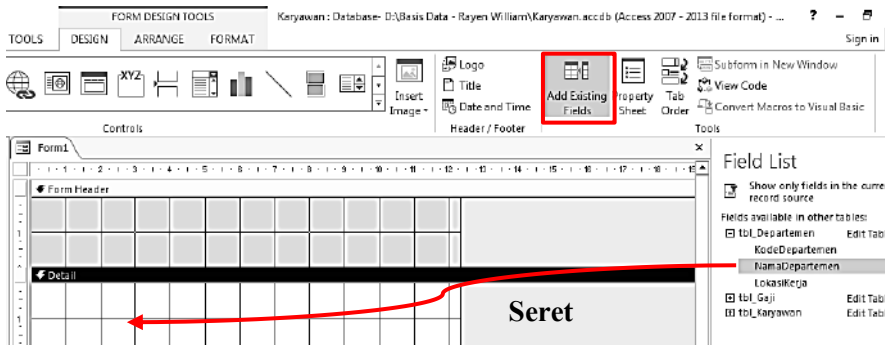


Gambar 10.17 Area *Property Sheet*

Untuk mengaitkan *form* dengan sumber data, tampilkan kotak *Property Sheet* (tekan F4) → klik **Tab Data** (pada *Property Sheet*). Pilih sumber data (*table* atau *query*) pada properti **Record Source**. Di sini kita bisa juga langsung memasukkan perintah SQL atau klik tombol [...] untuk membuat *query*.



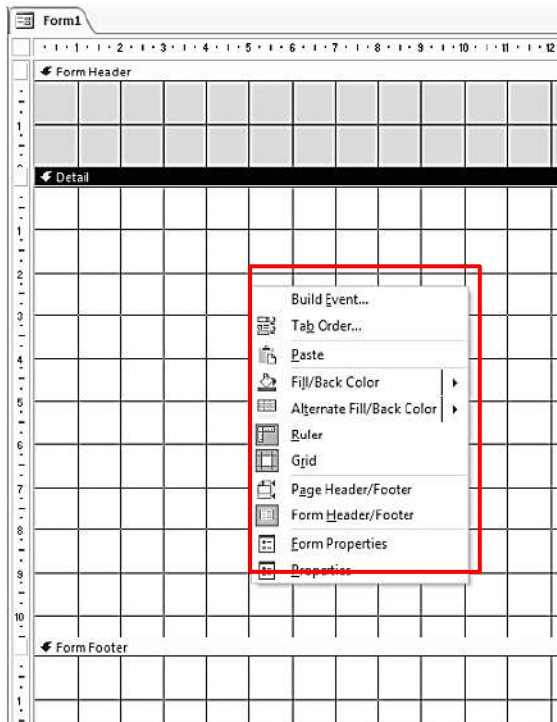
Tambahkan *control* yang diperlukan ke atas *form*, atau tampilkan daftar *field* dengan cara: klik **Ribbon Tab DESIGN** → klik **Add Existing Fields** → lalu seret *field* ke atas *form*.



Gambar 10.18 Area Add Existing Fields

Atur properti *form* lainnya agar sesuai dengan fungsi dan keindahan tampilan. Simpan dan beri nama *form*. Aturan pemberian nama *form* sama seperti pada *table* atau *query*, namun gunakan **frm_** sebagai *prefiks* (di awal nama).

Tampilan yang terlihat pada *Form Design* seperti **ruler**, **grid**, dan **section bar** bisa ditampilkan/tidak ditampilkan. Klik kanan pada bidang *form* dan pilih unsur yang mau ditampilkan/tidak ditampilkan.



Gambar 10.19 Menampilkan/Menyembunyikan Unsur pada Form

Perlu diketahui bahwa *ruler* dan *grid* hanyalah unsur bantuan dalam mendesain *form*, sedangkan *Form Header/Footer* adalah komponen yang akan menjadi bagian dari *form*. *Form Header/Footer* ini bersifat opsional (boleh dipakai atau tidak). Seksi yang tetap ada pada *form* adalah seksi **Detail**.

Form mempunyai unsur-unsur seperti *Record Selector*, *Navigation Buttons*, *Scroll Bars*, dan lain-lain. Kita bisa mengaktifkan atau tidak mengaktifkannya tergantung pada jenis *form* dan fungsinya. Untuk mengatur unsur-unsur *form* dilakukan melalui *Property Sheet*. Berikut adalah jenis *controls* yang ada di dalam *Ribbon Tab Design* pada area **FORM DESIGN TOOLS**.

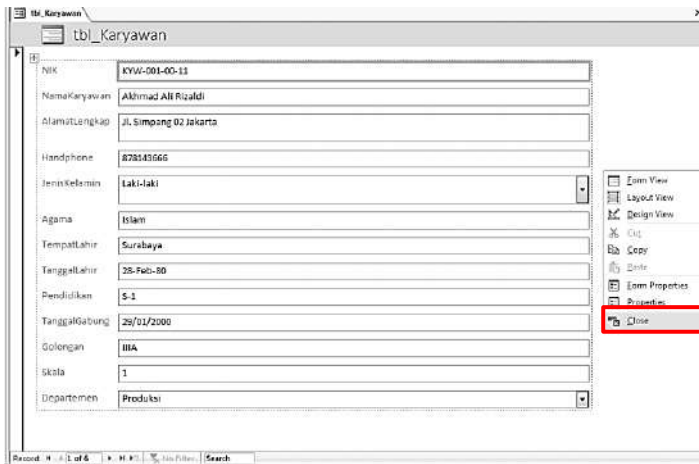


Gambar 10.20 Ribbon Form Design Tools

Form bisa mempunyai data jika dikaitkan dengan sumber data, namun bisa juga tidak mempunyai data hanya untuk tampilan saja. Data pada form bisa juga dikelola melalui komponennya misalnya menggunakan subform atau control yang lain, atau dikelola secara pemrograman.

10.8 Menutup dan Menyimpan Form

Untuk menutup form yang sedang berjalan, klik tombol **X (Close)** di ujung kanan area form. Kita juga bisa menekan **Ctrl+F4** untuk menutup form, atau mengklik kanan pada form dan pilih **close**.

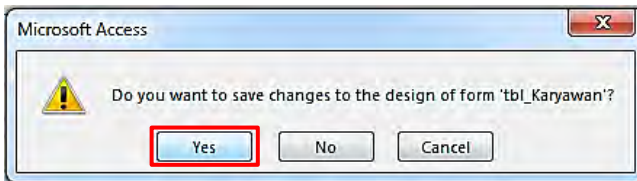


Gambar 10.13 Area Form View



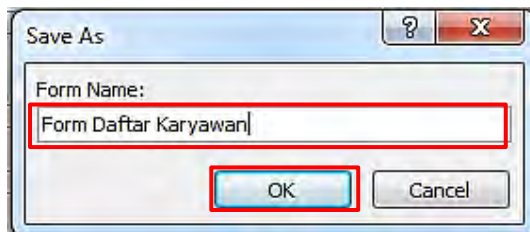
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Setelah menutup *form*, ketika sebuah *form* baru dibuat, maka akan muncul kotak dialog konfirmasi apakah yakin Anda menyimpan segala perubahan yang dilakukan dalam desain *form*? → klik **Yes**.



Gambar 10.14 Kotak Dialog Penyimpanan Perubahan *Form*

Kemudian muncul kotak dialog **Save As** ketikkan **nama form** pada *form name* → klik **OK**.



Gambar 10.15 Kotak Dialog *Save As*

10.9 Bentuk *Form*

Sebuah *form* dapat ditampilkan dalam berbagai bentuk, misalnya **Single Form**, **Datasheet**, dan lain-lain. Desain *form* bisa tetap sama (satu desain) namun ketika dijalankan bisa ditampilkan dalam berbagai bentuk. Bentuk tampilan *default* sebuah *form* ditentukan melalui properti **Default View** (ketika *form* diklik ganda pada *Navigation Pane* maka *form* akan ditampilkan dalam bentuk yang diatur pada properti *Default Value*).

Berikut ini adalah macam-macam bentuk tampilan *form* (dari desain yang sama):



1. Single Form

NIK	KYW-001-00-11
NamaKaryawan	Akhmad Ali Rizaldi
AlamatLengkap	Jl. Simpang 02 Jakarta
Handphone	878143666
JenisKelamin	Laki-laki
Agama	Islam
TempatLahir	Surabaya
TanggalLahir	28-Feb-80
Pendidikan	S-1
TanggalGabung	29/01/2000
Golongan	III A
Skala	1
Departemen	Produksi

Gambar 10.16 Tampilan *Single Form*

2. Continuous Form

NIK	NamaKaryawan	AlamatLengkap	Handphone	JenisKelamin	Agama
KYW-001-00-11	Akhmad Ali Rizaldi	Jl. Simpang 02 Jakarta	878143666	Laki-laki	Islam
KYW-002-12-11	Susanfi Anggraeni	Jl. Kebon Jeruk 09 Jakarta	982919111	Perempuan	Islam
KYW-003-10-11	I Made Itha Nuantara	Jl. Kebon Sarih 1 Jakarta	992871722	Perempuan	Hindu
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jakarta	828919189	Laki-laki	Kristen
KYW-005-07-11	I Nyoman Windu Asmara	Jl. Didina 9 Jakarta	928282027	Laki-laki	Hindu
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subroto Jakarta	918181818	Laki-laki	Budha
*	50		0		

Gambar 10.17 Tampilan *Continuous Form*



3. Split Form

NIK	NameKaryawan	AlamatLeng	Handphr	JenisKelamin	Agama	TempatLahi	TanggalLahi	Per	TanggalGabung
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subroto Jakarta	918181818	Laki-laki	Budha	Korea	16-Agust-87	S-1	09/01/2008
KYW-001-00-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143665	Laki-laki	Islam	Surabaya	28-Feb-80	S-1	29/01/2000
KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeru	982919111	Perempuan	Islam	Bandung	19-Sep-90	S-1	01/01/2012
KYW-003-10-11	Made Idha Nuantar	Jl. Kebon Sirih	992871722	Perempuan	Hindu	Nusa Dua Bali	06-Jun-89	S-1	08/08/2010
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jal	828919189	Laki-laki	Kristen	Medan	09-Sep-88	S-1	08/08/2010
KYW-005-07-11	Nyoman Windu Asn	Jl. Didina 9 Jak	928282827	Laki-laki	Hindu	Nusa Dua Bali	09-Agust-89	S-1	09/01/2007
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subro	918181818	Laki-laki	Budha	Korea	16-Agust-87	S-1	09/01/2008

Gambar 10.18 Tampilan Split Form

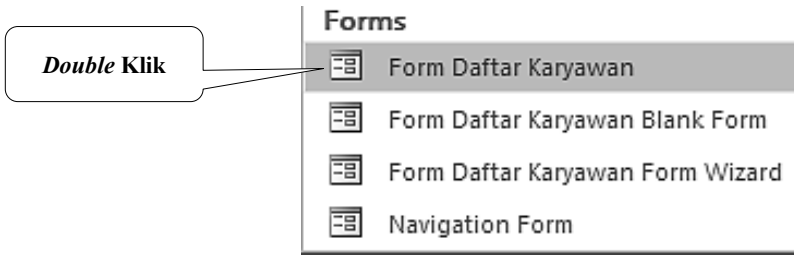
4. Datasheet

NIK	NameKaryawan	AlamatLeng	Handphr	JenisKelamin	Agama	TempatLahi	TanggalLahi	Per	TanggalGabung
KYW-001-00-11	Akhmad Ali Rizaldi	Jl. Simpang 02	878143665	Laki-laki	Islam	Surabaya	28-Feb-80	S-1	29/01/2000
KYW-002-12-11	Susanti Anggraeni	Jl. Kebon Jeru	982919111	Perempuan	Islam	Bandung	19-Sep-90	S-1	01/01/2012
KYW-003-10-11	Made Idha Nuantar	Jl. Kebon Sirih	992871722	Perempuan	Hindu	Nusa Dua Bali	06-Jun-89	S-1	08/08/2010
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jal	828919189	Laki-laki	Kristen	Medan	09-Sep-88	S-1	08/08/2010
KYW-005-07-11	Nyoman Windu Asn	Jl. Didina 9 Jak	928282827	Laki-laki	Hindu	Nusa Dua Bali	09-Agust-89	S-1	09/01/2007
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subro	918181818	Laki-laki	Budha	Korea	16-Agust-87	S-1	09/01/2008

Gambar 10.19 Tampilan Datasheet

10.10 Menjalankan Form

Menjalankan *form* artinya menampilkannya dalam **Form View** atau *view* lain yang ditentukan dalam properti **Default View**. Untuk menjalankan sebuah *form*, klik ganda *form* tersebut pada **Navigation Pane**.



Gambar 10.20 *Forms* pada *Navigation Pane*

10.11 Memasukkan/*Entry* Data Melalui *Form*

Untuk memasukkan data melalui *form* caranya: setelah diaktifkan *form* kemudian klik **Next Record** → isikan data.

The image shows a screenshot of the 'Form Daftar Karyawan' form view. The form has a title bar 'Form Daftar Karyawan' and a list of fields on the left: NIK, NamaKaryawan, AlamatLengkap, Handphone, JenisKelamin, Agama, TempatLahir, TanggalLahir, Pendidikan, TanggalGabung, Golongan, and Skala. The input fields for these fields are on the right. A red border highlights the input fields. The 'Handphone' field contains '0' and the 'Skala' field contains '0'. The 'JenisKelamin' field is a dropdown menu. The status bar at the bottom shows 'Record: 14 of 7', 'No Filter', and a search box.

Gambar 10.21 *Form View Form* Daftar Karyawan



Latihan Praktik

Soal 1:

Buka kembali *Database Akademik Nama Anda* pada praktikum sebelumnya dan buatlah *form* menggunakan *Split Form* sekreatif mungkin dari:

1. Tabel Mahasiswa berikan nama **Form Entri Mahasiswa**
2. Tabel Mata Kuliah berikan nama **Form Entri Mata Kuliah**
3. Tabel Dosen berikan nama **Form Entri Dosen**
4. Tabel Mempelajari berikan nama **Form Entri Mempelajari**
5. Tabel Mengajar berikan nama **Form Entri Mengajar**
6. *Query* Nilai BTIK32 >85 berikan nama **Hasil Nilai BTIK32 >85**

Soal 2:

1. Buatlah *Database* dengan nama **Perkuliahan**.
2. Buatlah Tabel dengan nama **Nilai**, dengan struktur data sebagai berikut:

	Field Name	Data Type
PK	NIM	Short Text
	NamaMahasiswa	Short Text
	MataKuliah	Short Text
	UTS	Number
	Tugas	Number
	UAS	Number

Ketentuan:

- *Field* NIM sebagai PK
- Untuk *field* NIM berikan *field size*, *Input Mask*, *Validation Rule*, dan *Validation Text* sebagai berikut:



General	Lookup
Field Size	7
Format	
Input Mask	LLL9999
Caption	
Default Value	
Validation Rule	Left([NIM];3)="RPL" Or Left([NIM];3)="PTI"
Validation Text	Awalan NIM berupa RPL dan PTI
Required	Yes
Allow Zero Length	Yes
Indexed	Yes (No Duplicates)
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

3. Buatlah *Form* dengan nama **Entri Nilai Mahasiswa** dengan menggunakan *Form Wizard*.
4. Lakukanlah modifikasi sebagai berikut:

The screenshot shows a Microsoft Access form titled "ENTRI NILAI MAHASISWA STKIP PGRI BANJARMASIN". The form is divided into two main sections: "Form Header" and "Detail".

Form Header: Contains the logo of STKIP PGRI Banjarmasin and the title "ENTRI NILAI MAHASISWA STKIP PGRI BANJARMASIN".


Detail Section: Contains the following fields and their validation rules:

Field Name	Validation Rule
Jurusan	=IIf(Left([NIM];3)="PTI";"
Nilai Angka	=0,3*[UTS]+0,3*[TUGAS]+
Nilai Huruf	=IIf([NILAI]>55;"Tidak Lul




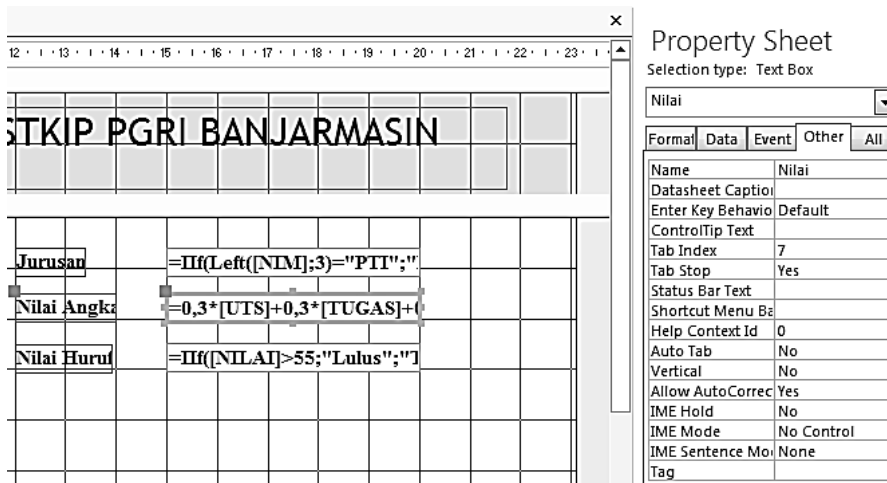
Ketentuan:

- Tambahkan *text box* untuk menambahkan informasi **Jurusan**.

Caranya pilih *text box*  → arahkan ke area *design view form* seperti gambar diatas → ketikkan rumus ini **=IIf(Left([NIM];3)="PTI";"Pendidikan Teknologi Informasi";"Rekayasa Perangkat Lunak")** pada kotak *Unbound*.


- Tambahkan *text box* untuk menambahkan informasi **Nilai**

Angka. Caranya pilih *text box*  → pilih *Properties* → pilih *Other* → ganti *Name* yang asalnya adalah **Text21** ubah **Nilai**.



Ketikkan rumus ini **=0,3*[UTS]+0,3*[TUGAS]+0,4*[UAS]** pada kotak *Unbound*.

- Tambahkan juga *text box* untuk menambahkan informasi **Nilai**

Huruf/Kelulusan. Caranya pilih *text box*  → arahkan ke area *design view form* seperti sebelumnya → ketikkan rumus ini



=Iif([NILAI]>55;"Lulus";"Tidak Lulus") pada kotak *Unbound*.

5. Masukkan data-data berikut melalui *Form* yang telah dibuat.

NIM	NamaMahasiswa	MataKuliah	UTS	Tugas	UAS
PTI0001	Muhammad Azriannor	Basis Data	90	85	90
PTI0002	Muhammad Zein Arifin	Jaringan Komputer	85	80	70
PTI0003	Rajo Langit Sidabutar	Pemrograman Internet	30	20	90
PTI0004	Immanuel Caesar Hitto	Multimedia	70	65	90
RPL0001	Nur Jasmine	Pemrograman Visual	90	80	80
RPL0002	Xim Lang Coung Wee	Logika Informatika	65	50	20
RPL0003	I Gede Agung Tirta Saylendra	Rekayasa Perangkat Lunak	85	75	75

6. Lihat hasilnya, misalnya salah satu *record*:

ENTRI NILAI MAHASISWA STKIP PGRI BANJARMASIN

NIM	<input type="text" value="PTI0004"/>	Jurusan	<input type="text" value="Pendidikan Teknologi Infor"/>
NamaMahasiswa	<input type="text" value="Immanuel Caesar Hitto"/>	Nilai Angka	<input type="text" value="76,5"/>
MataKuliah	<input type="text" value="Multimedia"/>	Nilai Huruf	<input type="text" value="Lulus"/>
UTS	<input type="text" value="70"/>		
Tugas	<input type="text" value="65"/>		
UAS	<input type="text" value="90"/>		

Jika Anda telah benar membuat *form* sesuai langkah-langkah di atas. Maka untuk data Jurusan, Nilai Angka, dan Nilai Huruf pada *form* akan terisi otomatis akibat dari pengisian rumus di masing-masing *Unbound*.



BAB XI

MERANCANG DAN MEMBUAT FORM LANJUTAN



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu mengetahui fungsi *control*
2. Mahasiswa mampu mengetahui jenis-jenis *control*
3. Mahasiswa mampu membuat *field* masukan data berbentuk *list box*
4. Mahasiswa mampu membuat *field* masukan data berbentuk *combo box*
5. Mahasiswa mampu membuat *field* masukan data berbentuk *option group*
6. Mahasiswa mampu membuat tombol perintah (*command button*)



11.1 Fungsi Control





Control adalah objek yang kita tambahkan ke atas *form* atau *report*. *Control* ini bisa berupa *Text Box* untuk menampilkan data atau untuk menerima *input* dari pemakai, bisa berupa *Combo Box*, *List Box*, *Buttons*, *Image*, *Subform*, dan sebagainya. *Field* yang kita tambahkan ke atas *form* dari **Field List** umumnya berupa *Text Box*, atau bisa juga berupa *Combo Box* jika kita membuat *Lookup*-nya, atau berupa *Check Box* jika tipe datanya *Yes/No* (tergantung pilihan pada properti **Display Control**).

Jika kita tidak mengatur properti *Display Control* suatu *field* pada desain tabel, kita tidak perlu mengubah desain tabel, namun bisa langsung mengubah jenis *control*-nya pada *form* atau *report* atau bisa juga menambahkan *control* yang ingin digunakan lalu mengaitkan *field* data pada *control* tersebut. Perlu diketahui bahwa *control* yang sama bisa kita tambahkan pada objek *form* atau *report*. Kita bahkan bisa meng-*copy* *control* yang ada pada *form* untuk di *paste* pada *report* dan sebaliknya.











11.2 Jenis-Jenis Control

Berikut ini adalah penjelasan tentang *control* yang tersedia pada Microsoft Office Access 2013:









Tabel 11.1 Penjelasan Masing-Masing Jenis Control

Tombol	Fungsi
 <i>Text Box</i>	Digunakan untuk menerima input dari pemakai atau untuk menampilkan data.
 <i>Label</i>	Digunakan untuk menambahkan teks.
 <i>Button</i>	Digunakan untuk menjalankan perintah tertentu.
 <i>Tab Control</i>	Digunakan untuk mengelompokkan <i>control</i> ke dalam beberapa tab. <i>Tab Control</i> bersifat <i>container</i> , yaitu menampung <i>control</i> yang lain yang dimasukkan ke dalamnya.



Tombol	Fungsi
 <i>Hyperlink</i>	Digunakan untuk menambahkan <i>hyperlink</i> yaitu teks yang umumnya jika diklik akan membuka halaman <i>web</i> .
 <i>Web Browser</i>	Digunakan untuk menampilkan halaman <i>web</i> (dokumen HTML).
 <i>Navigation</i>	Digunakan untuk membuat <i>form</i> navigasi.
 <i>Option Group</i>	Digunakan untuk menyediakan opsi atau pilihan. Jika pemakai memilih sebuah opsi, maka opsi yang lain menjadi tidak terpilih.
 <i>Page Break</i>	Digunakan untuk menandai pemisahan halaman. <i>Control</i> ini bisa digunakan pada <i>form</i> , namun terutama pada <i>report</i> .
 <i>Combo Box</i>	Digunakan untuk menampilkan pilihan data yang bisa langsung dipilih oleh pemakai ketika mengisi data.
 <i>Chart</i>	Digunakan untuk menambahkan <i>chart</i> .
 <i>Line</i>	Untuk menambahkan garis.
 <i>Toggle Button</i>	Digunakan untuk menambahkan tombol yang jika salah satu dipilih maka yang lain menjadi tidak terpilih.
 <i>List Box</i>	Digunakan untuk menampilkan data dalam bentuk daftar.



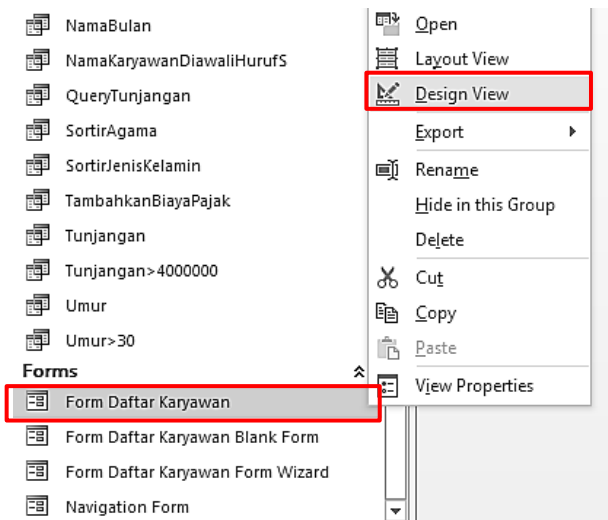
Tombol	Fungsi
 <i>Rectangle</i>	Digunakan untuk menambahkan kotak persegi.
 <i>Check Box</i>	Digunakan untuk menerima data <i>True</i> dan <i>False</i> .
 <i>Unbound Object Frame</i>	Digunakan untuk menerima objek yang berupa <i>file</i> eksternal (tidak disimpan dalam <i>database</i>).
 <i>Attachment</i>	Digunakan untuk menampilkan <i>file</i> atau dokumen yang di lampirkan (<i>attachment</i> yang disimpan dalam <i>field</i> bertipe <i>attachment</i>)
 <i>Option Button</i>	Digunakan untuk menjadi opsi di dalam <i>control Option Group</i> .
 <i>Subform/Subreport</i>	Digunakan untuk menambahkan <i>subform</i> atau <i>subreport</i> di dalam <i>form</i> atau <i>report</i> utama.
 <i>Bound Object Frame</i>	Digunakan untuk menampilkan objek yang disimpan dalam <i>database</i> .
 <i>Image</i>	Digunakan untuk menambahkan gambar

11.3 Membuat *Field* Entri Data Berbentuk *List Box*

Field entri data berbentuk *List Box* akan menampilkan sebuah kotak daftar pilihan untuk memilih isi suatu *field*. Untuk membuat sebuah *field* entri data berbentuk *List Box*, berikut caranya: buka **database** yang diinginkan → pilih **form** yang diinginkan → klik kanan dan pilih **Design View**.

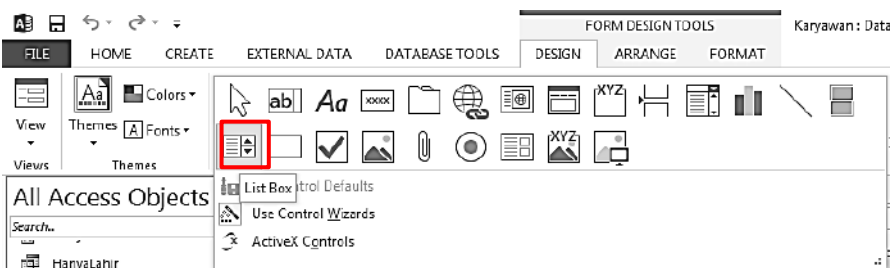


Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA



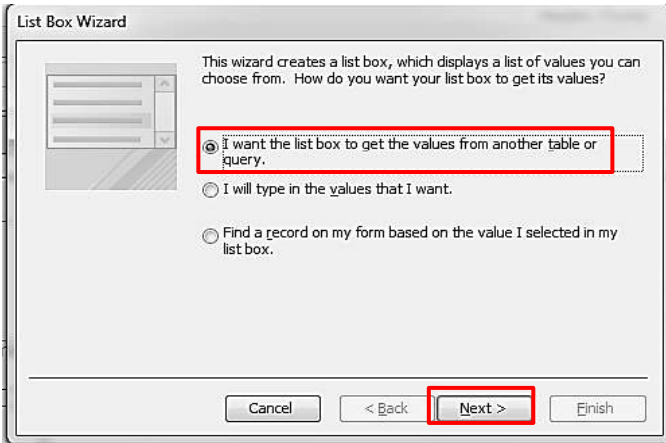
Gambar 11.1 Area Navigation Pane

Muncul area *Design View* dari *form* yang diinginkan → klik *Ribbon Tab DESIGN* → klik dan pilih *List Box* → arahkan ke area *Design View Form*.



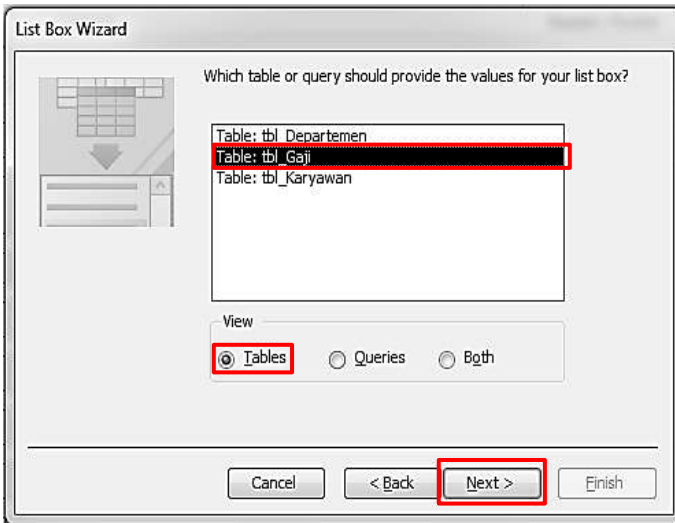
Gambar 11.2 Area Ribbon Tab DESIGN

Muncul kotak dialog *List Box Wizard* → klik *I want the list box....* (jika ingin nilai pilihannya dari *table/query*) → klik *Next*.



Gambar 11.3 Area List Box Wizard

Selanjutnya pilih dan klik *table/query* yang ingin dijadikan *List Box* → klik *Next*.

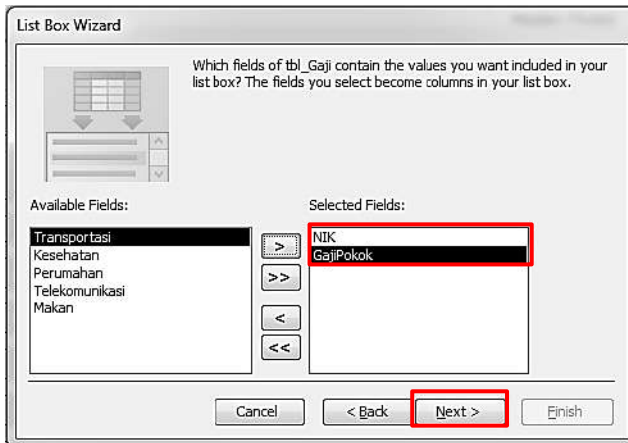


Gambar 11.4 Area List Box Wizard



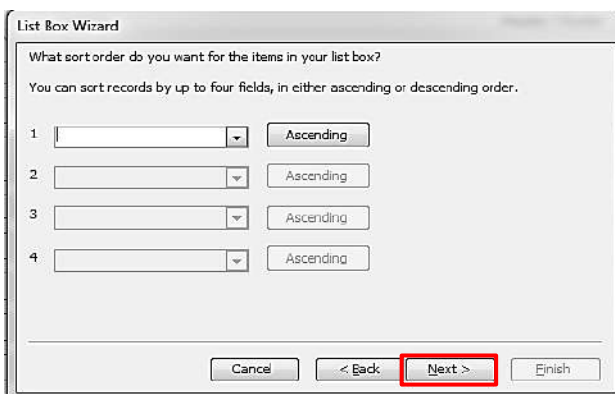
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Selanjutnya pindahkan *field* yang diinginkan untuk dijadikan *List Box* → klik *Next*.



Gambar 11.5 Area *List Box Wizard*

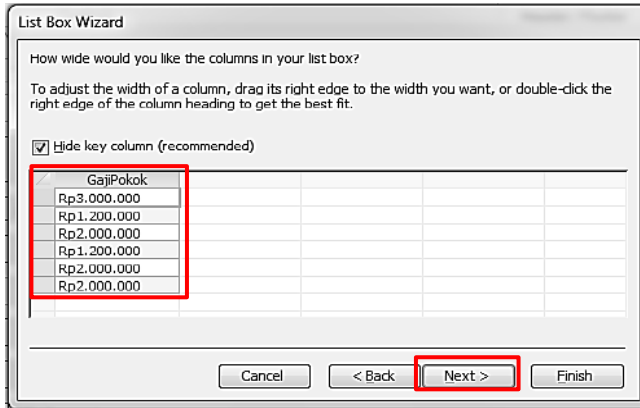
Selanjutnya apakah Anda ingin mengurutkan/tidak, kosongkan saja → klik *Next*.



Gambar 11.6 Area *List Box Wizard*

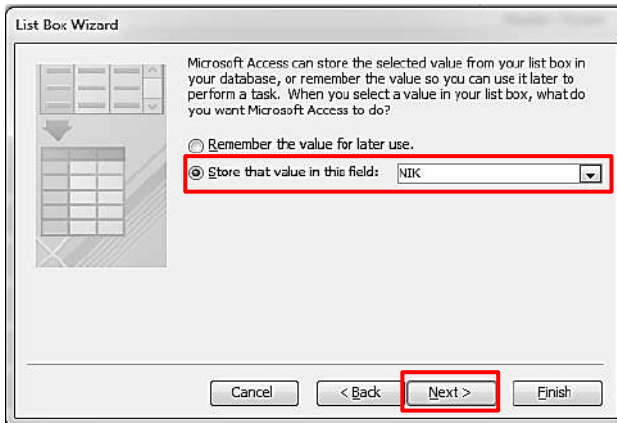


Selanjutnya hasil pengambilan data dari *field* dan *table* yang Anda pilih akan ditampilkan → klik *Next*.



Gambar 11.7 Area *List Box Wizard*

Beri tanda pada *Store that value in this field* → pilih dan klik nama *field* → klik *Next*.



Gambar 11.8 Area *List Box Wizard*



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Selanjutnya beri nama label pada *List Box* → klik **Finish**.



Gambar 11.9 Area *List Box Wizard*

Berikut hasil tampilan pembuatan *field* entri data berbentuk *List Box*.



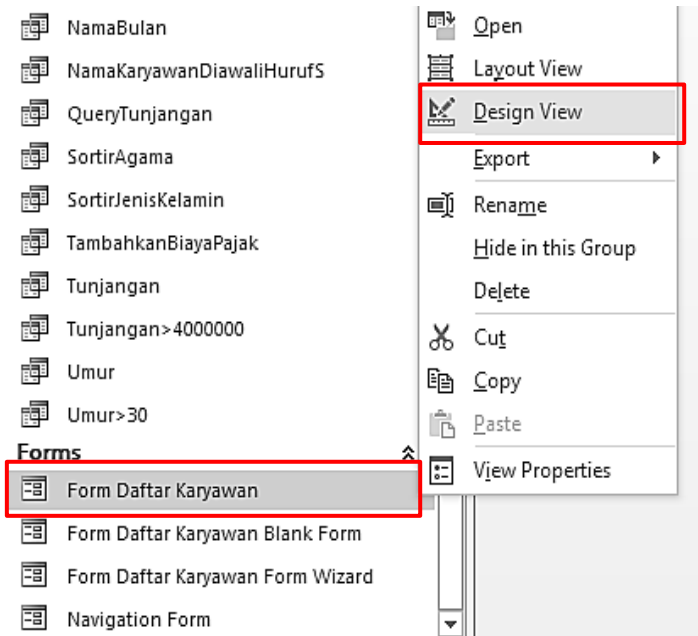
Form Daftar Karyawan	
Handphone	878143666
JenisKelamin	Laki-laki
Agama	
TempatLahir	Surabaya
TanggalLahir	28-Feb-80
Pendidikan	S-1
TanggalGabung	29/01/2000
Golongan	IIIA
Skala	1
Departemen	Produksi
Gaji Pokok	Rp3.000.000 Rp1.200.000 Rp2.000.000 Rp1.200.000 Rp3.000.000

Gambar 11.10 Area *Form View* Daftar Karyawan

11.4 Membuat *Field* Entri Data Berbentuk *Combo Box*

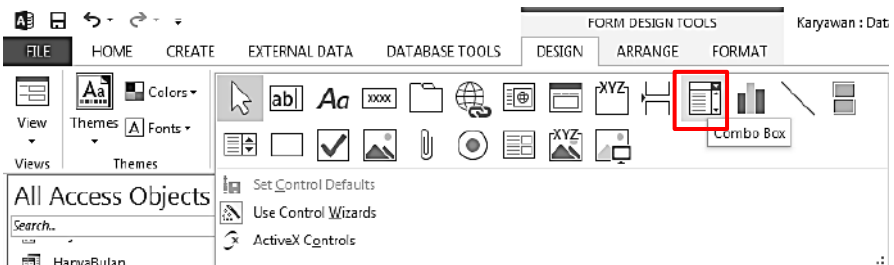
Field entri data berbentuk *combo box* akan menampilkan sebuah tombol atau daftar pilihan untuk memilih suatu *field*, tetapi Anda juga dapat mengetikkan data lainnya selain data yang tercantum pada tombol daftar pilihan tersebut. Untuk membuat sebuah *field* entri data berbentuk *combo box* sebagai berikut:

Buka *database* yang diinginkan → pilih *form* yang diinginkan → klik kanan dan pilih ***Design View***.



Gambar 11.11 Area Navigation Pane

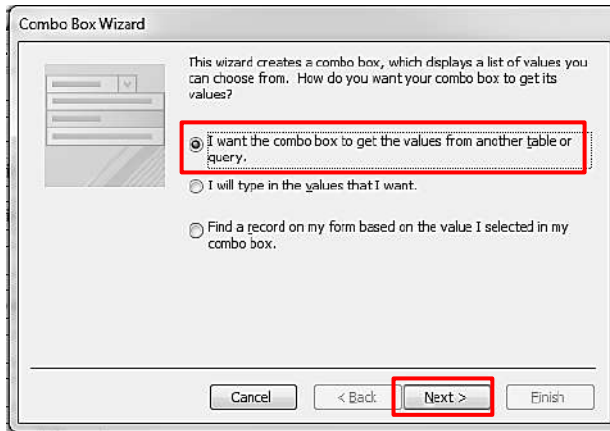
Muncul area *Design View* dari *form* yang diinginkan → klik *Ribbon Tab DESIGN* → klik dan pilih *List Box* → arahkan ke area *Design View Form*.



Gambar 11.12 Area Ribbon Tab DESIGN

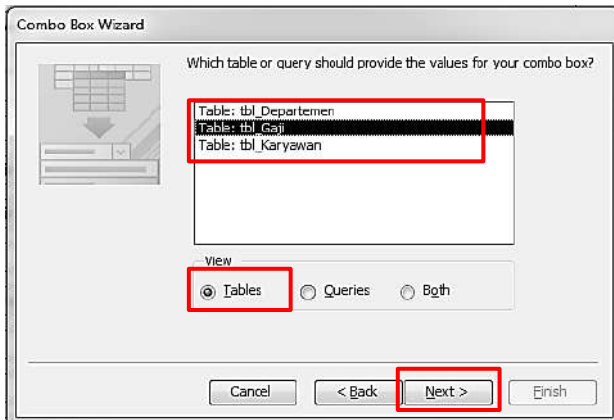


Muncul kotak dialog **Combo Box Wizard** → klik **I want the combo box....** (jika ingin nilai pilihannya dari *table/query*) → klik **Next**.



Gambar 11.12 Area Combo Box Wizard

Selanjutnya pilih dan klik *table/query* yang ingin dijadikan *Combo Box* → klik **Next**.

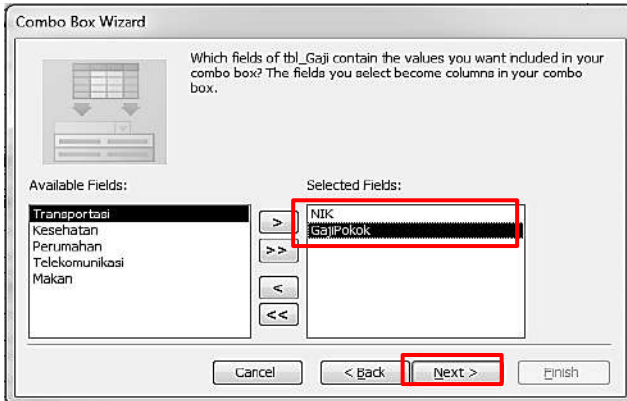


Gambar 11.13 Area Combo Box Wizard



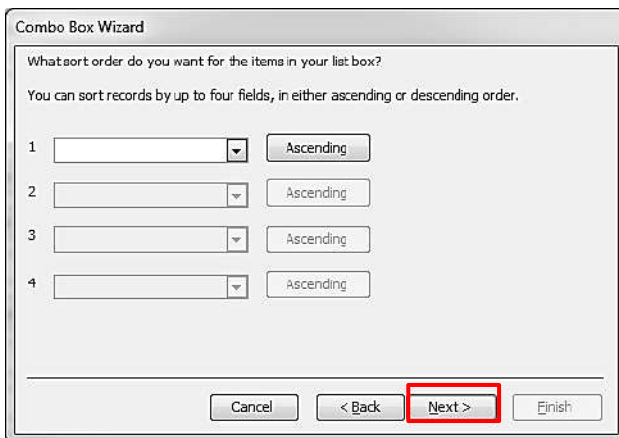
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Selanjutnya pindahkan *field* yang diinginkan untuk dijadikan *Combo Box* → klik *Next*.



Gambar 11.14 Area *Combo Box Wizard*

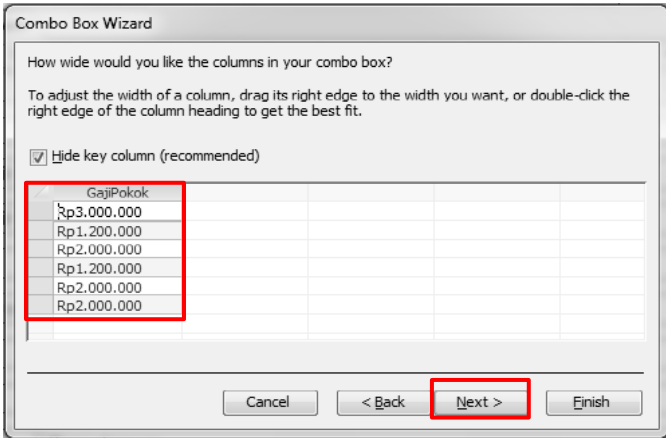
Selanjutnya apakah Anda ingin mengurutkan/tidak, kosongkan saja → klik *Next*.



Gambar 11.15 Area *Combo Box Wizard*

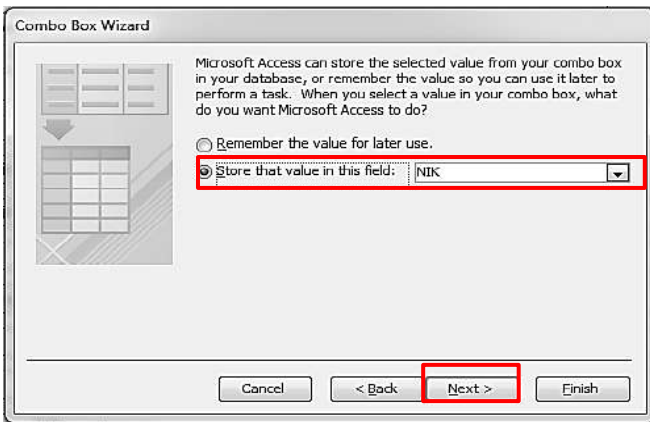


Selanjutnya hasil pengambilan data dari *field* dan tabel yang Anda pilih akan ditampilkan → klik *Next*.



Gambar 11.16 Area *Combo Box Wizard*

Beri tanda pada *Store that value in this field* → pilih dan klik nama *field* → klik *Next*.

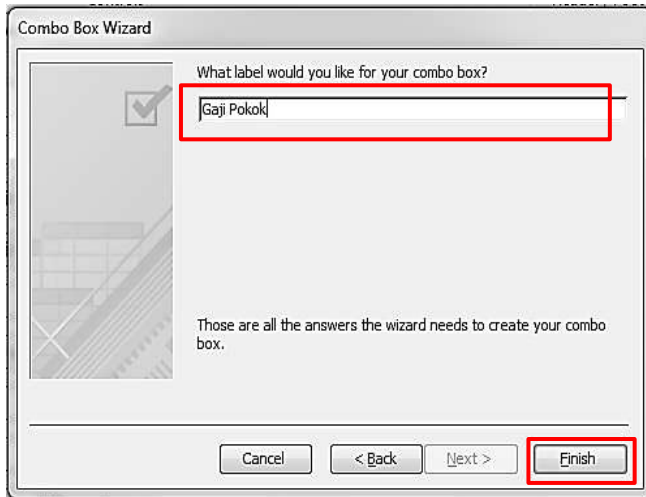


Gambar 11.17 Area *Combo Box Wizard*



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Selanjutnya beri nama label pada *Combo Box* → klik **Finish**.



Gambar 11.18 Area *Combo Box Wizard*

Berikut hasil tampilan pembuatan *field* entri data berbentuk *Combo Box*.



The screenshot shows a web-based form titled 'Form Daftar Karyawan'. The fields are as follows:

- Handphone: 878143666
- JenisKelamin: Laki-laki
- Agama: (empty)
- TempatLahir: Surabaya
- TanggalLahir: 28-Feb-80
- Pendidikan: S-1
- TanggalGabung: 29/01/2000
- Golongan: (dropdown menu open, showing options: Rp3.000.000, Rp1.200.000, **Rp2.000.000**, Rp1.200.000, Rp2.000.000)
- Skala: (empty)
- Departemen: (empty)
- Gaji Pokok: (empty)

Gambar 11.19 Area *Form View* Daftar Karyawan

11.5 Membuat *Field* Entri Data Berbentuk *Option Group*

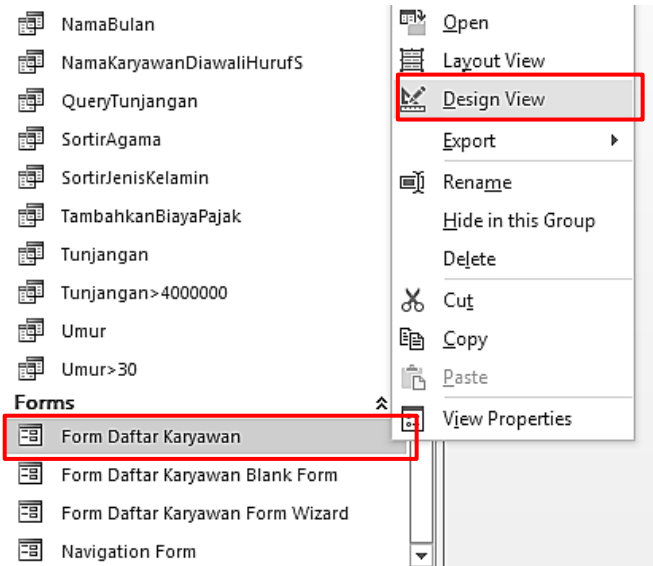
Field entri data berbentuk *option group* akan menampilkan sebuah kotak yang berisi data-data yang dapat Anda pilih (Anda hanya boleh memilih salah satu pilihan) dengan cara pemilihan menggunakan tombol kontrol berbentuk tombol pilihan, kotak cek, atau tombol *toggle*.

Control	Yes	No
Check Box	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Option Button	<input type="radio"/>	<input type="radio"/>
Toggle Button	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 11.20 Jenis Tombol *Control*

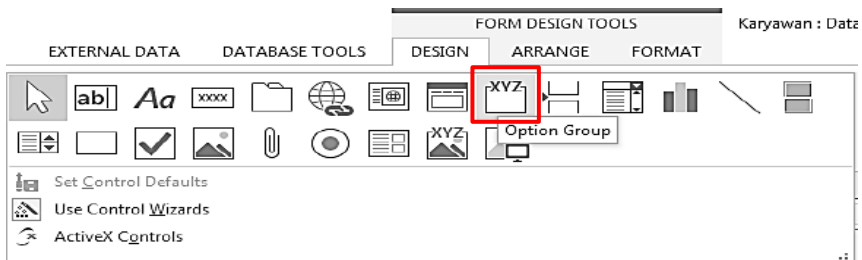


Caranya sebagai berikut: buka *database* yang diinginkan → pilih *form* yang diinginkan → klik kanan dan pilih **Design View**.



Gambar 11.21 Area Navigation Pane

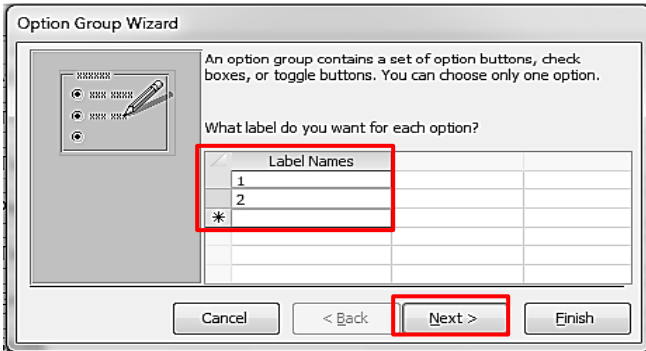
Muncul area **Design View** dari *form* yang diinginkan → klik **Ribbon Tab DESIGN** → klik dan pilih **Option Group** → arahkan ke area **Design View Form**.



Gambar 11.21 Area Ribbon Tab DESIGN

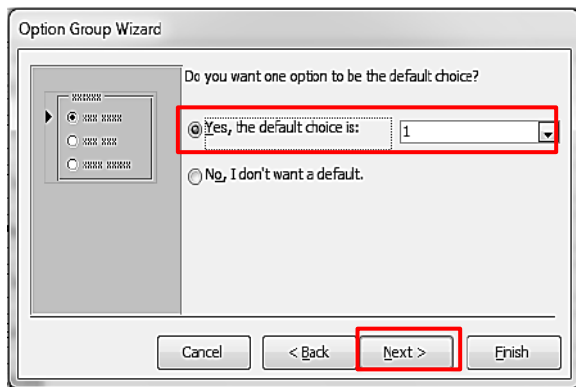


Muncul kotak dialog **Option Group Wizard** → ketikkan *label* atau keterangan yang ingin ditampilkan untuk masing-masing pilihan data pada **Label Names** → klik **Next**.



Gambar 11.22 Area **Option Group Wizard**

Apabila Anda menginginkan pada saat mengisi *record* baru secara *default* menampilkan pilihan data tertentu. Beri tanda atau klik tombol pilihan **Yes, the default choice is...**, pilih dan klik pilihan data *default* yang Anda inginkan → klik **Next**.

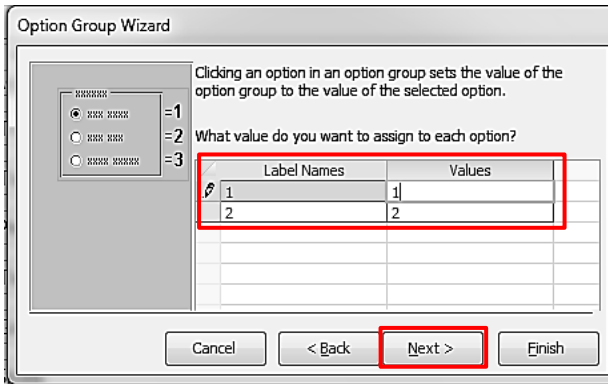


Gambar 11.23 Area **Option Group Wizard**



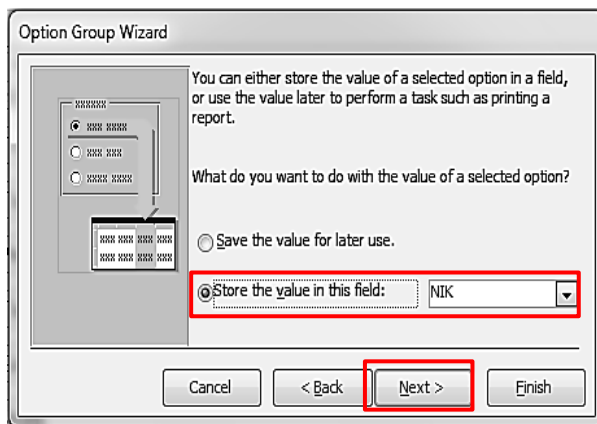
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Pada kotak *values* ketikkan data atau isi *field* yang akan dimasukkan sesuai dengan *label* yang dipilih → klik *Next*.



Gambar 11.24 Area Option Group Wizard

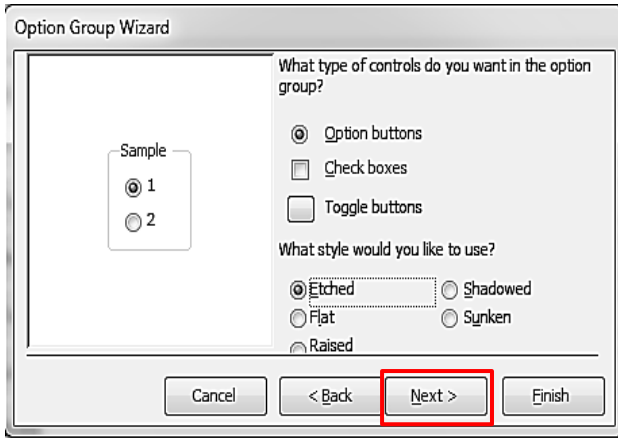
Beri tanda atau klik tombol pilihan *Store the value in this field*. Serta pilih dan klik nama *field* yang akan menerima data tersebut → klik *Next*.



Gambar 11.25 Area Option Group Wizard



Beri tanda atau klik bentuk tombol kontrol yang diinginkan → klik *Next*.



Gambar 11.26 Area Option Group Wizard

Isikan nama *label* untuk kotak *Option Group* yang diinginkan → klik *Finish*.



Gambar 11.27 Area Option Group Wizard



Berikut hasil tampilan pembuatan *field* entri data berbentuk **Option Group**.

The screenshot shows a web-based form titled "Form Daftar Karyawan". The form contains the following fields and values:

Handphone	982919111
JenisKelamin	Perempuan
Agama	Islam
TempatLahir	Bandung
TanggalLahir	19-Sep-90
Pendidikan	S-1
TanggalGabung	01/01/2012
Golongan	IIA
Skala	1
Departemen	Pemasaran

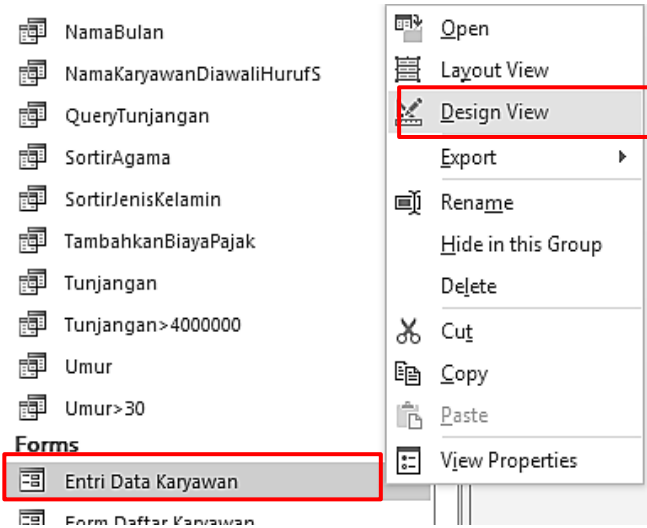
Below the main form, there is a section titled "Contoh Option Group" which displays three radio button options: 1 (selected), 2, and 3.

At the bottom of the form, there is a navigation bar with the text "Record: 1 of 6", "No Filter", and a "Search" button.

Gambar 11.28 Area *Form View* Daftar Karyawan

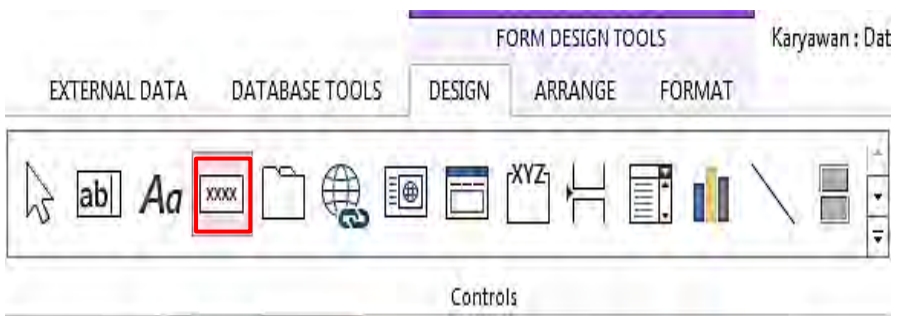
11.6 Membuat Tombol Perintah (*Command Button*)

Untuk memudahkan Anda bekerja dengan *form*, bila perlu Anda dapat membuat dan menempatkan tombol perintah berupa *Command Button* pada rancangan *form*. **Caranya** sebagai berikut: buka *database* yang diinginkan → pilih *form* yang diinginkan → klik kanan dan pilih **Design View**.



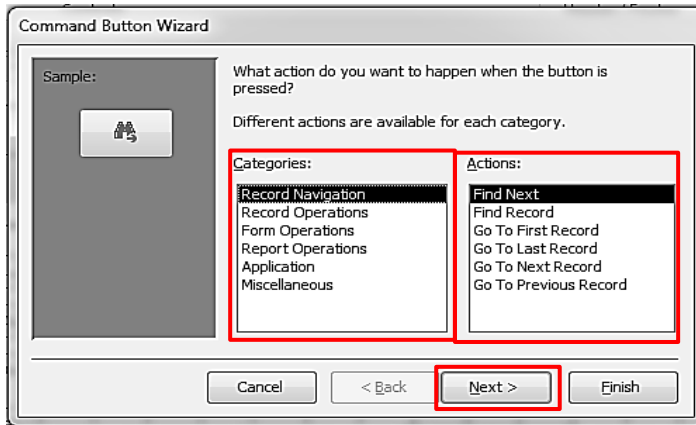
Gambar 11.29 Area Navigation Pane

Muncul area **Design View** dari **form** yang diinginkan → klik **Ribbon Tab DESIGN** → klik dan pilih **Button** → arahkan ke area **Design View Form**.



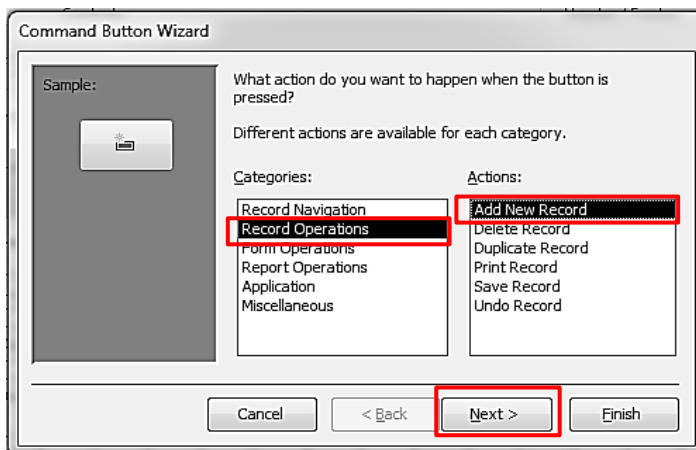
Gambar 11.30 Area Ribbon Tab DESIGN

Muncul kotak dialog **Command Button Wizard**.



Gambar 11.31 Kotak Dialog *Command Button Wizard*

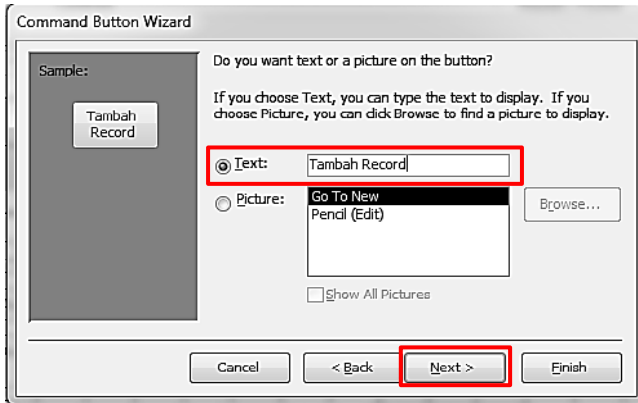
Pada kotak dialog lakukan pemilihan kategori dan aksi yang ingin dijalankan apabila tombol perintah dipilih, misalnya memilih **Record Operations** pada **Categories** → pada **Actions** pilih **Add New Record** (tombol yang berfungsi menambahkan *record* baru) → klik **Next**.



Gambar 11.32 Area Option Command Button Wizard



Pilih dan klik bentuk tampilan tombol perintah yang Anda inginkan. Anda dapat memilih bentuk tampilan teks atau gambar, misalnya yang dipilih *Text* ketikkan *Tambah Record* untuk nama tombol → klik *Next*.



Gambar 11.33 Area Option Command Button Wizard

Ketikkan nama untuk *name button* → klik *Finish*.



Gambar 11.34 Area Option Command Button Wizard



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Kemudian masuk ke dalam area *Design View*, editlah tampilan *Form* dan *Command Button* tadi.

ENTRI DATA KARYAWAN			
Detail			
NIK	NIK	TempatLahir	TempatLahir
NamaKaryawan	NamaKaryawan	TanggalLahir	TanggalLahir
AlamatLengkap	AlamatLengkap	Pendidikan	Pendidikan
Handphone	Handphone	TanggalGabung	TanggalGabung
JenisKelamin	JenisKelamin	Golongan	Golongan
Agama	Agama	Skala	Skala
		Departemen	Departemen
Tambah Record			
Form Footer			

Gambar 11.35 Area *Design View*

Berikut hasil tampilan dari pembuatan *Command Button* seperti gambar 11.36 berikut:



Entri Data Karyawan

ENTRI DATA KARYAWAN

NIK	KYW-3 - -	TempatLahir	Surabaya
NamaKaryawan		TanggalLahir	28-Feb-80
AlamatLengkap	Jl. Smpang 02 Jakarta	Pendidikan	S-1
Handphone	878143666	TanggalGabung	29/01/2000
JenisKelamin	Laki-laki	Golongan	IIIA
Agama		Skala	1
		Departemen	Produksi

Tambah Record

Record: 1 of 6 | No Filter | Search

Gambar 11.36 Area *Form View* Entri Data Karyawan

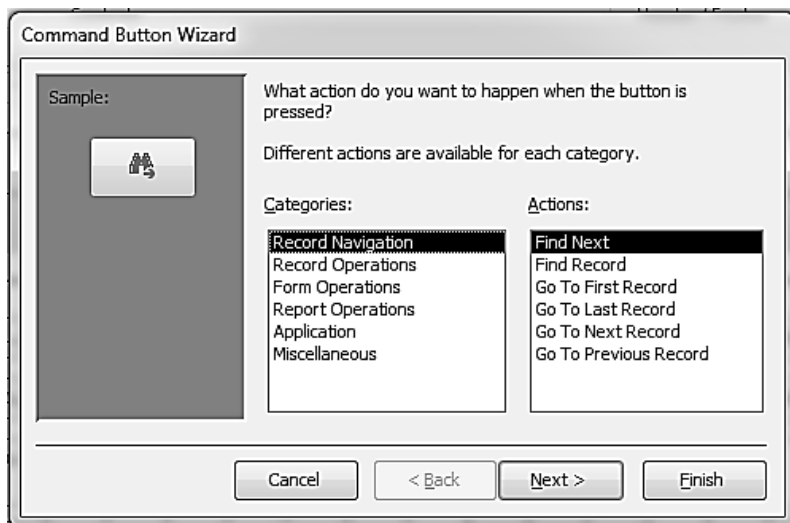
Latihan Praktik

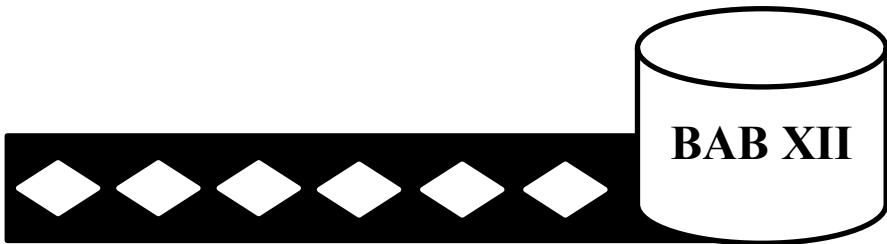
1. Buka kembali *database Akademik Nama Anda*, yang telah Anda buat pada praktikum sebelumnya!
2. Tambahkan tombol ***Find Record, Add New Record, Save Record, Close Form, Quit Application, Print Record, dan Delete Record*** pada *form* di bawah ini dan sesuaikan tempatnya sekreatif mungkin:
 - a. ***Form Entri Mahasiswa***
 - b. ***Form Entri Mata Kuliah***
 - c. ***Form Entri Dosen***
 - d. ***Form Entri Mempelajari***
 - e. ***Form Entri Mengajar***



Latihan Teori

Buatlah sebuah daftar tentang fungsi dari tombol-tombol yang disediakan oleh *Command Button Wizard*!





MERANCANG DAN MEMBUAT *REPORT*



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

Tujuan Instruksional Khusus :

1. Mahasiswa mampu membuat *report*
2. Mahasiswa mampu memodifikasi rancangan *report*
3. Mahasiswa mampu menyaring data pada *report*
4. Mahasiswa mampu mengurutkan data pada *report*
5. Mahasiswa mampu menghitung kelompok data pada *report*
6. Mahasiswa mampu membuat grafik pada *report*
7. Mahasiswa mampu menutup dan menyimpan *report*
8. Mahasiswa mampu mencetak *report*

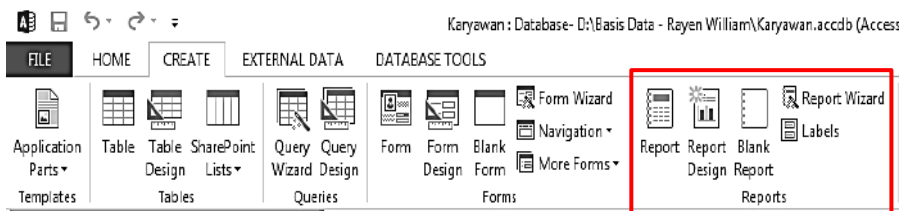


12.1 Report

Report adalah bentuk tampilan informasi yang menjadi keluaran program. Tujuan pembuatan *report* umumnya untuk melakukan pencetakan pada *printer*, namun bisa juga hanya ditampilkan di layar. *Design report* sebenarnya hanya berupa *template*. Kita menempatkan *field* apa di mana, lalu data akan mengalir sesuai data yang ada pada sumber data *report*.

12.2 Membuat Report

Seperti pada pembuatan *form*, untuk membuat *report*, Microsoft Office Access 2013 telah menyediakan beberapa alat (*tools*) yang terdapat pada *Ribbon Tab CREATE*.



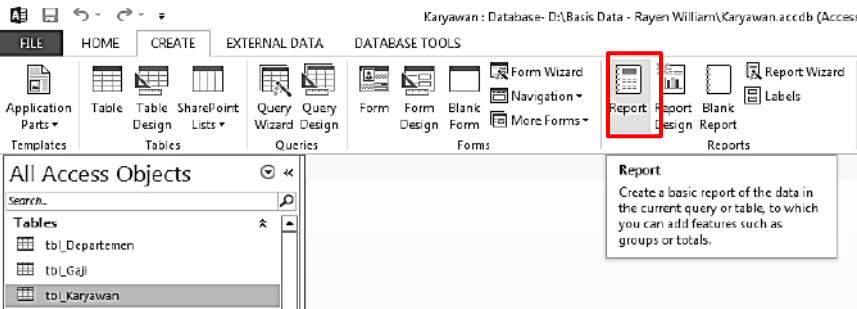
Gambar 12.1 Alat untuk Membuat Report pada Ribbon Tab CREATE

- **Report** digunakan untuk membuat *report* secara instan berdasarkan *table* atau *query* yang dipilih dari *Navigation Pane*.
- **Blank Report** digunakan untuk membuat *report* secara mudah, yaitu menambahkan *field* dari sumber data yang terdapat pada *Field List*.
- **Report Design** digunakan untuk membuat *report* secara manual.
- **Report Wizard** digunakan untuk membuat *report* dengan bantuan *wizard* (*step by step*).
- **Labels** digunakan untuk membuat *report* khusus yaitu untuk pembuatan *label* surat.

12.2.1 Membuat Report dengan Report



Berikut cara pembuatan *report* dengan *Report*: Klik *table* atau *query* yang ingin dijadikan sumber data pada *Navigation Pane* → Klik *Ribbon Tab CREATE* → klik *Report*.



Gambar 12.2 Membuat *Report* dengan *Report*

Setetika muncul sebuah *report* dalam tampilan *Layout View*, di mana kita bisa melakukan modifikasi seperti mengubah judul *report*, mengubah lebar kolom, dan sebagainya.

NIK	NamaKaryawan	AlamatLengkap	Handphone	JenisKelamin
KYW-002-08-77		Jl. Simpang 02 Jakarta	878143666	Laki-laki
KYW-001-09-10	Susanti Anggraeni	Jl. Kebon Jeruk 09 Jakarta	982919111	Perempuan
KYW-003-10-11	I Made Idha Nuantara	Jl. Kebon Sirih 1 Jakarta	992871722	Perempuan
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jakarta	828919189	Laki-laki
KYW-005-07-11	I Nyoman Windu Asmara	Jl. Didina 9 Jakarta	928282827	Laki-laki
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subroto Jakarta	918181818	Laki-laki

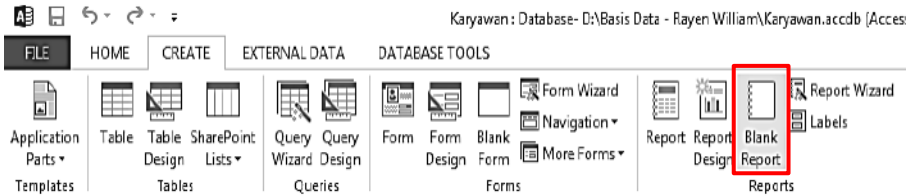
Gambar 12.3 *Layout View Report* Laporan Data Karyawan

12.2.2 Membuat Report dengan Blank Report



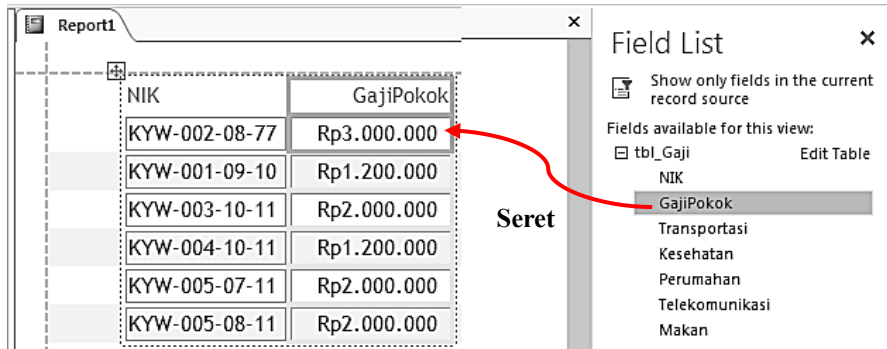
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Berikut cara membuat *report* dengan *Blank Report*: **tanpa memilih table** atau *query* terlebih dahulu → klik **Ribbon Tab CREATE** → klik **Blank Report**.



Gambar 12.4 Membuat *Blank Report*

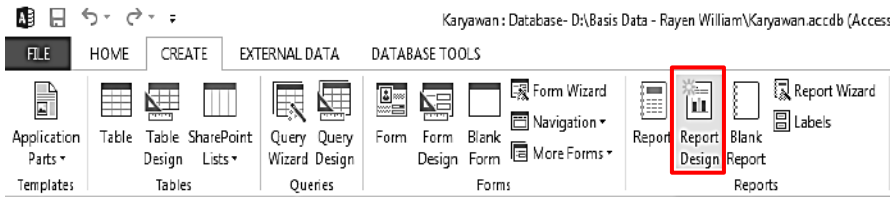
Klik **Show all tables** pada *Field List* → lalu buka (*expand*) *table* atau *query* yang ingin dijadikan sumber data → klik dan seret *field* yang ingin ditambahkan ke atas *report*.



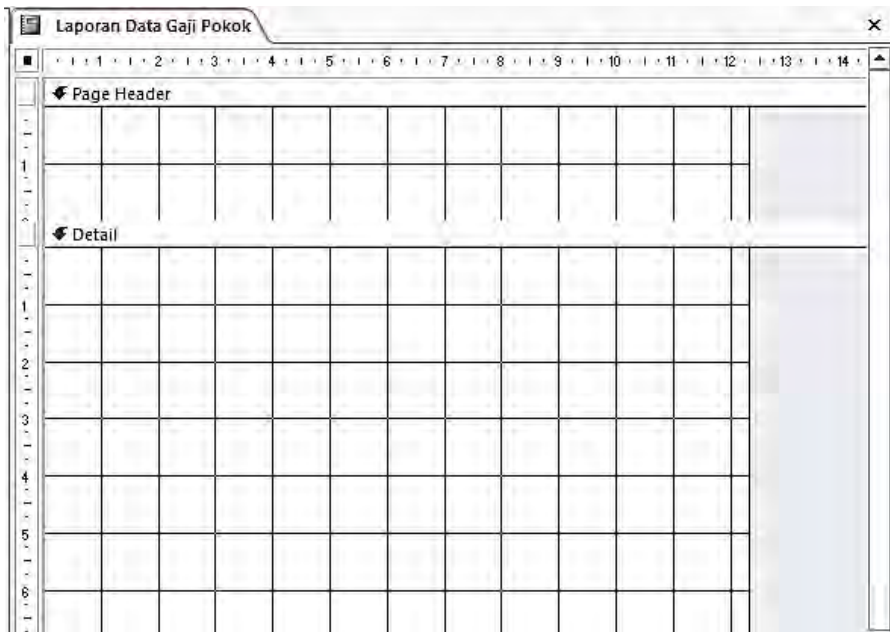
Gambar 12.5 Menambahkan *Field* ke Atas *Report*

12.2.3 Membuat Report dengan Report Design

Untuk membuat *report* dengan *Report Design* **juga tidak perlu memilih table** atau *query*, namun bisa langsung dengan cara: klik **Ribbon Tab CREATE** → klik **Report Design**.

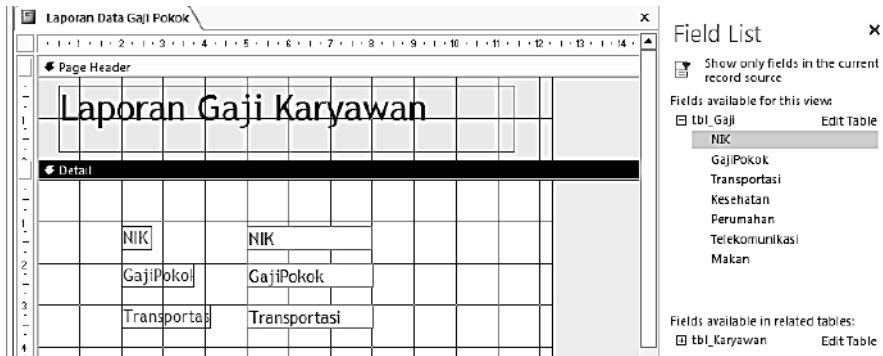


Gambar 12.6 Membuat *Report Design*



Gambar 12.7 Tampilan *Report Kosong* dalam *Design View*

Seret *field* yang ada pada *Field List* ke dalam area *Design View Report*. Sesuai keinginan, Anda dapat menentukan sendiri tata letak dari *field* dan *control* lainnya ke dalam *report*.



Gambar 12.8 Area Design View Report

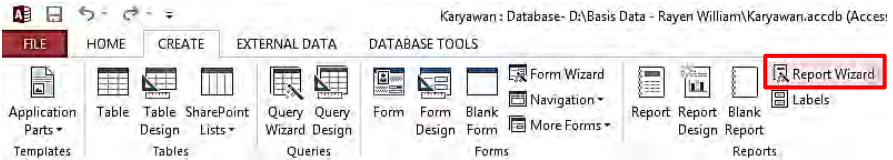
Berikut hasilnya:



Gambar 12.9 Report Laporan Gaji Karyawan

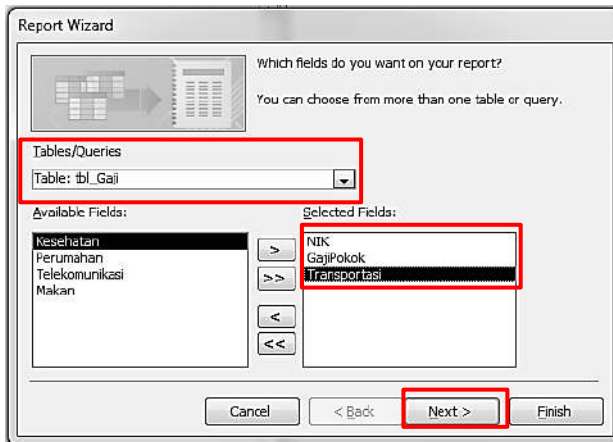
12.2.4 Membuat Report dengan Report Wizard

Berikut cara membuat *report* dengan *Report Wizard*: klik **Ribbon Tab CREATE** → klik **Report Wizard**.



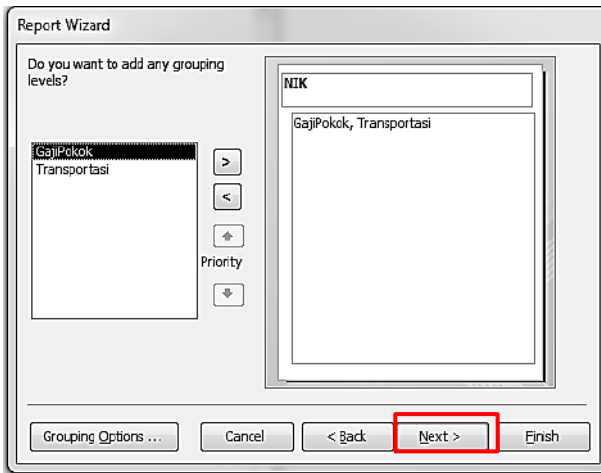
Gambar 12.10 Membuat Report Wizard

Muncul kotak dialog *Report Wizard* → pilih *table/query* sebagai sumber data → pilih *field* yang akan ditampilkan dalam *report* → klik *Next*.



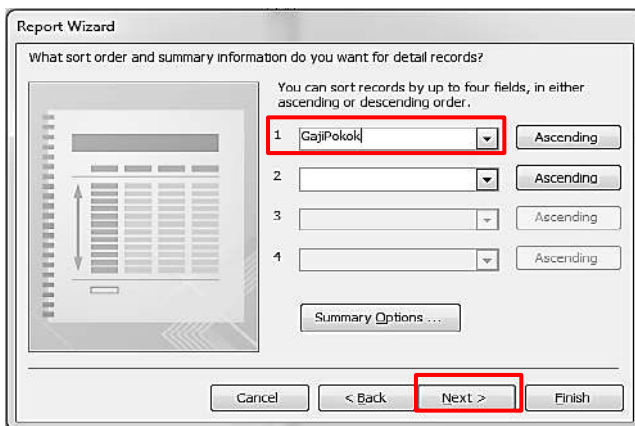
Gambar 12.11 Kotak Dialog Report Wizard

Tentukan *field* sebagai kunci pengelompokan data. Kita bisa mengelompokkan data dalam beberapa tingkat, namun bisa juga tidak → klik *Next*.



Gambar 12.12 Kotak Dialog Report Wizard

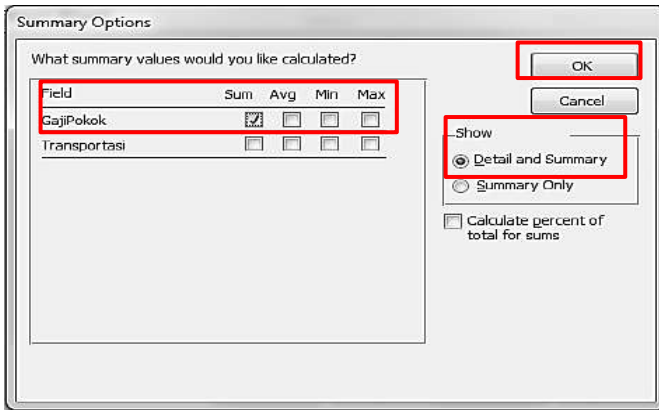
Tentukan *field* sebagai kunci pengurutan. Kita bisa mengurutkan data berdasarkan beberapa kunci pengurutan, namun bisa juga tidak (jika tidak menentukan kunci pengurutan, data akan ditampilkan berdasarkan urutan kunci *primernya*, kalau ada). → klik *Next*.



Gambar 12.13 Kotak Dialog Report Wizard

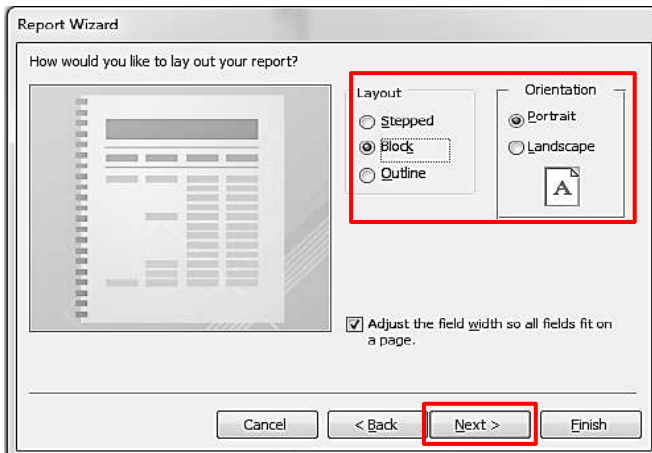


Klik tombol **Summary Options** untuk menambahkan baris total (jika ada kolom yang ingin di total), misal pilih dan centang **Sum** pada **field GajiPokok** → klik **OK**.



Gambar 12.14 Kotak Dialog *Summary Options*

Tentukan bentuk *layout* laporan dan orientasi kertas → klik **Next**.

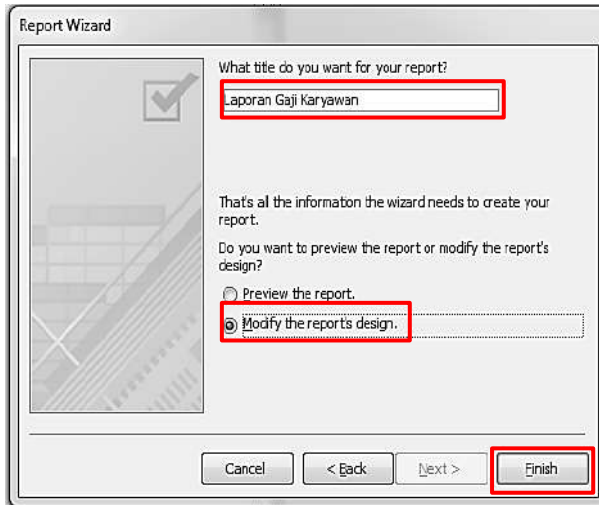


Gambar 12.15 Kotak Dialog *Report Wizard*



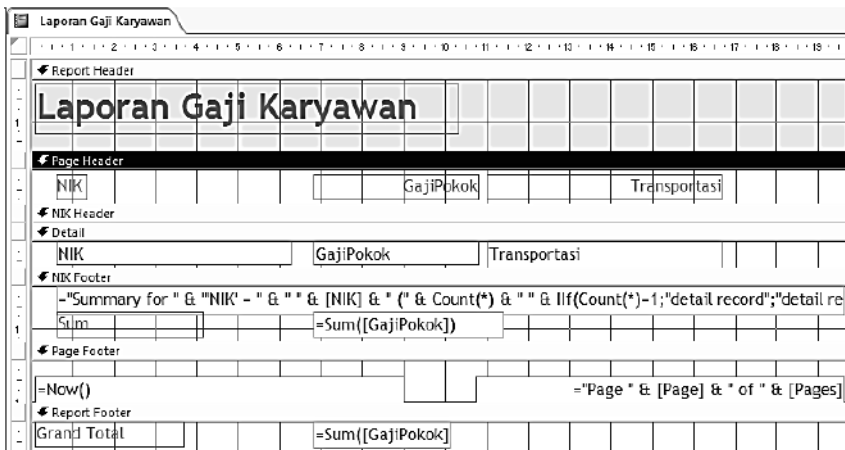
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Beri judul laporan → centang *Modify the report's design* → klik **Finish**.



Gambar 12.16 Kotak Dialog *Report Wizard*

Tampil area *Design View* dari *Report Wizard*.



Gambar 12.17 Area *Design View Report*



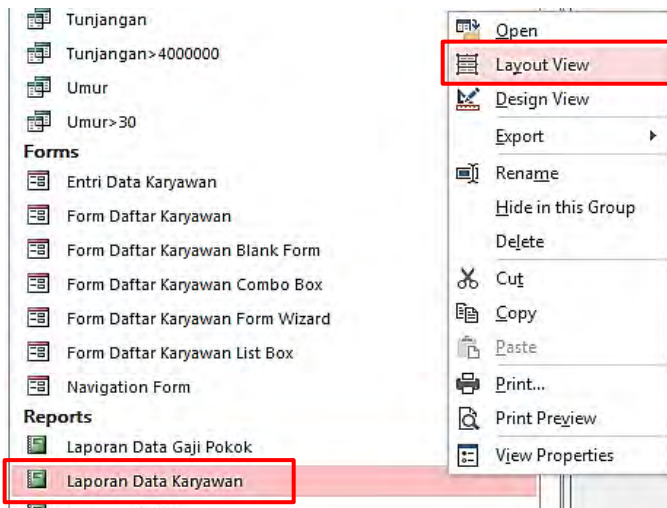
Hasil dari *Report Wizard* yang telah dimodifikasi.

Laporan Gaji Karyawan		
NIK	GajiPokok	Transportasi
KYW-001-09-10	Rp1.200.000	Rp500.000
Summary for 'NIK' = 0010910 (1 detail record)		
Sum	Rp1.200.000	
KYW-002-08-77	Rp3.000.000	Rp500.000
Summary for 'NIK' = 0020877 (1 detail record)		
Sum	Rp3.000.000	
KYW-003-10-11	Rp2.000.000	Rp500.000
Summary for 'NIK' = 0031011 (1 detail record)		
Sum	Rp2.000.000	
KYW-004-10-11	Rp1.200.000	Rp500.000
Summary for 'NIK' = 0041011 (1 detail record)		
Sum	Rp1.200.000	
KYW-005-07-11	Rp2.000.000	Rp500.000
Summary for 'NIK' = 0050711 (1 detail record)		
Sum	Rp2.000.000	
KYW-005-08-11	Rp2.000.000	Rp500.000
Summary for 'NIK' = 0050811 (1 detail record)		
Sum	Rp2.000.000	
Grand Total	Rp11.400.000	

Gambar 12.18 Hasil *Report*

12.3 Menyaring Data pada *Report*

Menyaring data dalam artian memilah/menyeleksi data yang mana yang ingin ditampilkan pada *report*. Untuk menyaring data *report* dengan cara: pilih dan klik *report* yang diinginkan → klik kanan → klik *Layout View*.



Gambar 12.19 Area Navigation Pane

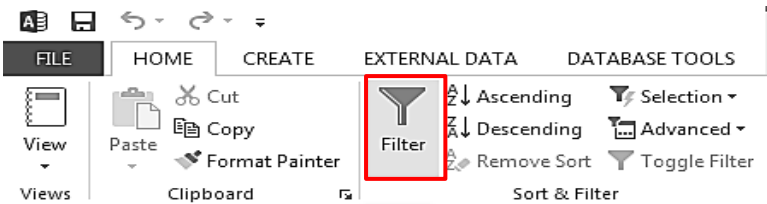
Pilih dan klik *field* pada *Layout View* misalnya *field* NIK.

Laporan Data Karyawan					03 Maret 2015
Laporan Data Karyawan					22:40:47
NIK	NamaKaryawan	AlamatLengkap	Handphone	JenisKelamin	
KYW-002-08-77		Jl. Simpang 02 Jakarta	878143666	Laki-laki	
KYW-001-09-10	Susanti Anggraeni	Jl. Kebon Jeruk 09 Jakarta	982919111	Perempuan	
KYW-003-10-11	I Made Idha Nuantara	Jl. Kebon Sirih 1 Jakarta	992871722	Perempuan	
KYW-004-10-11	Albert Sitohang	Jl. Siliwangi Jakarta	828919189	Laki-laki	
KYW-005-07-11	I Nyoman Windu Asmara	Jl. Didina 9 Jakarta	928282827	Laki-laki	
KYW-005-08-11	Lee Xing Wee	Jl. Gatot Subroto Jakarta	918181818	Laki-laki	

Gambar 12.20 Area *Layout View* Report

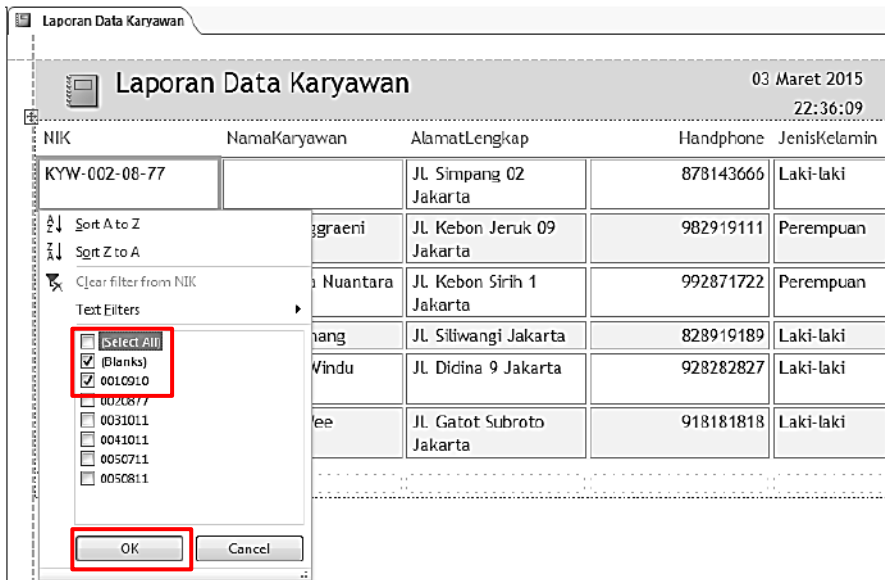


Klik **Ribbon Tab HOME** → pilih **Filter**.



Gambar 12.21 Filter untuk menyaring data

Kemudian muncul area dari **Text Filters**, **Select All** untuk memilih semua *record* untuk ditampilkan, **Blanks** untuk tidak menampilkan data apapun di dalam *report*, pilih misalnya **record NIK 0010910** berarti hanya data ini yang ditampilkan di dalam *report* selainnya tidak akan ditampilkan.



Gambar 12.22 Area Layout View Report

Berikut ini adalah hasil dari menyaring data (data *filter*).

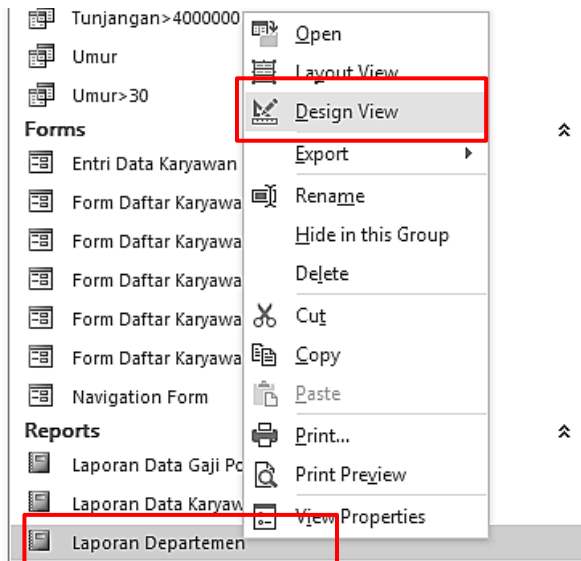


Laporan Data Karyawan					03 Maret 2015
					22:37:01
NIK	NamaKaryawan	AlamatLengkap	Handphone	JenisKelamin	
KYW-001-09-10	Susanti Anggraeni	Jl. Kebon Jeruk 09 Jakarta	982919111	Perempuan	

Gambar 12.23 Report Laporan Data Karyawan

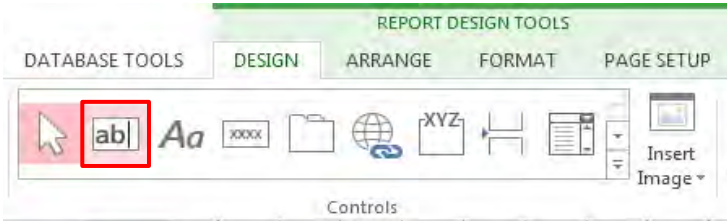
12.4 Menghitung Kelompok Data pada Report

Untuk menghitung kelompok data pada *report* dengan cara: klik dan pilih *report* → klik kanan → pilih **Design View**.



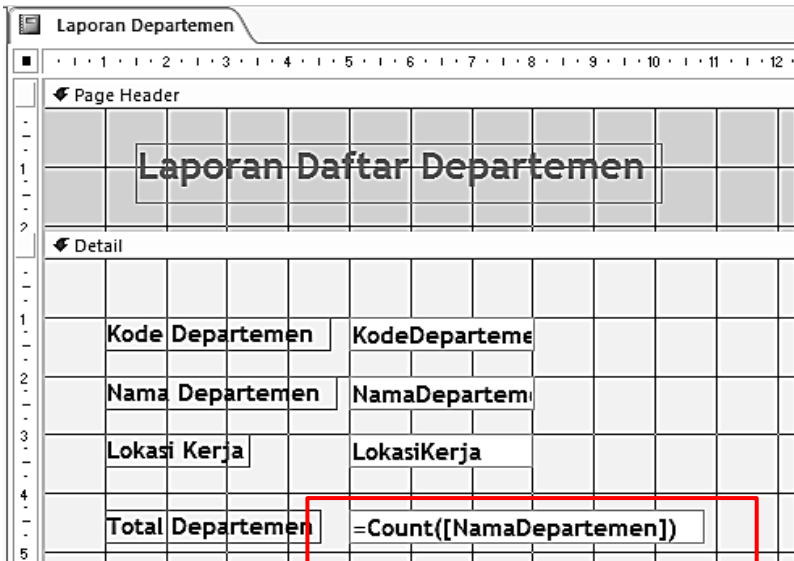
Gambar 12.24 Area Navigation Pane

Pilih dan klik tombol **Text Box** pada **Group Controls**.



Gambar 12.25 Area Controls

Arahkan ke dalam area *Design View report* → ketikkan rumus `=Count ([NamaDepartemen])` pada *Unbound*. Kita akan menghitung kelompok data dari ada berapa data dalam *field NamaDepartemen*. Rumus tadi untuk menghitung sub total data berdasarkan kriteria *NamaDepartemen*.



Gambar 12.26 Area *Design View*



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Catatan: Selain Anda dapat menggunakan **Count**, Anda juga dapat menggunakan fungsi lain seperti fungsi menghitung jumlah (**Sum**), rata-rata (**Avg**), nilai tertinggi (**Max**), nilai terendah (**Min**), dan lain-lain. Berikut adalah hasil dari *report* yang telah dibuat.

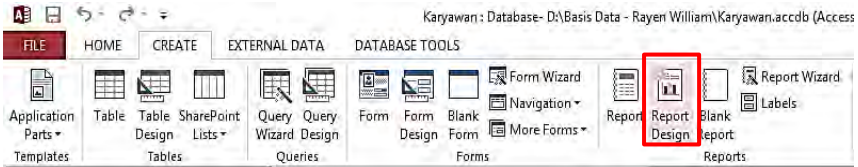
The screenshot shows a report window titled "Laporan Departemen". Inside, there is a header "Laporan Daftar Departemen". Below the header is a form with four rows of data:

Kode Departemen	1
Nama Departemen	Produksi
Lokasi Kerja	Cikarang
Total Departemen	4

Gambar 12.27 *Report* Laporan Daftar Departemen

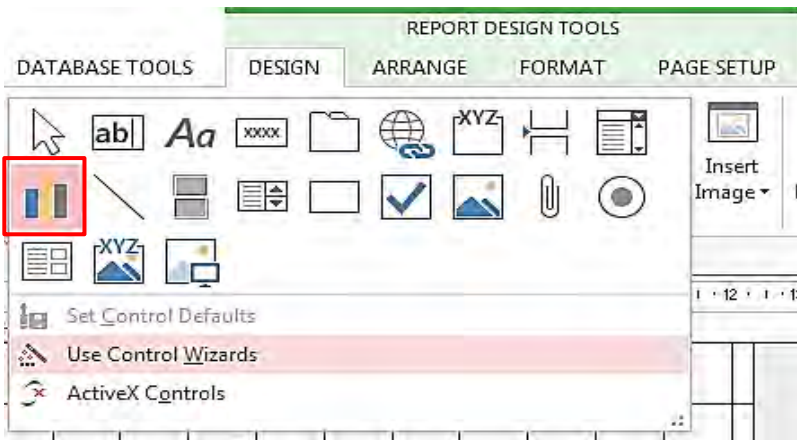
12.5 Membuat Grafik pada *Report*

Grafik diperlukan untuk mengetahui perkembangan dari semua data/berkaitan dengan capaian atau hasil dari sebuah data yang ingin kita lihat. Misalnya grafik gaji pokok karyawan tertinggi artinya melihat data berkaitan dengan gaji dalam bentuk format grafik agar dapat dilihat dengan mudah tingkat capaiannya. Grafik dapat diolah melalui *form* atau *report*. Dalam hal ini kita bekerja di *report* karena sebagai *output* data berupa laporan yang menarik dan mudah untuk dipahami. Berikut cara pembuatan grafik pada *report*: klik **Ribbon Tab CREATE** → klik **Report Design**.



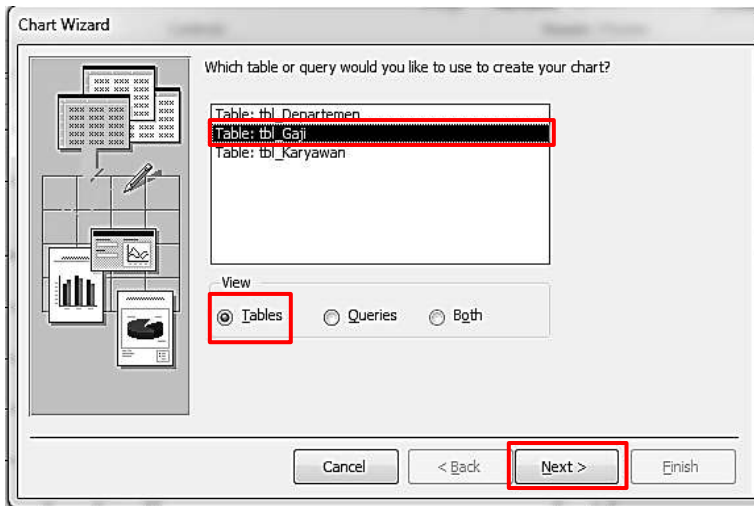
Gambar 12.28 Membuat Grafik dengan Report Design

Klik **Control Grafik** → seret ke dalam area **Design View Report**.



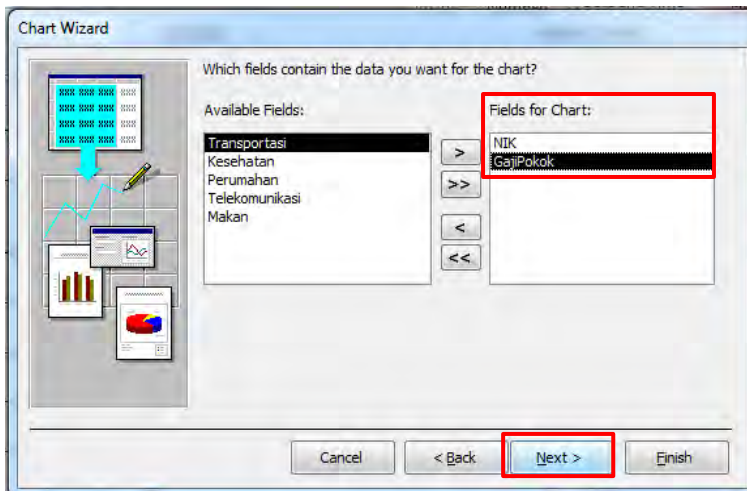
Gambar 12.29 Area Controls

Muncul kotak dialog **Chart Wizard** → pilih *table/query/both* yang ingin dibuat menjadi sebuah grafik → klik **Next**.



Gambar 12.30 Kotak Dialog *Chart Wizard*

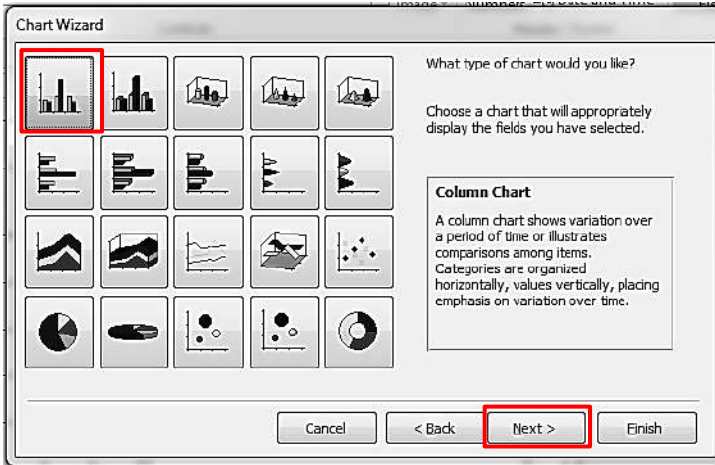
Pilih *field* yang ingin ditampilkan pada grafik → klik *Next*.



Gambar 12.31 Kotak Dialog *Chart Wizard*

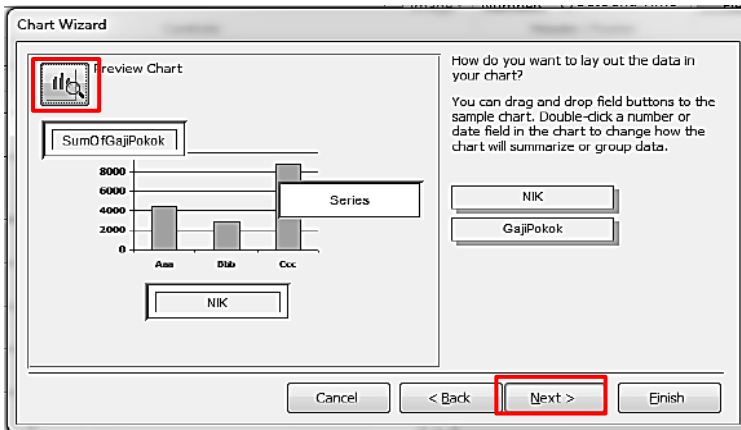


Pilih bentuk grafik → klik *Next*.



Gambar 12.32 Kotak Dialog *Chart Wizard*

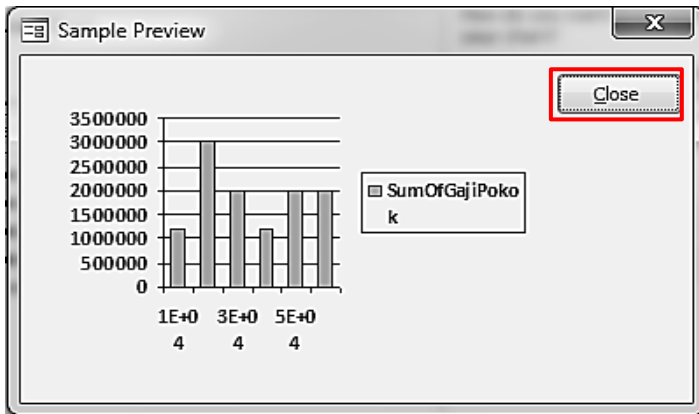
Atur bagaimana tata letak data pada grafik, untuk melihat grafik yang akan ditampilkan klik *Preview Chart* → *Next*.



Gambar 12.33 Kotak Dialog *Chart Wizard*

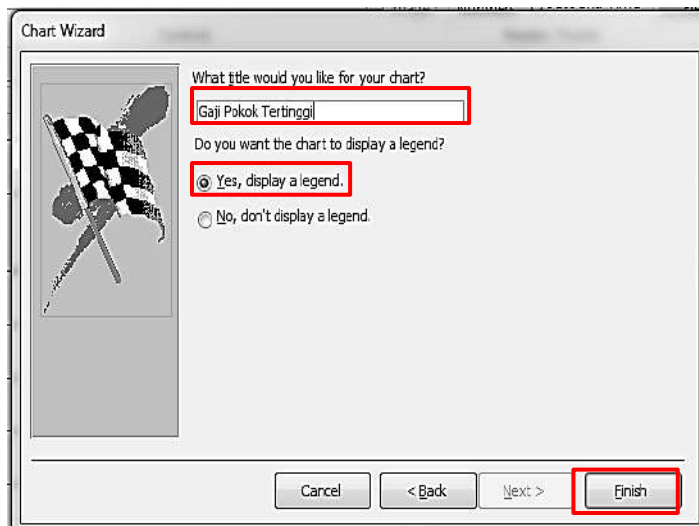


Tampilan *Preview Chart* → klik *Close*.



Gambar 12.34 Kotak Dialog *Sample Preview*

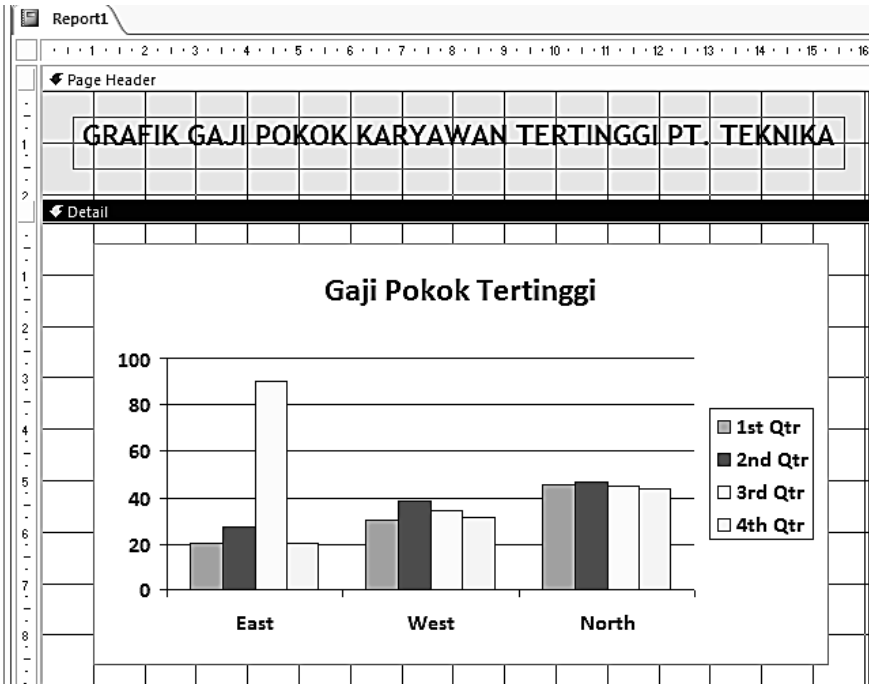
Berikan nama pada grafik → klik *Yes, display a legend.* → klik *Finish*.



Gambar 12.35 Kotak Dialog *Chart Wizard*

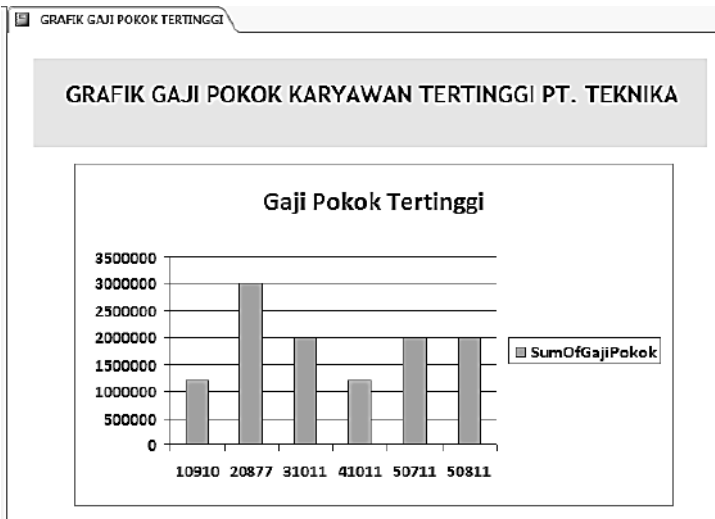


Muncul area *Design View* dari grafik yang kita buat pada *Report* → modifikasi dan atur tata letak grafik berikan judul dan lain-lain sebagainya.



Gambar 12.36 Area *Design View Report*

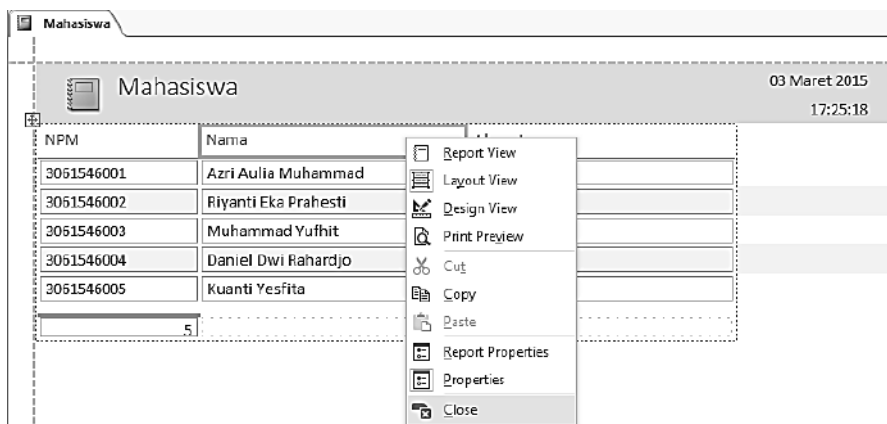
Hasil pembuatan grafik pada *report*. Didapat informasi bahwa NIK 20877 menempati posisi gaji pokok tertinggi karyawan PT. Teknika.



Gambar 12.37 Tampilan Grafik di dalam Report


12.6 Menutup dan Menyimpan Report

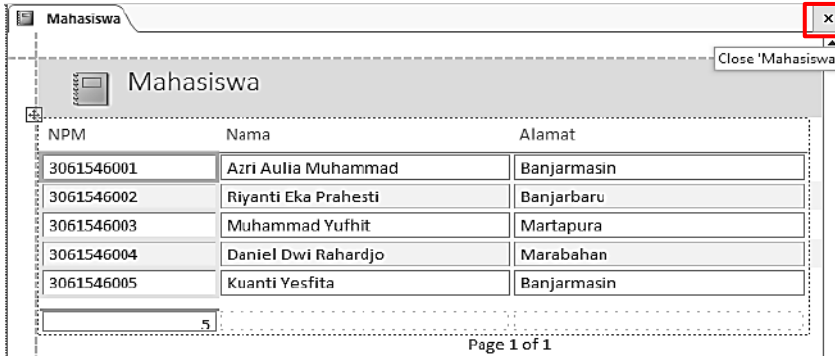
Untuk menutup dan menyimpan *report* yang telah Anda buat, dengan cara: klik kanan area *Design View* pada *Report* → klik *Close*.



Gambar 12.38 Area *Design View* Report

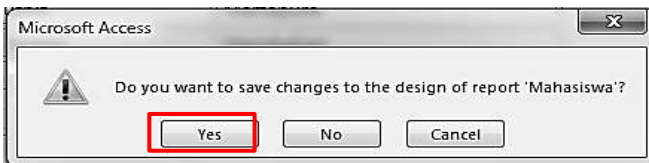


Klik tombol *close*  untuk keluar dari *Design View Report*.



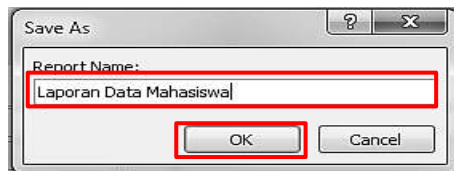
Gambar 12.39 Area *Design View Report*

Muncul kotak dialog Microsoft Office Access 2013 konfirmasi penyimpanan perubahan desain *report* → klik *Yes*.



Gambar 12.40 Kotak Dialog Microsoft Office Access 2013

Muncul kotak dialog *Save As* → ketikkan **nama report** → klik *OK*.



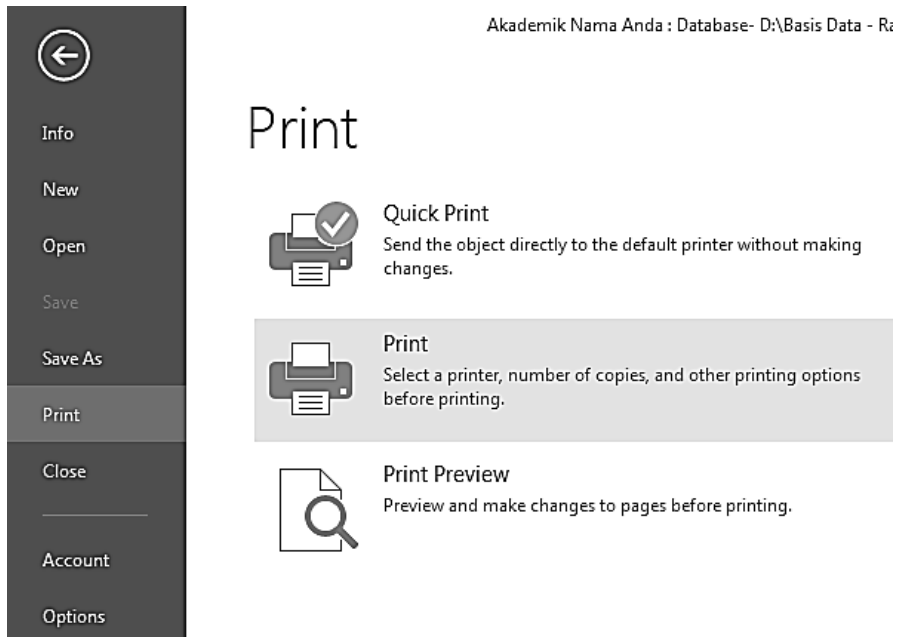
Gambar 12.41 Kotak Dialog *Save As*



12.7 Mencetak Report

Sebenarnya kita bisa mencetak data tanpa perlu membuat *report*, namun hasilnya tidak sebaik jika kita membuat desain *report*-nya. Untuk mencetak data diperlukan terlebih dahulu pembuatan *Report*, **cara** mencetak sebagai berikut: klik **Menu FILE** → klik **Print**. Di sini terdapat 3 pilihan, yaitu:

- **Quick Print**, akan segera melakukan pencetakan pada *printer default*.
- **Print**, akan menampilkan kotak dialog di mana kita bisa memilih *printer* dan melakukan *setup* lainnya.
- **Print Preview**, akan menampilkan hasil cetakan di layar.



Gambar 12.42 Tampilan Menu Print



Mahasiswa

Mahasiswa 03 Maret 2015 17:11:36

NPM	Nama	Alamat
3061546001	Azri Aulia Muhammad	Banjarmasin
3061546002	Riyanti Eka Prahesti	Banjarbaru
3061546003	Muhammad Yufit	Martapura
3061546004	Daniel Dwi Rahardjo	Marabahan
3061546005	Kuantu Yeslita	Banjarmasin

Gambar 12.43 Tampilan Hasil dari *Print Preview* dari *Report Mahasiswa*

Latihan Praktik

1. Buka kembali *database Akademik Nama Anda* pada praktikum sebelumnya!
2. Buatlah *Report* menggunakan *Report Wizard (step by step)* dari tabel *Mempelajari*, berikan nama *report Daftar Nilai Mahasiswa PTI*. Seperti gambar di bawah ini:

Daftar Nilai Mahasiswa PTI

Daftar Nilai Mahasiswa Pendidikan Teknologi Informasi

NPM	KdMK	Nilai
3061546001	BMKU1	70
	BTIK32	80
	BTIK33	90
	BTIK34	70
	BKDK1	60
3061546002	BKDK1	80
	BMKU1	79
	BTIK32	75
	BTIK33	75
	BTIK34	80
3061546003	BTIK33	85
	BKDK1	80
	BTIK32	89

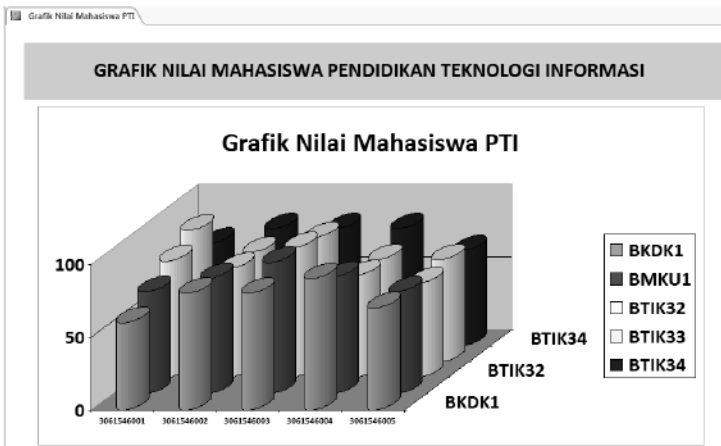


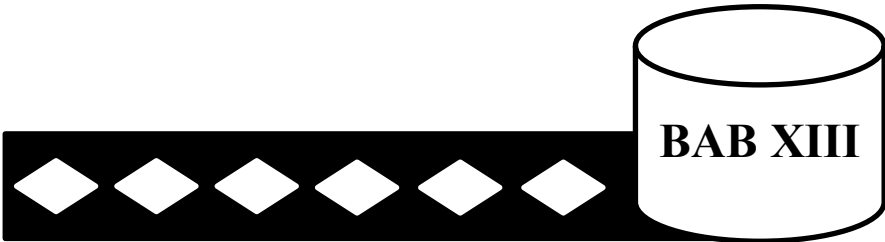
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

3. Tambahkan informasi baru berupa **Keterangan**, tambahkan **text box** untuk Keterangan beri rumus pada *Unbound* = **IIf([Nilai]>=85;"Memuaskan";"Baik")**. Sehingga menghasilkan *report* sebagai berikut:

NPM	KdMK	Nilai	Keterangan	
3061546001	BMKU1	70	Keterangan	Baik
	BTIK32	80	Keterangan	Baik
	BTIK33	90	Keterangan	Memuaskan
	BTIK34	70	Keterangan	Baik

4. Tampilkan data pada *report* hanya data yang memiliki NPM **3061546003** dan **3061546005**
5. Buatlah **grafik** di dalam *Report Design* dari tabel **Mempelajari** dengan tipe **Cylinder Column Chart** beri nama dengan **Grafik Nilai Mahasiswa PTI**, seperti gambar di bawah ini:





MEMBUAT MENU NAVIGASI DAN *LOGIN*



Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu untuk mengetahui, memahami, merancang, membuat, dan menguasai konsep dasar mengenai basis data dengan baik, sehingga dapat mengimplementasikannya menggunakan Microsoft Office Access 2013 sesuai dengan prinsip dan aturan yang berlaku di dalam basis data.

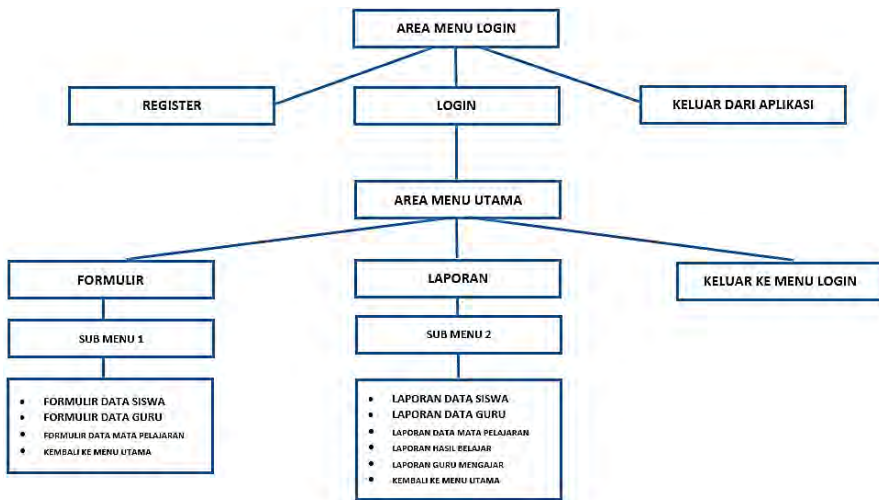
Tujuan Instruksional Khusus :

1. Mahasiswa mampu membuat menu navigasi dengan *form*
2. Mahasiswa mampu memberikan perintah dengan *macro*
3. Mahasiswa mampu memberikan perintah dengan Visual Basic Application (VBA)
4. Mahasiswa mampu membuat keamanan hak akses data dengan *form login*
5. Mahasiswa mampu mengatur tampilan awal ketika sebuah *database* dijalankan



13.1 Menu Navigasi dan Login

Untuk memudahkan navigasi semua objek pada sebuah *file database* yang telah dirancang dan dibuat, Anda dapat membuat *menu* navigasi buatan sendiri. Sebagai contoh Anda akan membuat *menu* navigasi untuk memudahkan Anda dalam membuka dan mengedit objek *database* dengan struktur sebagai berikut:



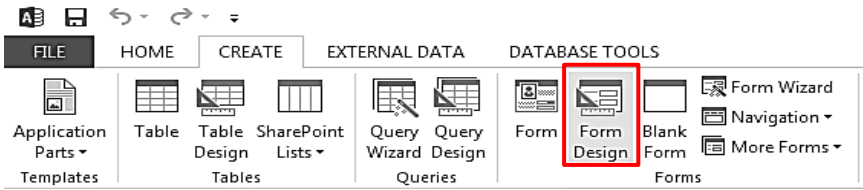
Gambar 13.1 Struktur Menu Navigasi dan Login

Kemudian ketika sebuah *menu* tersebut dibuat, untuk menjaga keamanan data serta diberikan hak akses pemakai ke dalam *database*, maka diperlukan *menu login* untuk memberikan *security* pada sebuah *database*, artinya suatu proteksi data terhadap perusakan atau penyalahgunaan data dan pemakaian data oleh pemakai yang tidak memiliki izin/hak akses.



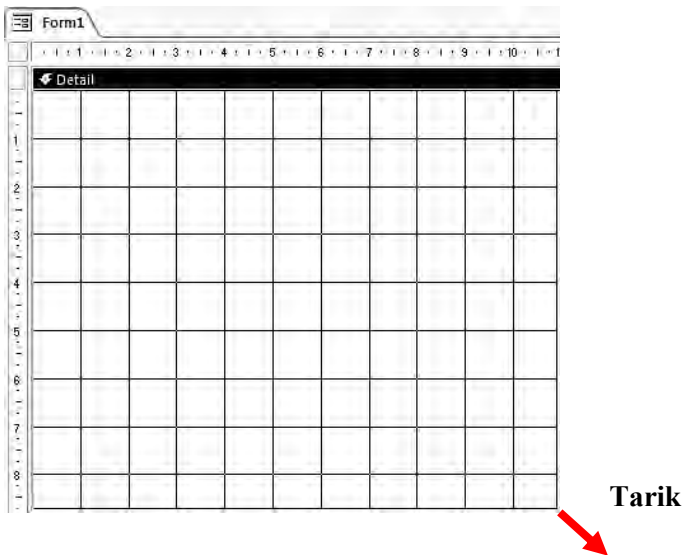
13.2 Membuat Menu Navigasi dengan Form, Macro, dan Visual Basic Application (VBA)

Untuk merancang dan membuat menu navigasi dengan *form* berikut adalah cara: buka *file database* yang Anda inginkan → pada jendela kerja klik **Ribbon CREATE** → klik **Form Design**.



Gambar 13.2 Ribbon Form Design

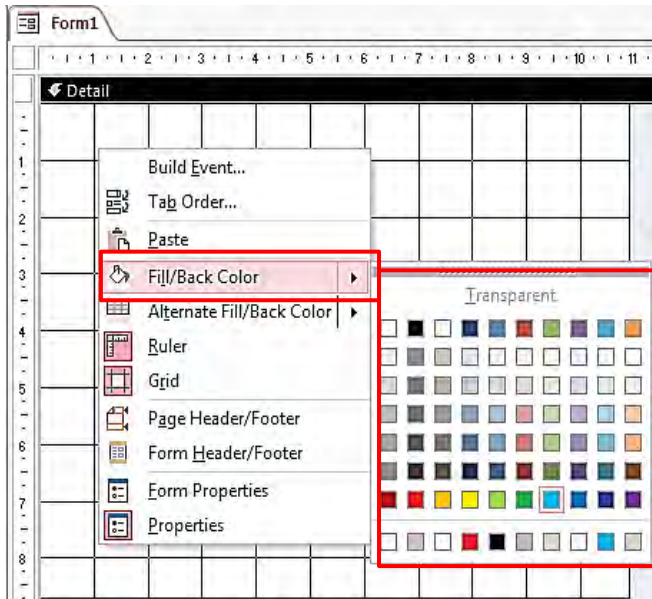
Perbesar area *form* yang akan digunakan untuk penempatan *menu*, dengan cara menggeser (*drag*) garis batas *design view form*.




Gambar 13.3 Area Design View

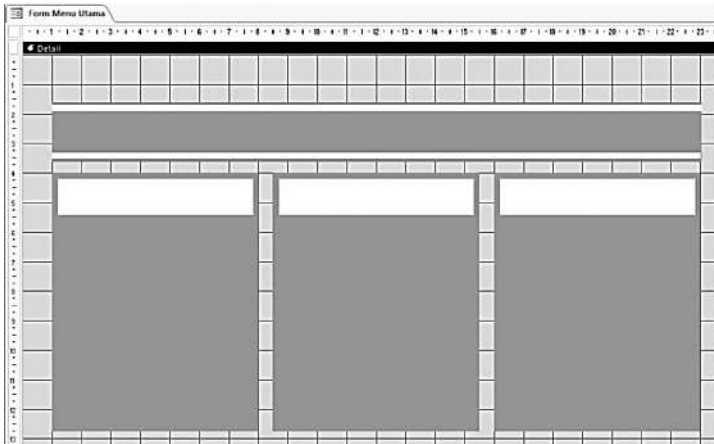


Bila perlu beri warna latar belakang, dengan memilih dan mengklik kanan area **Design View Form** → klik **Fill/Back Color**.



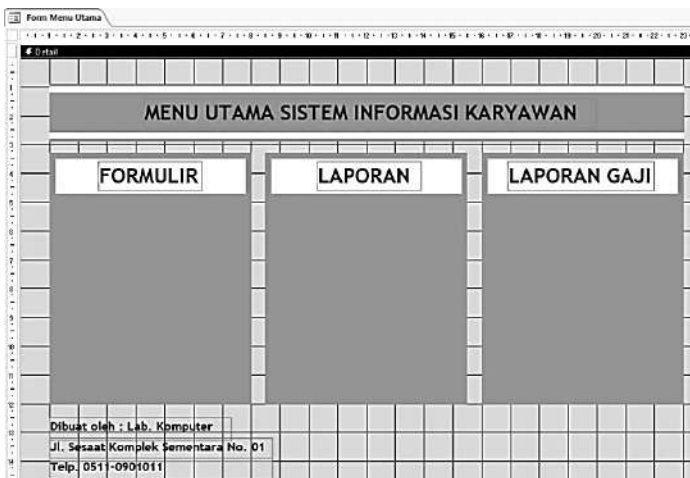
Gambar 13.4 Membuat Tampilan Warna Belakang *Form*

Buatlah tampilan dengan bantuan **Form Design Tools** misalnya dengan menggunakan **Controls Rectangle** , bila perlu beri warna latar belakang yang berbeda pada setiap bingkai, dengan memilih dan mengklik tombol **Fill/Back Color**, seperti gambar dibawah ini:



Gambar 13.5 Merancang Tampilan *Form*

Lengkapi *menu* navigasi tersebut dengan teks dan judul dengan menggunakan *Controls Label*. Sesuaikan penempatannya seperti gambar di bawah ini.

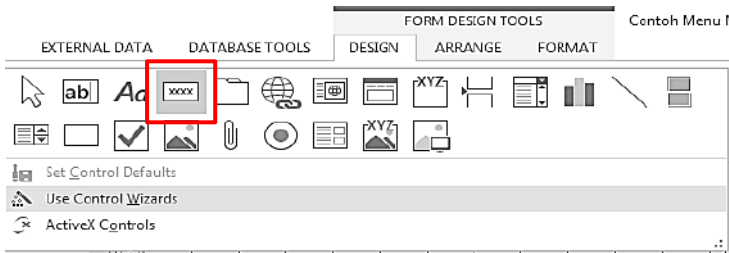


Gambar 13.6 Memberikan Judul dan Lain-Lain



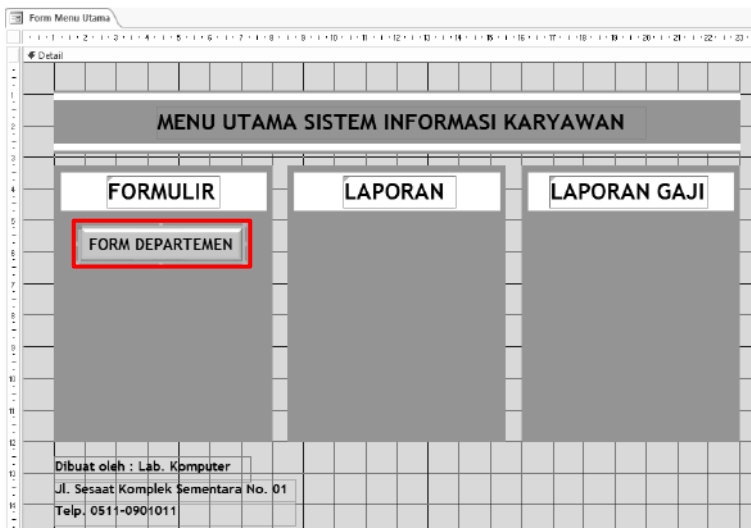
Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

Lanjutkan dengan menempatkan tombol perintah (*Command Button*) sebagai tempat pemilihan menu. Klik tombol **Button** → arahkan ke area *Design View Form*.




Gambar 13.7 Tombol *Button*

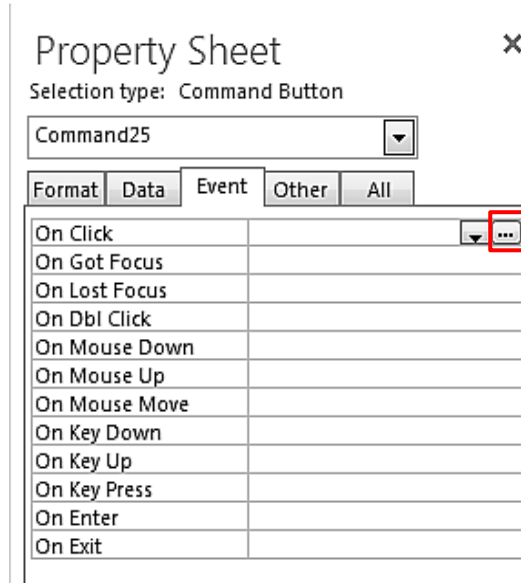
Untuk memberikan sebuah *action* pada tombol perintah **Button** → klik misalnya tombol **FORM DEPARTEMEN** → klik pada bagian *Property Sheet*.



Gambar 13.8 Tampilan *Design View Form*

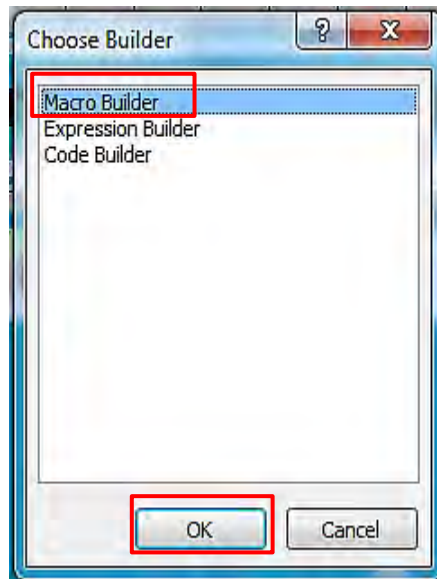


Pada area **Property Sheet** punyanya tombol perintah **FORM DEPARTEMEN** → klik **Menu Event** → klik tombol  pada **On Click**.



Gambar 13.9 Tombol **On Click**

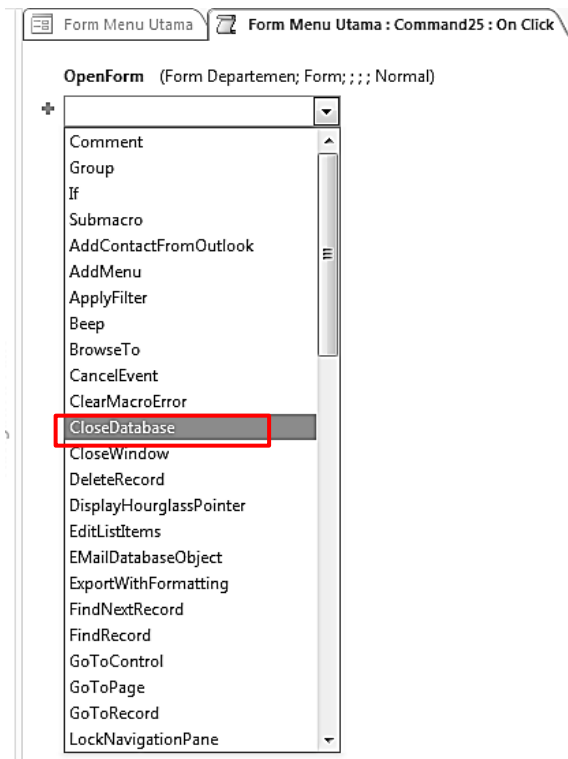
Muncul kotak dialog *Choose Builder* → pilih dan klik **Macro Builder**.



Gambar 13.10 Kotak Dialog *Choose Builder*

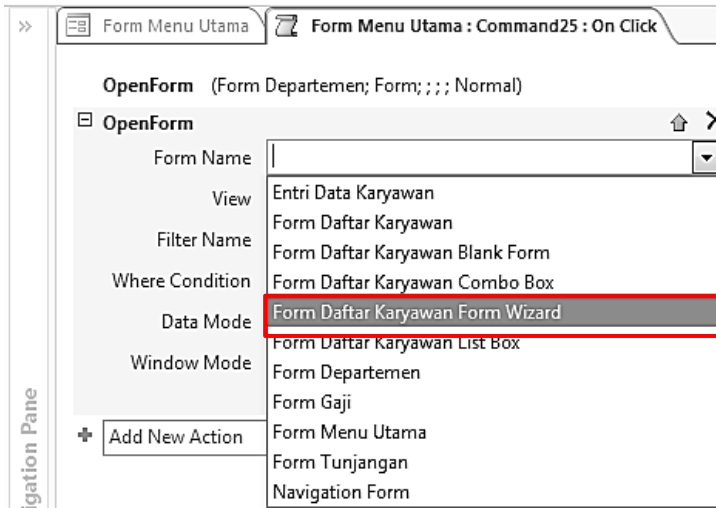
Macro adalah sarana untuk memberikan/menjalankan perintah. Misalnya kita ingin memberikan aksi kepada tombol dan lain-lain sebagainya. Pada Microsoft Office Access 2013 disediakan sarana untuk memberikan perintah ada dua, yaitu melalui **macro** dan **module**. Pada **module** kita menuliskan perintah dalam bahasa pemrograman, sedangkan pada **macro** berisi perintah-perintah yang telah disediakan dan tinggal memilih **action**.

Kemudian muncul area dari **Macro Builder**. Pada bagian + **Add New Action** itu akan ada banyak aksi/**action** misalnya **OpenForm**, **GoRoRecord**, **CloseDatabase**, dan lain-lain.



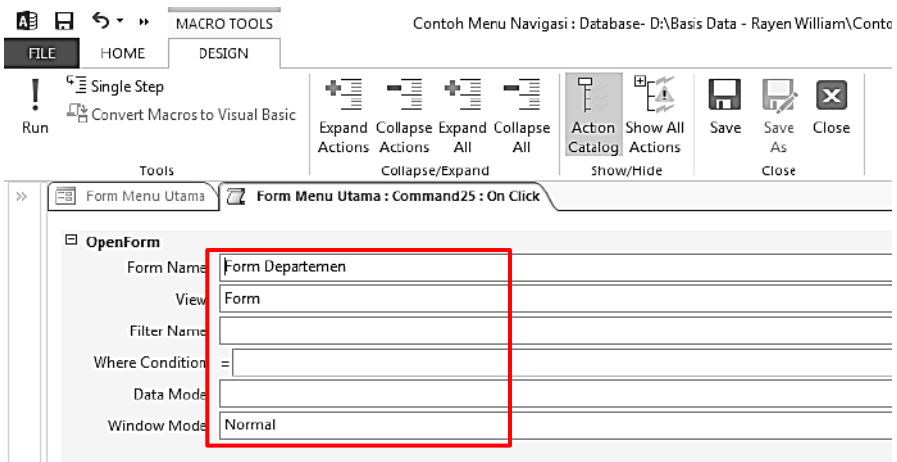
Gambar 13.11 Area Macro Builder

Pada bagian *Form Name* terdapat pilihan jenis *Form* yang mana yang ingin ditampilkan.



Gambar 13.12 Area Macro Builder

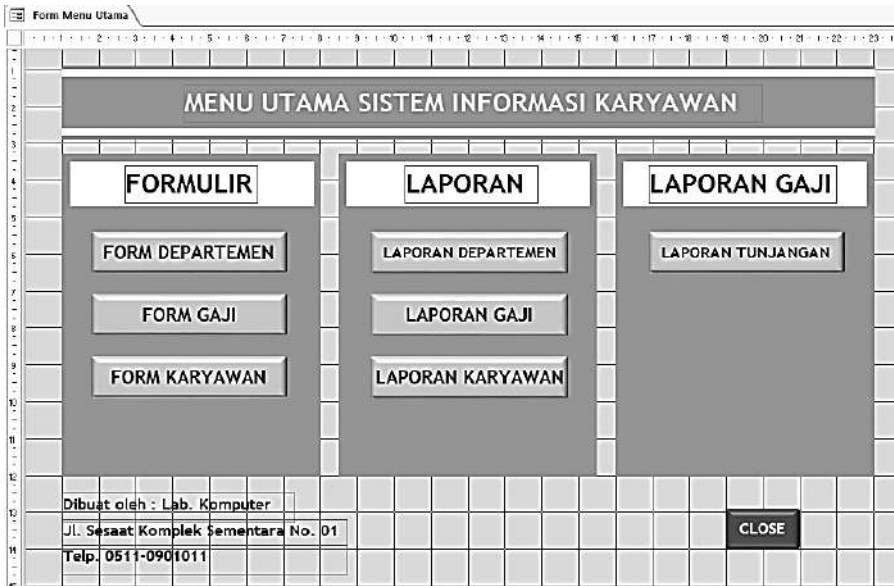
Misalnya kita akan memberikan *action* pada tombol perintah **FORM DEPARTEMEN**.



Gambar 13.13 Area Action Open Form



Lengkapi tombol perintah seperti dibawah ini dan berikan *action* melalui *Macro Builder*.



Gambar 13.14 Tampilan *form menu navigasi*

Simpan *form* dengan nama **UTAMA**, berikut hasil tampilan dari *menu navigasi* yang telah dibuat:



MENU UTAMA SISTEM INFORMASI KARYAWAN

FORMULIR

FORM DEPARTEMEN

FORM GAJI

FORM KARYAWAN

LAPORAN

LAPORAN DEPARTEMEN

LAPORAN GAJI

LAPORAN KARYAWAN

LAPORAN GAJI

LAPORAN TUNJANGAN


Dibuat oleh : Lab. Komputer
Jl. Sesaat Komplek Sementara No. 01
Telp. 0511-0901011

CLOSE

Gambar 13.15 *Form View* pada *Form Menu Navigasi*

13.3 Membuat Keamanan Data dengan *Form Login*

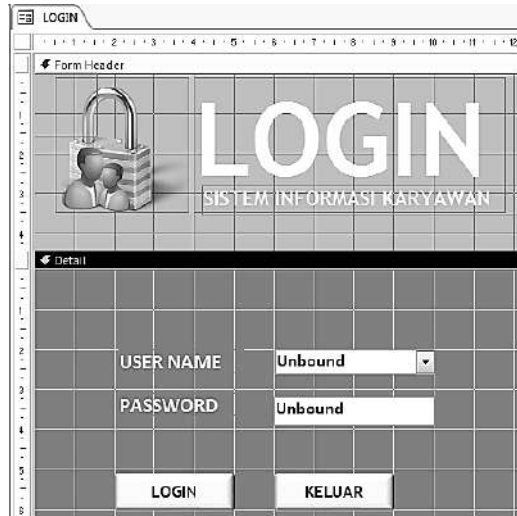
Untuk membuat *menu login* dengan *form* berikut adalah cara: buka *file database* yang Anda inginkan → buatlah *table* dengan nama dengan nama **USER** dengan struktur data sebagai berikut:



	Field Name	Data Type
PK	UserId	AutoNumber
	UserName	Short Text
	UserPassword	Short Text

Gambar 13.16 Struktur *Table USER*

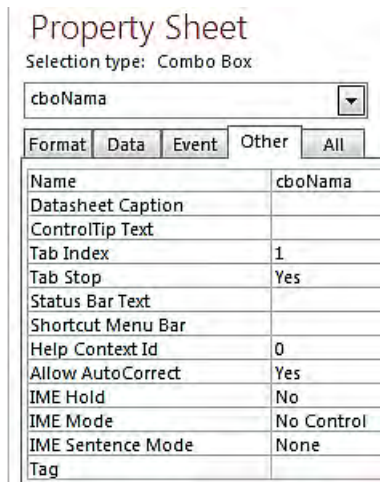
Untuk *field UserPassword* berikan *Field Size: 5* dan *Input Mask: Password*. Buatlah *form login* dengan *Form Design* sebagai berikut:



Gambar 13.17 Area Design View form Login

Ketentuan:

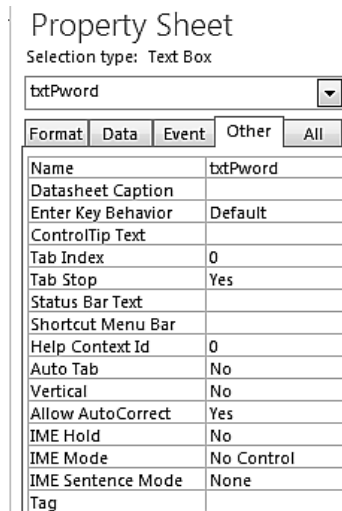
1. Untuk **Combo Box User Name** beri nama dengan **CboNama**.



Gambar 13.18 Area Property Sheet untuk Combo Box User Name

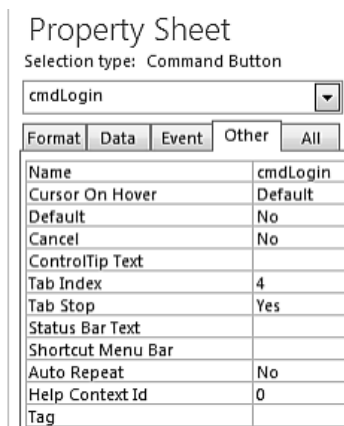


- Untuk **Text Box Password** beri nama dengan **txtPword**.



Gambar 13.19 Area Property Sheet untuk Text Box Password

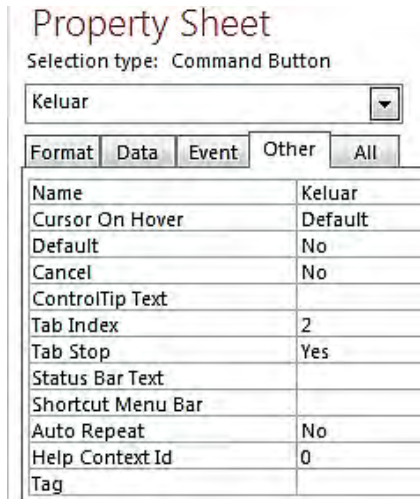
- Untuk **Button Login** beri nama dengan **cmdLogin**.



Gambar 13.20 Area Property Sheet untuk Button Login



4. Untuk **Button Keluar** beri nama dengan **Keluar**.



Gambar 13.21 Area *Property Sheet* untuk **Button Keluar**

5. Untuk **Button Login** isikan **Code Builder** sebagai berikut:
Menggunakan perintah **Visual Basic Application (VBA)**.

```
If IsNull(Me.cboNama) Or Me.cboNama = "" Then
    MsgBox "Pilih User Name anda!", vbOKOnly,
    "Masukkan User Name!"
    Me.cboNama.SetFocus
    Exit Sub
End If
```

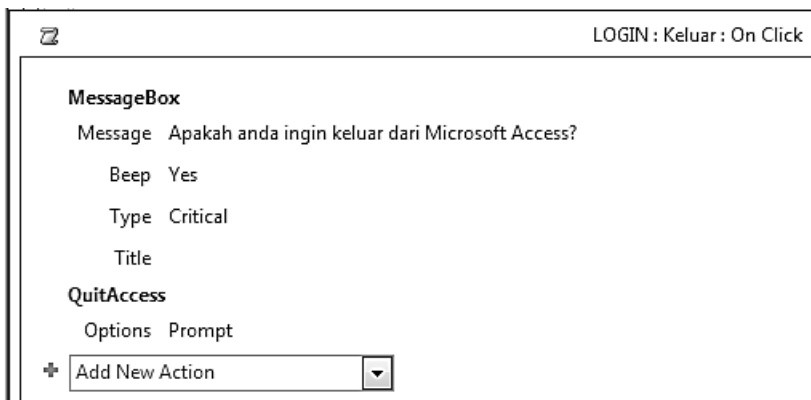
```
If IsNull(Me.txtPword) Or Me.txtPword = "" Then
    MsgBox "Masukkan Password anda!", vbOKOnly,
    "Masukkan Password!"
    Me.txtPword.SetFocus
    Exit Sub
End If
```



Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA

```
If Me.txtPword.Value = DLookup("UserPassword", "USER",  
"[UserID]=" & Me.cboNama.Value) Then  
    MyUserID = Me.cboNama.Value  
    MsgBox "Login Succes. Thank You!", vbOKOnly,  
    "Message Hellion"  
    DoCmd.Close acForm, "LOGIN", acSaveNo  
    DoCmd.OpenForm "UTAMA"  
  
Else  
    MsgBox "Password yang anda masukkan salah, review  
    kembali!", vbCritical, "Hellion Message"  
    Me.txtPword.SetFocus  
End If
```

6. Untuk **Button Keluar** atur **Macro Builder** sebagai berikut:



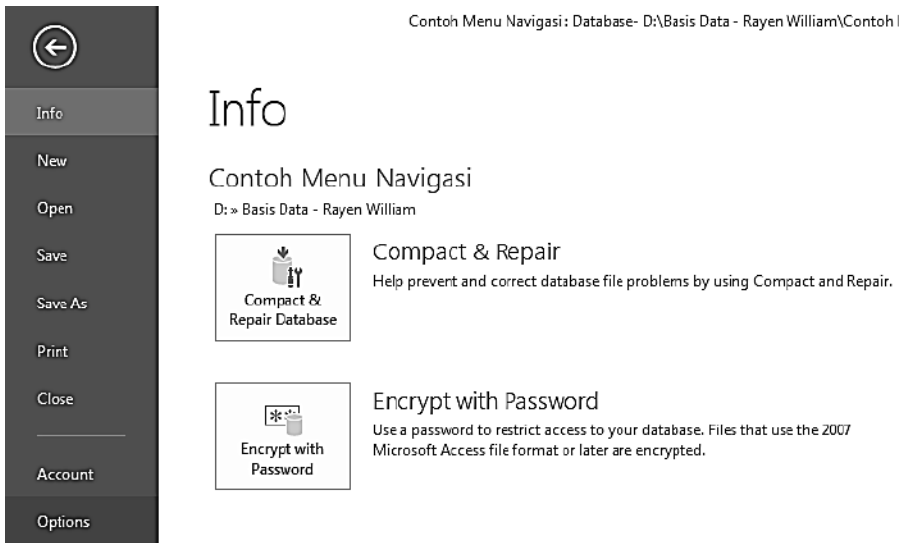
Gambar 13.22 Area *Macro Builder* untuk Aksi *Button Keluar*



13.4 Mengatur Tampilan Awal Ketika Database Dijalankan

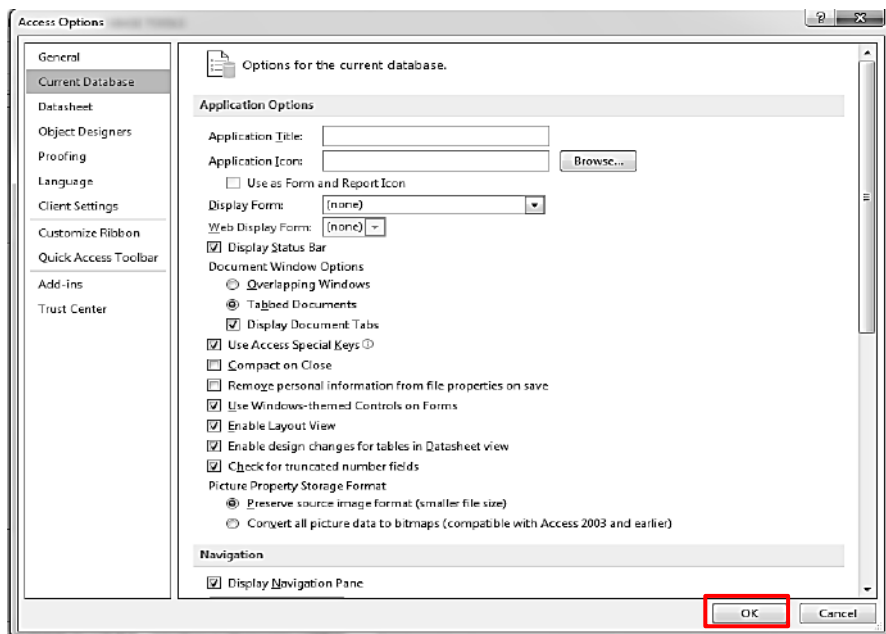
Untuk mengatur tampilan awal ketika *database* dijalankan secara otomatis, **caranya** sebagai berikut:

Buka dan aktifkan *database* yang diinginkan kemudian klik **Ribbon Tab FILE** → klik **Options**.

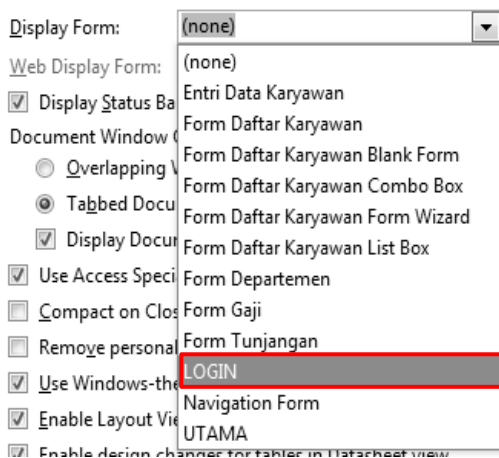


Gambar 13.23 Tab Ribbon FILE

Muncul kotak dialog **Access Options** → klik **Current Database**. Pada bagian **Display Form** → pilih *form* yang ingin ditampilkan di awal misalnya **LOGIN** → klik **OK**.



Gambar 13.24 Kotak Dialog *Access Options*



Gambar 13.25 Area *Display Form*



Tutup dulu *file database* kemudian buka kembali, dan ketika *database* dijalankan maka yang akan pertama kali muncul adalah *menu LOGIN*.

Latihan Praktik

Buatlah *Menu Navigasi* dan *Menu Login* pada *database Akademik Nama Anda* pada praktikum sebelumnya!

Proyek Akhir Mata Kuliah

1. Buatlah aplikasi *database* menggunakan **Microsoft Office Access 2013**, tentukan sendiri aplikasi apa yang ingin dijadikan sebagai Proyek Akhir Mata Kuliah!
2. Buatlah konsep basis data berupa *Entity Relationship Diagram* (ERD) dan struktur datanya terlebih dahulu!
3. *Database* tersebut harus memiliki:
 - a. *Table* Acuan
 - b. *Table* Relasi
 - c. *Query*
 - d. *Form*
 - e. *Report*
 - f. *Menu* Navigasi
 - g. *Menu Security/Login*
4. Setelah *database* dibuat, masukkan data-data minimal 15 *record*!
5. Buat juga tutorial penggunaan aplikasi tersebut!
6. Buat aplikasi sekreatif mungkin dan jangan menghilangkan konsep relasi antar *table*!



DAFTAR PUSTAKA

- Cox, Joyce. Joan Lambert. 2013. *Step by Step Microsoft Access 2013 (Build Exactly The Skills You Need. Learn At The Pace You Want)*. Redmond, Washington: Microsoft Press.
- Elmasri, Ramez. Sam Navathe. 2000. *Fundamentals of Database Systems – 4rd Edition*. Addison Wesley Publishing Company.
- Güting, Ralf Hartmut. Markus Schneider. 2005. *Moving Objects Databases*. San Francisco: Morgan Kauffman Publishers.
- Kadir, Abdul. 2002. *Penuntun Praktis Belajar Database menggunakan Microsoft Access*. Yogyakarta: Penerbit Andi.
- Madcoms. 2013. *Microsoft Access 2013 untuk Pemula*. Yogyakarta: Penerbit ANDI.
- _____. 2011. *Memaksimalkan Rumus dan Fungsi Microsoft Access 2010*. Yogyakarta : Penerbit Andi dan Madcoms.
- McFadden, Fred R. Et al 1999. *Modern Database Management*. Upper Saddle River: Addison Wesley Educational Publisher Inc.
- Nugroho, Adi. 2011. *Perancangan dan Implementasi Sistem Basis Data*. Yogyakarta: Penerbit Andi.
- Pallaw, Vijay Krishna. 2008. *Database Management Systems – Second Edition*. New Delhi: Compudata Services Asian Books Pvt. Ltd.
- Ramakrishnan, Raghu. Johannes Gehrke. 2003. *Database Management Systems – Third Edition*. The McGraw-Hill Companies, Inc.
- S, Rosa A. M. Shalahuddin. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Sadeli, Muhammad. 2012. *Aplikasi Bisnis dengan Access 2010 untuk Orang Awam*. Palembang : Maxikom.

Subandi, S.T., M.Kom. & Aulia Akhrian Syahidi, S.Pd., MTA



Sadeli, Muhammad. 2013. *7 Jam Belajar Interaktif Access 2013 untuk Orang Awam*. Palembang : Maxikom.

Talib, Haer. 2014. *Panduan Lengkap Ms Access 2013*. Jakarta: PT. Elex Media Komputindo.

Utomo, Wiranto Herry. 2010. *Pemodelan Basis Data Berorientasi Objek*. Yogyakarta: Penerbit ANDI.

Whitehorn, Mark. Bill Marklyn. 2003. *Inside Relational Databases – Second Edition*. Jakarta: Penerbit Erlangga.

Widianti, Sri. 2009. *Pengantar Basis Data*. Jakarta: Lentera Ilmu Cendekia.



TENTANG PENULIS



Subandi, S.T., M.Kom. lahir di Kediri, 20 Oktober 1965 adalah dosen tetap Program Studi Teknik Informatika Politeknik Negeri Banjarmasin dan dosen Program Studi Pendidikan Teknologi Informasi STKIP PGRI Banjarmasin. Lulusan dari S-1 Teknik Elektro Universitas Diponegoro (1995) dan S-2 Teknologi Informasi Sekolah Tinggi Teknik Surabaya (2007).

Sekarang ini menjabat sebagai Ketua Program Studi Pendidikan Teknologi Informasi STKIP PGRI Banjarmasin. Mata kuliah yang pernah diampu seperti Basis Data 1, Basis Data 2, Basis Data, Manajemen Proyek, Logika Informatika, Elektronika Digital, Sistem Digital, Kurikulum Pendidikan TIK, Perencanaan Pembelajaran TIK, dan lain-lain. Aktif sebagai anggota dari APTIKOM Indonesia. *E-mail Address:* subandi@poliban.ac.id

Aulia Akhrian Syahidi, S.Pd., MTA lahir di Rantau, 16 November 1994 adalah lulusan S-1 Program Studi Pendidikan Teknologi Informasi STKIP PGRI Banjarmasin (2016). Pengalaman di antaranya asisten laboratorium komputer Program Studi Pendidikan Teknologi Informasi STKIP PGRI Banjarmasin dan asisten mata kuliah Basis Data tahun 2014, tahun 2015, dan tahun 2016, mata kuliah Logika Informatika tahun 2014 dan tahun 2015, dan mata kuliah Dasar-Dasar Pengenalan Komputer tahun 2014. Aktif juga sebagai guru Teknik Komputer dan Jaringan (TKJ) SMK Muhammadiyah 1 Banjarmasin dan guru Rekayasa Perangkat Lunak (RPL) SMK Negeri 4 Banjarmasin. Telah lulus ujian dan meraih gelar MTA (Microsoft *Technology Associate*) untuk program 98-366: *Networking Fundamentals* oleh Microsoft *Learning by CERTIPORT id 90062723 certification number F549-3937* pada 22 Desember 2015. *E-mail Address:* auliaakhriansyahidi@gmail.com



BASIS DATA **Teori dan Praktik** **Menggunakan Microsoft Office Access**

Subandi, S.T., M. Kom.
Aulia Akhrian Syahidi, S.Pd., MTA.

Sistem Basis Data adalah sistem yang terdiri atas kumpulan tabel/file yang saling berhubungan dalam sebuah basis data dan sekumpulan program berupa DBMS yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi tabel-tabel tersebut. Dan dapat disimpulkan juga bahwa Sistem Basis Data mempunyai beberapa elemen penting yaitu basis data sebagai inti dari sistem basis data, software untuk mengelola dan memanipulasi basis data, hardware sebagai pendukung operasi pengolahan data sampai kepada tahap penyimpanan, serta manusia sebagai brainware yang mempunyai peran penting dalam sistem tersebut. Microsoft Office Access adalah sebuah software khusus untuk menyimpan dan mengelola data secara database. Jika anda mulai menyimpan data yang banyak, maka anda sudah membutuhkan database. Anda bisa saja menyimpan dan mengelola data menggunakan spreadsheet seperti Microsoft Excel, namun jika data tersebut sudah banyak dan penambahan data terus dilakukan. Anda perlu menggunakan software yang khusus disediakan untuk menyimpan dan mengelola data. Saat ini sebuah database yang mudah dan instan tidak ada lagi selain Microsoft Office Access. Ada yang masih menggunakan Microsoft Foxpro atau Microsoft Visual Foxpro, tetapi tidak semudah Microsoft Office Access. Software database kelas ini umumnya disebut desktop database. File database Access bisa terdiri dari sebuah file saja yang bisa dibawa kemana-mana, bisa dibuka di komputer mana saja yang mempunyai Microsoft Office Access versi yang sama atau yang lebih baru. Di dalam file Microsoft Office Access kita bisa membuat table untuk menyimpan data, membuat query, form, report, dan program aplikasi untuk mengelola data. Dalam buku ini akan dibahas teori konsep basis data dan praktik implementasi database menggunakan Microsoft Office Access 2013.



Penerbit Poliban Press

Redaksi :

**Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara**

Telp : (0511)3305052

Email : poliban@poliban.ac.id

ISBN 978-602-53609-3-6

