

1êr

PWA

Codelgniter

#### 0 D D 7 2 **北**お C 0 • $\mathbf{\bullet}$ (0) RMAS Ξ

RAMADHANI NOOR PRATAMA

۲

30

# PENERAPAN FRAMEWORK CODEIGNITER 3 PADA SISTEM INFORMASI PENJUALAN

#### Undang-Undang 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 Perlindungan Pasal 26

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

- 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 Rp 100.000.000 (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 Rp 500.000.000,000 (lima ratus juta rupiah).
- 3. 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 a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
- Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,000 (empat miliar rupiah).

# PENERAPAN FRAMEWORK CODEIGNITER 3 PADA SISTEM INFORMASI PENJUALAN

Ramadhani Noor Pratama



**Poliban Press** 

#### PENERAPAN FRAMEWORK

#### **CODEIGNITER 3 PADA**

#### SISTEM INFORMASI PENJUALAN

Penulis : Ramadhani Noor Pratama

> ISBN : 978-623-7694-54-0

ISBN Elektronik : 978-623-7694-55-7 (PDF)

Editor dan Penyunting : Adi Pratomo

**Desain Sampul dan Tata letak :** Eko Sabar Prihatin ; Rahma Indra

Penerbit :

POLIBAN PRESS Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia) no.004.098.1.06.2019 Cetakan Pertama, 2021

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

> **Diterbitkan pertama kali oleh :** Poliban Press, Banjarmasin, Desember 2021

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunianya sehingga buku Penerapan Framework CodeIgniter 3 Pada Sistem Informasi Penjualan telah dapat diselesaikan. Buku ini merupakan pengantar bagi mata kuliah Pemrograman Web 2.

Terimakasih disampaikan kepada Joni Riadi S.ST., M.T. selaku Direktur Politeknik Negeri Banjarmasin dan Nurmahaludin, S.T., M.T. selaku Ketua Pusat Penelitian dan Pengabdian Masyarakat beserta sekretaris dan staf. Terimakasih juga disampaikan kepada Faris Ade Irawan, Reza Fauzan, Eko Sabar Prihatin dan Rahma Indera yang telah berkontribusi dalam editing serta seluruh tim Poliban Press dan semua pihak yang telah ikut membantu dalam penyelesaian buku ini.

Kami menyadari masih terdapat kekurangan dalam buku ini untuk itu kritik dan saran terhadap penyempurnaan buku ini sangat diharapkan. Semoga buku ini dapat memberi maanfaat bagi semua pihak.

Banjarmasin,

Poliban Press

# **DAFTAR ISI**

5.2	Menambahkan Data Master Kota
5.3	Mengubah Data Master Kota
5.4	Menghapus Data Master Kota41
5.5	Menampilkan Data Master Supplier
5.6	Menambahkan Data Master Supplier
5.7	Mengubah Data Master Supplier
5.8	Menghapus Data Master Supplier
5.9	Latihan
BAB 6 I	nplementasi CRUD pada Data Transaksi
6.1	Menampilkan Data Transaksi Pembelian57
6.2	Menambahkan Data Transaksi Pembelian61
6.3	Mengubah Data Transaksi Pembelian67
6.4	Menghapus Data Transaksi Pembelian70
6.5	Menampilkan Data Transaksi Penjualan73
6.6	Menambahkan Data Transaksi Penjualan76
6.7	Mengubah Data Transaksi Penjualan82
6.8	Menghapus Data Transaksi Penjualan85
6.9	Latihan
BAB 7 N	1embuat Report
7.1	Membuat Laporan Pembelian
7.2	Membuat Laporan Penjualan95
7.3	Membuat Laporan Stock Akhir99
7.4	Latihan102
BAB 8 N	Iembuat Login Pengguna103
8.1	Session103
8.2	Membuat Halaman Login104
8.3	Latihan
BAB 9 A	Application Deployment
9.1	Instalasi dan Konfigurasi Program pada Website 109
9.2	Latihan112
GLOSAI	RIUM
DAFTAI	R PUSTAKA

# DAFTAR GAMBAR

Gambar 2. 1 Konfigurasi autoload pada CodeIgniter	7
Gambar 2. 2 Konfigurasi base URL pada CodeIgniter	7
Gambar 2. 3 Konfigurasi database pada CodeIgniter	7
Gambar 2. 4 Konfigurasi route pada CodeIgniter	7
Gambar 2. 5 Tampilan awal SI Penjualan	8
Gambar 3. 1 Cara kerja MVC	10
Gambar 4. 1 Membuat database	11
Gambar 4. 2 Struktur tabel barang	11
Gambar 4. 3 Struktur tabel karyawan	12
Gambar 4. 4 Struktur tabel kota	12
Gambar 4. 5 Struktur tabel pelanggan	12
Gambar 4. 6 Foreign Key pada tabel pelanggan	12
Gambar 4. 7 Struktur tabel supplier	12
Gambar 4. 8 Foreign Key pada tabel supplier	13
Gambar 4. 9 Struktur tabel penjualan	13
Gambar 4. 10 Foreign Key pada tabel penjualan	13
Gambar 4. 11 Struktur tabel itempenjualan	13
Gambar 4. 12 Foreign Key pada tabel itempenjualan	14
Gambar 4. 13 Struktur tabel pembelian	14
Gambar 4. 14 Foreign Key pada tabel pembelian	14
Gambar 4. 15 Struktur tabel itempembelian	14
Gambar 4. 16 Foreign Key pada tabel itempembelian	15
Gambar 5. 1 functionconstruct() pada controller Kota	30
Gambar 5. 2 class Model Kota Model	31
Gambar 5. 3 function index() pada controller Kota	31
Gambar 5. 4 Antarmuka daftar_kota	32
Gambar 5. 5 Menampilkan Data Master Kota	33
Gambar 5. 6 function create() pada controller Kota	34
Gambar 5. 7 function create_action() pada controller Kota	35
Gambar 5. 8 function _rules() pada controller Kota	35
Gambar 5. 9 function insert() pada Kota_model	36
Gambar 5. 10 Menambah Data Master Kota	37

Gambar 5. 14 function update() pada Kota\_model ...... 40 Gambar 5. 24 function create\_action pada controller Supplier..... 48 Gambar 5. 27 menambah data master supplier ......51 Gambar 5. 29 function edit action pada controller Supplier ....... 53 Gambar 5. 30 function get\_byid pada Supplier\_model......54 Gambar 6. 1 function \_\_\_\_\_construct() pada transaksi pembelian ..... 57 Gambar 6. 3 function index() pada transaksi pembelian......58 Gambar 6. 6 function create\_pembelian() pada controller Pembelian......61 Gambar 6. 7 function create\_pembelian\_action() pada controller Pembelian......62 Gambar 6. 8 function \_rules\_pembelian() pada controller Pembelian......63 Gambar 6. 9 function \_create\_itempembelian() pada controller Pembelian......64

Gambar 6. 10 function create_itempembelian_action() pada
controller Pembelian
Gambar 6. 11 function _rules_itempembelian() pada controller
Pembenan
Gambar 6. 12 function get_penerima dan get_pembuat pada
Karyawan_model
Gambar 6. 15 function insert_pembenan() pada Pembenan_model
Combox 6 14 function got lastid() node Dombolion model
Gambar 6, 15 function get_nashd() pada Feinbenan_model
Dembelian model 66
Gambar 6, 16 function get_itempembelian() pada
Pembelian model 66
Gambar 6 17 function insert itempembelian() pada
Pembelian model 67
Gambar 6 18 Menambahkan Data Transaksi Pembelian 67
Gambar 6, 19 function edit, pembelian() pada controller Pembelian
68
Gambar 6. 20 function edit pembelian action() pada controller
Pembelian
Gambar 6. 21 function update() pada Pembelian_model70
Gambar 6. 22 Mengubah Data Transaksi Pembelian
Gambar 6. 23 function delete() pada Controller Pembelian71
Gambar 6. 24 function deleteitem() pada controller Pembelian71
Gambar 6. 25 function delete() pada Pembelian_model72
Gambar 6. 26 function deleteitem() pada Pembelian_model72
Gambar 6. 27 class controller Penjualan
Gambar 6. 28 class model Penjualan_model74
Gambar 6. 29 function index pada controller Penjualan74
Gambar 6. 30 antarmuka daftar_penjualan75
Gambar 6. 31 menampilkan data transaksi penjualan
Gambar 6. 32 function create_penjualan pada controller Penjualan
Gambar 6. 33 function create_penjualan_action pada controller
Penjualan78
Gambar 6. 34 function _rules_penjualan pada controller Penjualan

Gambar 6. 35 function _create_itempenjualan pada controller
Penjualan79
Gambar 6. 36 function create_itempenjualan_action pada
controller Penjualan80
Gambar 6. 37 function _rules_itempenjualan pada controller
Penjualan
Gambar 6. 38 function insert pada Penjualan_model80
Gambar 6. 39 function get_lastid pada Penjualan_model81
Gambar 6. 40 function get_penjualan_byid pada Penjualan_model
Gambar 6. 41 function get_itempenjualanid pada Penjualan_model
Gambar 6 42 function insert itempeniualan pada Peniualan model
82
Gambar 6, 43 menambah data transaksi penjualan
Gambar 6. 44 function edit penjualan pada controller Penjualan 83
Gambar 6. 45 function edit penjualan action pada controller
Penjualan
Gambar 6, 46 function update pada Penjualan model
Gambar 6. 47 mengubah data transaksi penjualan
Gambar 6. 48 function delete pada controller Penjualan
Gambar 6. 49 function deleteitem pada controller Penjualan
Gambar 6. 50 function delete pada controller Penjualan
Gambar 6. 51 function deleteitem pada Penjualan model
Gambar 7. 1 class controller Laporan
Gambar 7. 2 class model Laporan model
Gambar 7. 3 function pembelian pada controller Laporan
Gambar 7. 4 function get_laporan_pembelian pada Laporan_model
Gember 7 5 function get tabun lanoren nombolien node
Laboran model
Cambor 7 6 antermula defter lanoren nombolion 04
Gambar 7. 7 manamnilkan lanaran namhalian
Cambor 7. 8 function ponjuolon peda controllor Lenoren
Gambar 7. 0 function get lanoran paniudan pada Lanoran model
Gambai 7. 7 Iuncuon get_iaporan_penjuaran pada Laporan_model

Gambar 7. 10 function get_tahun_laporan_penjualan pada	
Laporan_model	96
Gambar 7. 11 antarmuka daftar_laporan_penjualan	98
Gambar 7. 12 menampilkan laporan penjualan	99
Gambar 7. 13 function stockakhir pada controller Laporan	99
Gambar 7. 14 function get_stock_akhir pada Laporan_model	100
Gambar 7. 15 antarmuka daftar_stock_akhir	101
Gambar 7. 16 menampilkan laporan stock akhir	102
Gambar 8. 1 Controller Auth	104
Gambar 8. 2 function index() pada controller Auth	105
Gambar 8. 3 function proclogin()	106
Gambar 8. 4 function is_logged_in()	107
Gambar 8. 5 function loginfailed()	107
Gambar 8. 6 function proclogout()	108
Gambar 8. 7 menambahkan dan memanggil function	
is_logged_in() pada functionconstruct() di setiap controller	108
Gambar 9. 1 Halaman Login CPanel Website	109
Gambar 9. 2 Menu File Manager pada halaman CPanel website	110
Gambar 9. 3 Halaman File Manager	110
Gambar 9. 4 Menu MySQL Databases pada halaman CPanel	
Website	111
Gambar 9. 5 Menu phpMyAdmin pada halaman CPanel website	)
	111
Gambar 9. 6 konfigurasi database	111
Gambar 9. 7 Mengakses aplikasi pada website	112

#### Capaian Pembelajaran:

Mahasiswa mampu menjelaskan dan menerapkan konsep OOP pada pemrograman web

Program yang kompleks dapat dibangun dari berbagai package (paket). Sebuat paket terdiri dari sejumlah class, setiap class terdiri dari beberapa method atau constructor. Setiap class, minimal memiliki sebuah constructor, walaupun secara explicit tidak dibuat, namun secara implisit sesungguhnya tetap ada, hanya tidak ditampilkan.

# 1.1 Object Oriented Programming

*Object Oriented Programming (OOP)* adalah pandangan pemrograman yang berbasis kepada object. Semua fungsi dan data dalam OOP dikemas dalam objek-objek atau kelas-kelas. Berbeda dengan pemrograman terstruktur, setiap objek menerima pesan, memproses data, dan mengirim pesan ke objek lain, OOP dikembangkan untuk mengatasi masalah keterbatasan pada bahasa pemrograman tradisional karena konsep OOP adalah membagi permasalahan-permasalahan ke dalam objek-objek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabung menjadi satu kesatuan yang dapat disebut dengan objek.

# 1.2 Kelebihan OOP

Kelebihan tidak hanya berlaku untuk pemrograman web saja, tetapi di semua bahasa pemrograman yang menggunakan OOP memiliki kelebihan ini. Berikut beberapa kelebihan OOP antara lain:

- a. Syntax lebih terstruktur
- b. Efektif jika digunakan untuk mengembangkan aplikasi dengan skala yang besar
- c. Lebih menghemat waktu dalam mengembangkan aplikasi
- d. Lebih mudah digunakan karena fungsi yang sama dapat dipanggil atau digunakan berkali-kali

# 1.3 Object

Object adalah kumpulan dari variable dan fungsi yang digabung menjadi satu entitas. Entitas tersebut dapat merupakan variable biasa. Objek diciptakan melalui sebuah class atau dikenal dengan istilah instance of class. Objek memiliki 2 elemen utama, yaitu :

- Attibutes atau Properties, yaitu nilai-nilai yang tersimpan di dalam object dan secara langsung ataupun tidak langsung menentukan karakteristik dari object
- b. Method, yaitu aksi yang dijalankan atau dikerjakan oleh object

Setiap Object dibangun dari sekumpulan data (atribut) yang disebut "variabel" (untuk menjabarkan karakteristik khusus dari obyek) dan juga terdiri dari sekumpulan method (menjabarkan tingkah laku dari obyek) atau Obyek adalah = sebuah perangkat lunak yg berisi sekumpulan variabel dan method yang berhubungan. Obyek merupakan sebuah *instance* (keturunan) dari *class*. Variabel dan method diketahui sebagai variabel *instance* dan *method instance*.

# 1.4 Class

Class didefinisikan sebagai struktur data dari sebuah object. Class memiliki kelebihan dengan salah satu karakteristiknya yaitu reuseable, atau dapat digunakan kembali sehingga pengguna dapat menggunakan sebuah class berulang kali untuk membuat banyak object. Kemampuan objek sangat bergantung pada sarana yang telah disediakan oleh kelas. Kelas merupakan program Java yang akan dieksekusi.

Deklarasi class biasanya menambah kata kunci. Isi dari kelas (daerah antara dua tkita kurung kurawal) berisi semua kode yang disediakan untuk obyek yang diciptakan dari kelas, yaitu konstruktor untuk inisialisasi obyek baru, deklarasi field yang menetapkan keadaan kelas dan obyeknya dan method untuk mengimplementasikan lingkungan dari kelas dan obyeknya.

Secara umum, deklarasi kelas dapat termasuk komponen-komponen

- a. Modifier seperti public, private dan modifier yang lain yang akan kita bicarakan kemudian.
- b. Nama kelas, dengan diawali huruf besar sebagai kesepakatan.
- Nama dari induk kelasnya (superclass), jika ada, diawali dengan kata kunci extends. Sebuah kelas hanya boleh mempunyai satu induk
- Daftar interface (dipisahkan dengan titik koma) yang akan diimplementasikan dalam kelas, jika ada, diawali dengan kata kunci implements. Sebuah kelas boleh mengimplementasikanlebih dari satu interface
- e. Isi dari kelas yang diawali dan diakhiri dengan tanda kurung kurawal buka dan tutup { }

Bagian yang tidak kalah pentingnya adalah deklarasi variabel. Ada beberapa macam variabel yang ada antara lain :

 a. Variabel anggota dalam sebuah kelas – ini disebut fields. Fields ini terletak di luar method. Dan bisa diakses dai method dengan menggunakan referensi ke kelas yang memiliki field tersebut (dengan memperhatikan aturan akses modifier) b. Variabel dalam sebuah method atau blok kode – ini disebut variabel lokal. Variabel ini biasanya hanya digunakan selama method itu dikerjakan. Sehingga tidak perlu diakses dari luar method. Bahkan variabel yang ada di dalam blok bisa diakses dari blok itu saja.

c. Variabel dalam deklarasi method – ini disebut parameter. Deklarasi field terdiri dari 3 komponen, yaitu :

- a. Tidak ada atau ada modifier, seperti public atau private. Sebenarnya dengan tanpa menuliskan modifier, maka kita membuat deklarasi field tersebut sebagai default.
- b. Tipe field. Tipe ini bisa saja bukan hanya tipe sederhana tetapi tipe yang kompleks.
- c. Nama field. Dalam pembuatan nama, aturan penamaan harus diikuti. Dan sangat dianjurkan untuk menggunakan huruf kecil sebagai huruf pertama.

## 1.5 Method

Method merupakan bagian code yang dapat dipanggil oleh program utama atau method lain untuk melakukan suatu fungsi tertentu. Beberapa ciri dari method antara lain :

- a. Dapat mengembalikan nilai atau tidak sama sekali
- Dapat menerima beberapa parameter yang dibutuhkan atau tidak menerima parameter sama sekali. Parameter berguna sebagai nilai masukan yang akan diolah oleh fungsi
- c. Setelah method selesai dieksekusi, maka proses akan kembali ke method yang memanggil sebelumnya

#### 1.6 Latihan

Bandingkan perbedaan antara Object Oriented Programming dan Pemrograman Terstruktur.

#### Capaian Pembelajaran:

Mahasiswa mampu mengimplementasikan framework ke dalam pemrograman web

Bab ini akan membahan tentang CodeIgniter, keunggulan CodeIgniter, cara instalasi dan pengaturan CodeIgniter

#### 2.1 Pengenalan CodeIgniter

CodeIgniter merupakan sebuah *framework* berbasis *web* yang berguna untuk memudahkan para *web developer* membangun sebuah aplikasi berbasis *web. Framework* CodeIgniter merupakan suatu kerangka kerja yang terdiri dari *library* dan *tools* yang dipadukan untuk bahasa pemrograman PHP.

CodeIgniter dikembangkan oleh Rick Ellis, pendiri Ellis Lab, pada tahun 2006. Ellis Lab sebagai pengembang awal CodeIgniter telah menyerahkan hak kepemilikan CodeIgniter kepada *British Collumbia Institute of Technology* (BCIT) untuk pengembangan lebih lanjut sejak tahun 2014 hingga sekarang. CodeIgniter memiliki rancangan yang bersifat fleksibel. CodeIgniter mengizinkan pengembang web untuk menggunakannya secara parsial atau secara utuh, yang berarti CodeIgniter memberikan kebebasan kepada pengembang web untuk menuliskan kode-kode secara konvensional.

Untuk memudahkan pengembang web untuk membangun aplikasi dengan lebih cepat, CodeIgniter menggunakan arsitektur *Model-View-Controller (MVC)* yang memisahkan kode berdasarkan penanganan proses bisnis dan untuk keperluan tampilan. Dengan menggunakan arsitektur MVC, memungkinkan para pengembang web dapat membangun aplikasi secara bersamaan sehingga lebih fokus pada bagian kode masing-masing tanpa mengganggu bagian kode yang lain.

# 2.2 Keunggulan CodeIgniter

Adapun beberapa keunggulan dari CodeIgniter antara lain :

- 1. CodeIgniter bersifat open-source
- 2. CodeIgniter memiliki ukuran yang relatif lebih kecil daripada *framework* yang lain
- 3. CodeIgniter menggunakan arsitektur MVC sehingga masing-masing bagian tidak terlalu banyak kode program
- 4. CodeIgniter menyediakan dokumentasi yang lengkap untuk penggunaan *library* yang telah disertakan
- CodeIgniter mengizinkan penggunaan *library* yang tidak disertakan pada paketnya (contoh : Google Map API, FPDF, Facebook API, dan lain-lain)
- 6. CodeIgniter bersifat fleksibel sehingga mengizinkan pengembang untuk menggunakan *framework* secara parsial atau tidak sama sekali
- 7. CodeIgniter mendukung banyak RDBMS (*Relational Database Manegement System*)
- 8. CodeIgniter mendukung SEO, dan lain-lain

# 2.3 Instalasi CodeIgniter 3

Langkah pertama yaitu instalasi CodeIgniter 3. Adapun langkahlangkah instalasi adalah sebagai berikut :

- Unduh CodeIgniter pada halaman resmi (CodeIgniter v3.1.11) pada alamat <u>https://codeload.github.com/bcit-ci/CodeIgniter/zip/3.1.11</u>
- 2. Ekstrak *package* yang sudah diunduh
- Unggah folder dan file CodeIgniter pada *server* (pada folder htdocs jika di *localhost*, atau pada folder public\_html jika di *website*)

4. Ubah nama folder menjadi sipenjualan

#### 2.4 Pengaturan CodeIgniter

Sebelum menggunakan CodeIgniter, berikut beberapa pengaturan yang dilakukan :

1.Bukafileautoload.phppadasipenjualan/application/configmenggunakanTextEditor

2. Tambahkan code berikut kemudian Simpan file

```
$autoload['libraries'] = array('database','session', 'form_validation');
$autoload['helper'] = array('url','html','string', 'form');
```

```
Gambar 2. 1 Konfigurasi autoload pada CodeIgniter
```

- 3. Buka file **config.php** pada **sipenjualan/application/config** menggunakan *Text Editor*
- 4. Tambahkan code berikut kemudian Simpan file

\$config['base\_url'] = 'http://localhost/sipenjualan/';

Gambar 2. 2 Konfigurasi base URL pada CodeIgniter

- 5. Buka file **database.php** pada sipenjualan/application/config menggunakan *Text Editor*
- 6. Tambahkan code berikut kemudian Simpan file



Gambar 2. 3 Konfigurasi database pada CodeIgniter

- 7. Buka file **routes.php** pada **sipenjualan/application/config** menggunakan *Text Editor*
- 8. Tambahkan code berikut kemudian Simpan file

\$route['default\_controller'] = 'beranda';

Gambar 2. 4 Konfigurasi route pada CodeIgniter

9. Anda juga dapat mengunduh folder CodeIgniter yang sudah penulis siapkan beserta file Bootstrap yang sudah terpasang pada alamat :

https://drive.google.com/file/d/1GURhxjZDCvFB1TkaA1 EZLNzOIWlq2fcl/view?usp=sharing

10. Jalankan aplikasi pada *internet browser* dengan mengakses alamat <u>http://localhost/sipenjualan</u>



Gambar 2. 5 Tampilan awal SI Penjualan

#### 2.5 Latihan

Lakukan percobaan instalasi dan konfigurasi pada komputer anda masing-masing.

#### Capaian Pembelajaran:

Mahasiswa mampu menjelaskan dan mengimplementasikan konsep MVC (*Model View Controller*)

Bab ini membahas tentang konsep Model-View-Controller (MVC) dan cara kerja konsep MVC

#### 3.1 Model-View-Controller (MVC)

Seperti yang sudah dibahas pada Bab sebelumnya, CodeIgniter menggunakan arsitektur MVC. MVC adalah sebuah metode yang memisahkan aplikasi menjadi 3 bagian utama, yaitu Model, View dan Controller.

#### 3.1.1 Model

Model mewakili struktur data. Biasanya model berisi fungsi untuk membantu informasi dan operasi dalam database seperti *Create, Read, Update* dan *Delete* (CRUD)

#### 3.1.2 View

View merupakan informasi yang disajikan kepada pengguna. Tampilan view biasanya akan menjadi halaman web. Pada View CodeIgniter tidak selalu menampilkan halaman web, tetapi juga dapat menampilkan fragmen halaman seperti *header*, *sidebar*, dan *footer* 

#### 3.1.3 Controller

Controller berfungsi sebagai perantara antara *Model* dan *View*. Controller juga bersifat *bussiness logic* yang artinya controller akan merespon HTTP request yang datang dari user (melalui internet browser), dan dari request ini controller akan menentukan apa yang akan dilanjutkan selanjutnya.



#### 3.2 Cara kerja MVC

Gambar 3. 1 Cara kerja MVC

Controller menerima permintaan dari pengguna lalu berinteraksi dengan Model jika perlu kemudian mengembalikan hasilnya kepada *internet browser* dalam bentuk kode HTML yang diterjemahkan oleh internet browser menjadi format yang dapat dibaca dan ditampilkan ke pengguna.

#### 3.3 Latihan

Tuliskan konsep MVC dengan menggunakan bahasa anda sendiri

# Capaian Pembelajaran:

Mahasiswa mampu mengintegrasikan database dengan framework

Bab ini membahas langkah-langkah membuat database, membuat tabel, mengatur primary key dan foreign key, membuat view.

#### 4.1 Membuat database

Pada halaman PHPMyAdmin tekan tombol *New*, isikan nama database pada *Database Name*, kemudian tekan tombol *Create*. Jika berhasil maka database baru akan tercipta.

🕞 Create database   😡		
Database name	utf8mb4_general_ci	<ul> <li>✓ Create</li> </ul>

Gambar 4. 1 Membuat database

#### 4.2 Membuat Tabel

Berikut tabel – tabel yang diperlukan dalam membangun Sistem Informasi Penjualan :

#### 1. Tabel Barang

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	barangid 🄑	int(11)			No	None		AUTO_INCREMENT
2	nmbarang	varchar(50)	utf8mb4_general_ci		Yes	NULL		
3	harga	double			Yes	NULL		
4	stockawal	int(11)			Yes	NULL		



# 2. Tabel Karyawan

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	karyawanid 🔑	int(11)			No	None		AUTO_INCREMENT
2	nmkaryawan	varchar(50)	utf8mb4_general_ci		Yes	NULL		
3	hakakses	int(11)			Yes	NULL		
4	password	varchar(50)	utf8mb4_general_ci		Yes	NULL		



## 3. Tabel Kota

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	kotaid 🄑	int(11)			No	None		AUTO_INCREMENT
2	nmkota	varchar(50)	utf8mb4_general_ci		Yes	NULL		

Gambar 4. 4 Struktur tabel kota

#### 4. Tabel Pelanggan

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	pelangganid 🔑	int(11)			No	None		AUTO_INCREMENT
2	nmpelanggan	varchar(50)	utf8mb4_general_ci		Yes	NULL		
3	almtpelanggan	varchar(50)	utf8mb4_general_ci		Yes	NULL		
4	kotaid 🔊	int(11)			Yes	NULL		

Gambar 4. 5 Struktur tabel pelanggan

#### Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint (INNODB)				
		Database	Table	Column		
pelanggan_FK	kotaid 🗸	sineniualan	kota 🗙	kotaid		
ON DELETE RESTRICT V ON UPDATE CASCADE V	+ Add column	openjourun	- Total	inotana -		

#### Gambar 4. 6 Foreign Key pada tabel pelanggan

#### 5. Tabel Supplier

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	supplierid 🄑	int(11)			No	None		AUTO_INCREMENT
2	nmsupplier	varchar(50)	utf8mb4_general_ci		Yes	NULL		
3	almtsupplier	varchar(50)	utf8mb4_general_ci		Yes	NULL		
4	kotaid 🔊	int(11)			Yes	NULL		

## Gambar 4. 7 Struktur tabel supplier

Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint (INNODB)			
		Database	Table	Column	
Supplier_FK ON DELETE RESTRICT V ON UPDATE CASCADE V	kotaid  V + Add column	sipenjualan 🗸	kota 🗸	kotaid 🗸	

Gambar 4. 8 Foreign Key pada tabel supplier

# 6. Tabel Penjualan

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	penjualanid 🄌	int(11)			No	None		AUTO_INCREMENT
2	tanggal	timestamp			Yes	current_timestamp()		
3	pelangganid 🔊	int(11)			Yes	NULL		
4	pembuatid <i> </i>	int(11)			Yes	NULL		
5	penerima	varchar(50)	utf8mb4_general_ci		Yes	NULL		

Gambar 4. 9 Struktur tabel penjualan

Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint (INNODB)				
		Database	Table	Column		
penjualan_FK           ON DELETE         RESTRICT         •         ON UPDATE         CASCADE         •	pelangganid  + Add column	sipenjualan 🗸	pelanggan 🗸	pelangganid V		
penjualan_FK_1 ON DELETE RESTRICT V ON UPDATE CASCADE V	pembuatid  + Add column	sipenjualan 🗸	karyawan 🗸	karyawanid 🗸		

Gambar 4. 10 Foreign Key pada tabel penjualan

#### 7. Tabel ItemPenjualan

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	barangid 🔑	int(11)			No	None		
2	qty	int(11)			Yes	NULL		
3	harga	double			Yes	NULL		
4	penjualanid 🔑 🔊	int(11)			No	None		

# Gambar 4. 11 Struktur tabel itempenjualan

Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint	(INNODB)	
		Database	Table	Column
itemperjualan_FK ON DELETE RESTRICT   ON UPDATE CASCADE	barangid  + Add column	sipenjualan 🗸	barang ~	barangid 🗸
Itempenjualan_FK_1 ON DELETE CASCADE  V ON UPDATE CASCADE  V	penjualanid 🗸 🗸	sipenjualan 🗸	penjualan 🗸	penjualanid 🗸

Gambar 4. 12 Foreign Key pada tabel itempenjualan

# 8. Tabel Pembelian

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	pembelianid 🄑	int(11)			No	None		AUTO_INCREMENT
2	tanggal	timestamp			Yes	current_timestamp()		
3	supplierid 🔊	int(11)			Yes	NULL		
4	penerimaid 🔊	int(11)			Yes	NULL		
5	pengirim	varchar(50)	utf8mb4_general_ci		Yes	NULL		

Gambar 4. 13 Struktur tabel pembelian

Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint	(INNODB)	
		Database	Table	Column
pembelian_FK           ON DELETE         RESTRICT           V         ON UPDATE           CASCADE         V	penerimaid  + Add column	sipenjualan 🗸	karyawan 🗸	karyawanid V
pembelian_FK_1 ON DELETE RESTRICT V ON UPDATE CASCADE V	supplierid  + Add column	sipenjualan 🗸	supplier 🗸	supplierid 🗸

Gambar 4. 14 Foreign Key pada tabel pembelian

# 9. Tabel ItemPembelian

#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra
1	barangid <i> </i>	int(11)			No	None		
2	qty	int(11)			Yes	NULL		
3	harga	double			Yes	NULL		
4	pembelianid 🔑 🖉	int(11)			No	None		

Gambar 4. 15 Struktur tabel itempembelian

Dengan pengaturan Foreign Key sebagai berikut :

Constraint properties	Column 😡	Foreign key constraint (INNODB)				
		Database	Table	Column		
itompembelian_FK ON DELETE RESTRICT V ON UPDATE CASCADE V	barangid + Add column	sipenjualan 🗸	barang ~	barangid 🗸		
Itempembelian_FK_1 ON DELETE CASCADE  V ON UPDATE CASCADE  V	pembelianid  + Add column	sipenjualan 🗸	pembelian ~	pembelianid V		

Gambar 4. 16 Foreign Key pada tabel itempembelian

### 4.3 Membuat View

Berikut view - view yang diperlukan dalam membangun Sistem Informasi Penjualan :

1. View vbarang

```
SELECT
    `b`.`barangid` AS `barangid`,
    `b`.`nmbarang` AS `nmbarang`,
    `b`.`harga` AS `harga`,
    `b`.`stockawal` + IFNULL(
        (
        SELECT
            SUM(`i`.`qty`)
        FROM
            `sipenjualan`.`itempembelian` `i`
        WHERE
            `i`.`barangid` = `b`.`barangid`
    ),
    0
    ) - IFNULL(
        (
        SELECT
            SUM(`i2`.`qty`)
        FROM
            `sipenjualan`.`itempenjualan` `i2`
        WHERE
            `i2`.`barangid` = `b`.`barangid`
    ),
    0
    ) AS `stockakhir`
FROM
    `sipenjualan`.`barang` `b`
ORDER BY
    `b`.`nmbarang`
```

#### 2. View vdpembelian

```
SELECT
                      `i`.`barangid` AS `barangid`,
                    `i`.`qty` AS `qty`,
                    `i`.`harga` AS `harga`,
                    `i`.`pembelianid` AS `pembelianid`,
                    `b`.`nmbarang` AS `nmbarang`,
```

#### 3. View vditempembelian

#### 4. View vdpenjualan

```
SELECT
      p`.`penjualanid` AS `penjualanid`,
     CAST(`p`.`tanggal` AS DATE) AS `tanggal`,
       p`.`pelangganid` AS `pelangganid`,
      `p2`.`nmpelanggan` AS `nmpelanggan`,
      `p`.`pembuatid` AS `pembuatid`,
      `k`.`nmkaryawan` AS `pembuat`,
      `p`.`penerima` AS `penerima`,
     SUM(`i`.`qty`) AS `jmlbarang`,
SUM(`i`.`qty` * `i`.`harga`) * 110 / 100 AS `jmlnota`
FROM
      `sipenjualan`.`penjualan` `p`
JOIN `sipenjualan`.`pelanggan` `p2`
JOIN `sipenjualan`.`karyawan` `k`
JOIN `sipenjualan`.`itempenjualan` `i`
WHERE
`p`.`pembuatid` = `k`.`karyawanid` AND `p`.`pelangganid` =
`p2`.`pelangganid` AND `i`.`penjualanid` = `p`.`penjualanid`
GROUP BY
      p`.`penjualanid`
```

#### 5. View vkota

```
ORDER BY
`k`.`nmkota`
```

#### 6. View vpelanggan

```
SELECT
    `p`.`pelangganid` AS `pelangganid`,
    `p`.`nmpelanggan` AS `nmpelanggan`,
    `p`.`almtpelanggan` AS `almtpelanggan`,
    `k`.`nmkota` AS `kotapelanggan`
FROM
    `sipenjualan`.`pelanggan` `p`
JOIN `sipenjualan`.`kota` `k`
WHERE
    `p`.`kotaid` = `k`.`kotaid`
ORDER BY
    `p`.`nmpelanggan`
```

#### 7. View vpembuat

#### 8. View vpenerima

#### 9. View vrpembelian

```
SELECT
   YEAR(`p`.`tanggal`) AS `tahun`,
   `b`.`nmbarang` AS `nmbarang`,
   IFNULL(
        (
        SELECT
        SUM(`i2`.`qty`)
   FROM
        (
```

```
`sipenjualan`.`itempembelian` `i2`
JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
               `tahun`
) AND MONTH(`p2`.`tanggal`) = 1 AND `i2`.`pembelianid`
= `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    Ω
     ) AS `jan`,
    IFNULL(
         (
         SELECT
              SUM(`i2`.`qtv`)
         FROM
               (
                   `sipenjualan`.`itempembelian` `i2`
              JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
               `tahun`
  ) AND MONTH(`p2`.`tanggal`) = 2 AND `i2`.`pembelianid`
`p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `feb`,
    IFNULL(
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
               (
                   `sipenjualan`.`itempembelian` `i2`
              JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(p2`.`tanggal`) = (
         SELECT
               `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 3 AND `i2`.`pembelianid`
  `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
=
    ),
    0
     ) AS `mar`,
    IFNULL (
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
               (
                   `sipenjualan`.`itempembelian` `i2`
              JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
```

YEAR(`p2`.`tanggal`) =(

```
SELECT
               `tahun`
  ) AND MONTH(`p2`.`tanggal`) = 4 AND `i2`.`pembelianid`
`p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `apr`,
    IFNULL(
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
              (
                   `sipenjualan`.`itempembelian` `i2`
              JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 5 AND `i2`.`pembelianid`
= `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `mei`,
    IFNULL (
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
              (
                   `sipenjualan`.`itempembelian` `i2`
              JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
  ) AND MONTH(`p2`.`tanggal`) = 6 AND `i2`.`pembelianid`
`p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `jun`,
    IFNULL (
          (
         SELECT
             SUM(`i2`.`qty`)
         FROM
              (
              `sipenjualan`.`itempembelian` `i2`
JOIN `sipenjualan`.`pembelian` `p2`
              )
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 7 AND `i2`.`pembelianid`
  `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
=
```

),

```
0
    ) AS `jul`,
    IFNULL(
        (
        SELECT
            SUM(`i2`.`qty`)
       FROM
            (
                `sipenjualan`.`itempembelian` `i2`
            JOIN `sipenjualan`.`pembelian` `p2`
            )
        WHERE
            YEAR(`p2`.`tanggal`) =(
        SELECT
            `tahun`
       ) AND MONTH(`p2`.`tanggal`) = 8 AND `i2`.`pembelianid`
  `p2`.`pembelianid` AND `i2`.`barangid` = `b`.`barangid`
   ),
   0
   ) AS `ags`,
   IFNULL(
       (
        SELECT
           SUM(`i2`.`qtv`)
        FROM
            (
                `sipenjualan`.`itempembelian` `i2`
            JOIN `sipenjualan`.`pembelian` `p2`
            )
        WHERE
            YEAR(`p2`.`tanggal`) =(
        SELECT
            `tahun`
        ) AND MONTH(`p2`.`tanggal`) = 9 AND `i2`.`pembelianid`
  `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
_
   ),
   0
   ) AS `sep`,
   IFNULL(
        (
        SELECT
           SUM(`i2`.`qty`)
        FROM
                `sipenjualan`.`itempembelian` `i2`
            JOIN `sipenjualan`.`pembelian` `p2`
            )
        WHERE
            YEAR(`p2`.`tanggal`) =(
        SELECT
             `tahun`
        ) AND MONTH(`p2`.`tanggal`) = 10 AND `i2`.`pembelianid`
 `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
   ),
   0
   ) AS `okt`,
   IFNULL(
        (
```

SELECT

```
SUM(`i2`.`qty`)
         FROM
              (
             `````sipenjualan`.`itempembelian```i2`
JOIN `sipenjualan`.`pembelian```p2`
             )
         WHERE
             YEAR(p2`.`tanggal`) = (
         SELECT
              `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 11 AND `i2`.`pembelianid`
  `p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `nop`,
    IFNULL(
         (
         SELECT
             SUM(`i2`.`qty`)
         FROM
              (
                   `sipenjualan`.`itempembelian` `i2`
             JOIN `sipenjualan`.`pembelian` `p2`
             )
         WHERE
             YEAR(`p2`.`tanggal`) =(
         SELECT
             `tahun`
 ) AND MONTH(`p2`.`tanggal`) = 12 AND `i2`.`pembelianid`
`p2`.`pembelianid` AND `i2`.`barangid` = `i`.`barangid`
=
    ),
    0
    ) AS `des`
FROM
    (
         (
             `sipenjualan`.`pembelian` `p`
         JOIN `sipenjualan`.`itempembelian` `i`
         ON
              (`i`.`pembelianid` = `p`.`pembelianid`)
         )
    JOIN `sipenjualan`.`barang` `b`
    ON
         (`b`.`barangid` = `i`.`barangid`)
    )
GROUP BY
    (
SELECT
     `tahun`
),
`b`.`baranqid`
ORDER BY
    (
SELECT
    `tahun`
)
DESC
    `b`.`barangid`
```

#### 10. View vrpenjualan

```
SELECT
    YEAR(`p`.`tanggal`) AS `tahun`,
     `b`.`nmbarang` AS `nmbarang`,
     IFNULL(
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
              `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
) AND MONTH(`p2`.`tanggal`) = 1 AND `i2`.`penjualanid`
= `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `jan`,
    TENULL (
         (
         SELECT
             SUM(`i2`.`qty`)
         FROM
              `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
              YEAR(`p2`.`tanggal`) =(
         SELECT
               `tahun`
) AND MONTH(`p2`.`tanggal`) = 2 AND `i2`.`penjualanid`
= `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `feb`,
    IFNULL (
         (
         SELECT
              SUM(`i2`.`qty`)
         FROM
              `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
             YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 3 AND `i2`.`penjualanid`
  `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    Ω
    ) AS `mar`,
    IFNULL (
         (
         SELECT
```

SUM(`i2`.`qty`)

```
FROM
             `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
             YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
) AND MONTH(`p2`.`tanggal`) = 4 AND `i2`.`penjualanid`
= `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    Ω
    ) AS `apr`,
    IFNULL(
         (
         SELECT
             SUM(`i2`.`qtv`)
         FROM
             `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
             YEAR(p2`.`tanggal`) = (
         SELECT
             `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 5 AND `i2`.`penjualanid`
   `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `mei`,
    IFNULL(
        (
         SELECT
             SUM(`i2`.`qty`)
         FROM
             `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
             YEAR(`p2`.`tanggal`) =(
         SELECT
              `tahun`
         ) AND MONTH(`p2`.`tanggal`) = 6 AND `i2`.`penjualanid`
  `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `jun`,
    IFNULL (
        (
         SELECT
             SUM(`i2`.`qty`)
         FROM
             `sipenjualan`.`itempenjualan` `i2`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
             YEAR(`p2`.`tanggal`) =(
         SELECT
             `tahun`
  ) AND MONTH(`p2`.`tanggal`) = 7 AND `i2`.`penjualanid`
`p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
```

0
```
(
        SELECT
            SUM(`i2`.`qty`)
        FROM
            `sipenjualan`.`itempenjualan` `i2`
        JOIN `sipenjualan`.`penjualan` `p2`
        WHERE
            YEAR(p2`.`tanggal`) = (
        SELECT
            `tahun`
       ) AND MONTH(`p2`.`tanggal`) = 10 AND `i2`.`penjualanid`
= `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
    0
    ) AS `okt`,
    IFNULL(
        (
        SELECT
            SUM(`i2`.`qty`)
        FROM
            `sipenjualan`.`itempenjualan` `i2`
        JOIN `sipenjualan`.`penjualan` `p2`
        WHERE
            YEAR(`p2`.`tanggal`) =(
        SELECT
```

```
) AS `jul`,
 IFNULL(
      (
      SELECT
         SUM(`i2`.`qty`)
      FROM
          `sipenjualan`.`itempenjualan` `i2`
      JOIN `sipenjualan`.`penjualan` `p2`
      WHERE
         YEAR(`p2`.`tanggal`) =(
      SELECT
          `tahun`
      ) AND MONTH(`p2`.`tanggal`) = 8 AND `i2`.`penjualanid`
`p2`.`penjualanid` AND `i2`.`barangid` = `b`.`barangid`
 ),
 0
 ) AS `ags`,
 IFNULL(
     (
      SELECT
          SUM(`i2`.`qty`)
```

`sipenjualan`.`itempenjualan` `i2`

= `p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`

) AND MONTH(`p2`.`tanggal`) = 9 AND `i2`.`penjualanid`

JOIN `sipenjualan`.`penjualan` `p2`

YEAR(`p2`.`tanggal`) =(

FROM

WHERE

SELECT

) AS `sep`, IFNULL(

),

`tahun`

```
`tahun`
  ) AND MONTH(`p2`.`tanggal`) = 11 AND `i2`.`penjualanid`
`p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
    ),
     0
     ) AS `nop`,
     IFNULL(
          (
          SELECT
               SUM(`i2`.`qty`)
          FROM
               `sipenjualan`.`itempenjualan` `i2`
          JOIN `sipenjualan`.`penjualan` `p2`
          WHERE
               YEAR(`p2`.`tanggal`) =(
          SELECT
               `tahun`
  ) AND MONTH(`p2`.`tanggal`) = 12 AND `i2`.`penjualanid`
`p2`.`penjualanid` AND `i2`.`barangid` = `i`.`barangid`
=
    ),
     0
     ) AS `des`
FROM
     (
          (
          `sipenjualan`.`penjualan` `p`
JOIN `sipenjualan`.`itempenjualan` `i`
          ON
               (`i`.`penjualanid` = `p`.`penjualanid`)
          )
     JOIN `sipenjualan`.`barang` `b`
     ON
          (`b`.`barangid` = `i`.`barangid`)
     )
GROUP BY
     (
SELECT
     `tahun`
),
`b`.`barangid`
ORDER BY
    (
SELECT
     `tahun`
)
DESC
     `b`.`barangid`
```

11. View vsupplier

```
SELECT
   `s`.`supplierid` AS `supplierid`,
   `s`.`nmsupplier` AS `nmsupplier`,
   `s`.`almtsupplier` AS `almtsupplier`,
   `k`.`nmkota` AS `kotasupplier`
FROM
```

```
`sipenjualan`.`supplier` `s`
JOIN `sipenjualan`.`kota` `k`
WHERE
`s`.`kotaid` = `k`.`kotaid`
ORDER BY
`s`.`nmsupplier`
```

#### 12. View vstockakhir

```
SELECT
    `b`.`barangid` AS `barangid`,
    `b`.`nmbarang` AS `nmbarang`,
    `b`.`harga` AS `harga`,
    `b`.`stockawal` + IFNULL(
        (
        SELECT
            SUM(`i`.`qty`)
        FROM
             `sipenjualan`.`itempembelian` `i`
         JOIN `sipenjualan`.`pembelian` `p`
         WHERE
             `i`.`pembelianid`
   `p`.`pembelianid`
                                   =
   AND
`i`.`barangid` = `b`.`barangid` AND CAST(`p`.`tanggal` AS DATE)
<= `p`.`tanggal`
    ),
    0
    ) - IFNULL(
         (
         SELECT
             SUM(`ip`.`qty`)
         FROM
             `sipenjualan`.`itempenjualan` `ip`
         JOIN `sipenjualan`.`penjualan` `p2`
         WHERE
                                   = `p2`.`penjualanid`
             `ip`.`penjualanid`
   AND
`ip`.`barangid` = `b`.`barangid`
   ),
    0
    ) AS `stockakhir`,
    IF(
         `b`.`stockawal` + IFNULL(
             (
             SELECT
                 SUM(`ip2`.`qty`)
             FROM
                 `sipenjualan`.`itempembelian` `ip2`
             JOIN `sipenjualan`.`pembelian` `p3`
`ip2`.`pembelianid` = `p3`.`pembelianid` AND
`ip2`.`barangid` = `b`.`barangid` AND CAST(`p3`.`tanggal` AS
DATE) <= `p3`.`tanggal`
             WHERE
        ),
        0
         ) - IFNULL(
             (
             SELECT
```

SUM(`ip3`.`qty`)

```
`p`.`pembelianid` AS `pembelianid`,
CAST(`p`.`tanggal` AS DATE) AS `tanggal`,
     `p`.`supplierid` AS `supplierid`,
`s`.`nmsupplier` AS `nmsupplier`,
`s`.`almtsupplier` AS `almtsupplier`,
     `k`.`nmkota` AS `kotasupplier`,
     `k2`.`nmkaryawan` AS `penerima`,
     `p`.`pengirim` AS `pengirim`,
     `i`.`qty` AS `qty`,
     `i`.`harga` AS `harga`,
     `i`.`qty * `i`.`harga` AS `subtotal`,
    SUM(`i`.`qty` * `i`.`harga`) AS `total`,
     (
     SELECT
         SUM(`i`.`qty` * `i`.`harga`) * 10 / 100
) AS `ppn`,
(
     SELECT
         SUM(`i`.`qty` * `i`.`harga`) * 110 / 100
) AS `grandtotal`,
SUM(`i`.`qty`) AS `jmlbarang`
FROM
     (
         (
              (
                   (
                         (
                         `sipenjualan`.`pembelian` `p`
JOIN `sipenjualan`.`supplier` `s`
                         )
                    JOIN `sipenjualan`.`kota` `k`
                    )
               JOIN `sipenjualan`.`karyawan` `k2`
          JOIN `sipenjualan`.`itempembelian` `i`
          )
     JOIN `sipenjualan`.`barang` `b`
```

13. View vnotapembelian

SELECT

```
)
WHERE
`p`.`supplierid` = `s`.`supplierid` AND `s`.`kotaid` =
`k`.`kotaid` AND `p`.`penerimaid` = `k2`.`karyawanid` AND
`i`.`pembelianid` = `p`.`pembelianid` AND `i`.`barangid` =
`b`.`barangid`
GROUP BY
`p`.`pembelianid`
```

#### 14. View vnotapenjualan

`p`.`penjualanid`

```
SELECT
     `p`.`penjualanid` AS `penjualanid`,
     CAST(`p`.`tanggal` AS DATE) AS `tanggal`,
     `p`.`pelangganid` AS `pelangganid`,
     `s`.`nmpelanggan` AS `nmpelanggan`,
     `s`.`almtpelanggan` AS `almtpelanggan`,
`k`.`nmkota` AS `kotasupplier`,
     `k2`.`nmkaryawan` AS `pembuat`,
    k2 . hmkaryawan AS penduat,
`p`.`penerima`AS `penerima`,
`i`.`qty`AS `qty`,
`i`.`harga`AS `harga`,
`i`.`qty` * `i`.`harga`AS `subtotal`,
SUM(`i`.`qty` * `i`.`harga`) AS `total`,
     SELECT
         SUM(`i`.`qty` * `i`.`harga`) * 10 / 100
) AS `ppn`,
(
     SELECT
         SUM(`i`.`qty` * `i`.`harga`) * 110 / 100
) AS `grandtotal`,
SUM(`i`.`qty`) AS `jmlbarang`
FROM
     (
          (
               (
                     (
                          (
                               `sipenjualan`.`penjualan` `p`
                          JOIN `sipenjualan`.`pelanggan` `s`
                     JOIN `sipenjualan`.`kota` `k`
               JOIN `sipenjualan`.`karyawan` `k2`
               )
          JOIN `sipenjualan`.`itempenjualan` `i`
     JOIN `sipenjualan`.`barang` `b`
    )
WHERE
     `p`.`pelangganid` = `s`.`pelangganid` AND `s`.`kotaid` =
`k`. kotaid` AND `p`.`pembuatid` = `k2`.`karyawanid` AND
`i`.`penjualanid` = `p`.`penjualanid` AND `i`.`barangid` =
`b`.`barangid`
GROUP BY
```

## 4.4 Latihan

Buatlah database sesuai dengan langkah-langkah yang sudah dituliskan sebelumnya.

# **BAB 5** Implementasi CRUD Pada Data Master

# Capaian Pembelajaran:

Mahasiswa mampu mengimplementasikan CRUD (*Create Read Update Delete*) pada pemrograman web

Bab ini membahas bagaimana membuat aplikasi menggunakan framework CodeIgniter 3 dan Bootstraps sebagai antarmuka pada menu Data Master.

# 5.1 Menampilkan Data Master Kota

Berikut langkah-langkah dalam menampilkan data master kota menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Kota.php** pada folder **sipenjualan/application/controllers/**
- 2. Tambahkan fungsi berikut pada *Controller* **Kota**, letakkan di atas fungsi *index()*



Gambar 5. 1 function \_\_construct() pada controller Kota

Keterangan :

Fungsi tersebut digunakan untuk memuat file **Kota\_model.php** di dalam folder model

 Buatlah sebuah file *model* baru di dalam folder sipenjualan/application/model/, beri nama Kota\_model.php

```
class Kota_model extends CI_Model {
    function __construct() {
        parent::__construct();
    }
    function get_kota_all() {
        ssql = "SELECT kota.* FROM kota";
        return $this->db->query($sql)->result();
    }
```

Gambar 5. 2 class Model Kota Model

Fungsi yang ada di dalam kotak merah berfungsi untuk memanggil *query* untuk mengambil semua data didalam tabel **kota** 

4. Tambahkan code program berikut pada file **Kota.php** yang ada di dalam folder *Controllers* 

Gambar 5. 3 function index() pada controller Kota

- a. Memanggil fungsi get\_kota\_all() pada file
   Kota\_model.php yang ada di dalam folder *model*,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$kota\_data
- b. Array **'kota\_data'** pada variable **\$data** dimasukkan nilai yang sudah ada dari variable **\$kota\_data**
- c. Semua nilai pada array \$data disertakan pada saat pemanggilan daftarkota.php pada folder views/content/kota
- 5. Tambahkan kode program pada file **daftarkota.php** yang ada di dalam folder **views/content/kota**



Gambar 5. 4 Antarmuka daftar\_kota

- a. Membuat tombol di tampilan program untuk menambahkan data
- Merupakan perulangan sesuai dengan jumlah data yang ada di dalam variable **\$kota\_data**. Isi nilai pada

variable **\$kota** disamakan dengan isi nilai pada variable **\$kota\_data** 

- Kolom pertama diisi dengan nilai yang ada dalam variable \$no. Dimulai dari angka 1 dan terus ditambahkan 1 sesuai dengan jumlah perulangan
- Kolom kedua, diisi dengan isi variable **\$kota** dalam field **kotaid** (sesuaikan dengan nama field di database masing masing)
- Kolom ketiga, diisi dengan isi variable **\$kota** dalam field **namakota** (sesuaikan dengan nama field di database masing masing)
- Kolom keempat, membuat tampilan tombol untuk mengubah dan menghapus data

SI Penjualan	=			
<ul> <li>Online</li> </ul>	Data Kota			
NAVIGASI UTAMA	🕈 Daftar Kota	+ Tambah		
🍘 Beranda	No.	Kota ID	Nama Kota	Aksi
DATA MASTER	1	1	Banjarmasin	Edit Hapus
📥 Data Karyawan	2	3	Barabai	Edit Hapus
🚚 Data Supplier				
矕 Data Pelanggan				
📍 Data Kota				
🗅 Data Barang				
DATA TRANSAKSI				
🖾 Data Pembelian				
💊 Data Penjualan				
LAPORAN				
🕍 Laporan Pembelian				
🛃 Laporan Penjualan				
E Stock Akhir	2021 - Administrasi Bionis Politeknik Negeri Banjarmasin. All rights reserved.			

#### 6. Simpan dan jalankan program

Gambar 5. 5 Menampilkan Data Master Kota

## 5.2 Menambahkan Data Master Kota

Berikut langkah-langkah dalam menambahkan data master kota menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Kota

```
public function create() {
    $data = array(
        'action' => site_url('Kota/create_action'),
        'kotaid' => set_value('kotaid'),
        'nmkota' => set_value('nmkota')
    );
    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('content/kota/form_kota', $data);
    $this->load->view('template/footer');
}
```

Gambar 5. 6 function create() pada controller Kota

- a. Action yang ada pada form nanti akan diproses oleh fungsi **create\_action** pada Controller **Kota**
- Memasukkan nilai ke dalam array 'kotaid' pada variable \$data yang berasal dari form yang memiliki name kotaid.
- c. Memasukkan nilai ke dalam array 'nmkota' pada variable \$data yang berasal dari form yang memiliki name **nmkota**.
- d. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_kota.php pada folder views/content/kota
- 2. Tambahkan fungsi berikut pada Controller Kota



- a. Memanggil fungsi \_rules() yang ada pada Controller
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi create()
- c. Mengambil nilai pada form dengan name 'nmkota' dan dimasukkan ke array pada variable \$data
- Memanggil fungsi insert(\$data) yang ada pada model Kota\_model
- e. Memanggil Controller Kota

#### 3. Tambahkan fungsi berikut pada Controller Kota

```
public function _rules() {
    $this->form_validation->set_rules('nmkota', 'Nama Kota', 'required');
    $this->form_validation->set_error_delimiters('<span class="label label-danger">', '</span>');
}
```

Gambar 5. 8 function \_rules() pada controller Kota

- a. Form yang memiliki name nmkota harus diisi (*required*)
- b. Memunculkan pesan error jika tidak memenuhi syarat validasi

Fungsi yang diawali dengan *underscore*, artinya hanya bisa dipanggil melalui proses pemanggilan pada kode program, tidak bisa dipanggil melalui *address bar*.

4. Tambahkan fungsi berikut pada model Kota\_model

```
public function insert($data) {
    $this->db->insert('kota', $data);
}
```

Gambar 5. 9 function insert() pada Kota\_model

Keterangan :

Fungsi ini juga berfungsi untuk memasukkan nilai dalam variable **\$data** ke dalam tabel **kota**.

5. Buat file baru dalam folder views/content/kota dengan nama form\_kota.php

```
<div class="content-wrapper">
 <section class="content-header">
   <h1>Data Kota</h1>
   <a href="<?php
   ?>"><i
                        echo base url();
class="fa fa-dashboard"></i> Beranda</a>
     Kota
     Form Kota
   </section>
 <section class="content">
   <div class="row">
     <section
              class="col-lg-12 connectedSortable">
<div class="box box-info">
        <div class="box-header">
          <i class="fa fa-map-pin"></i>
          <h3 class="box-title">Form Kota</h3>
        </div>
        <div class="box-body">
          <?php echo form open($action) ?>
          <div class="form-group">
            <div class="row">
```

```
<div class="col-md-2 col-sm-4 col-xs-4">
<?php echo form label('Nama Kota'); ?>
                </div>
              <div class="col-md-10 col-sm-8 col-xs-8">
                 <?php
                          echo
                                    form input('nmkota',
$nmkota, 'placeholder="Masukkan Nama Kota" class="form-
control"'); ?>
                </div>
              </div>
            \langle /div \rangle
            <div class="form-group">
              <div class="row">
               <div class="col-md-2 col-sm-4 col-xs-4">
                  </div>
              <div class="col-md-10 col-sm-8 col-xs-8">
           <?php echo form hidden('kotaid', $kotaid); ?>
                                   form submit('submit',
              <?php
                         echo
'Simpan', 'class="btn btn-primary"'); ?>
                </div>
              </div>
            </div>
          </div>
          <div class="box-footer">
          </div>
        </div>
      </section>
    </div>
  </section>
</div>
```

#### 6. Simpan dan jalankan program

SI Penjualan			🙆 Nama Karyawan
• Online	Data Kota		🏟 Beranda 🖂 Kota 🌾 Form Kota
	🕈 Form Kota		
🍘 Beranda	Nama Kota	Masukkan Nama Kota	
		Simpan	
🏝 Data Karyawan			
🚚 Data Supplier			
📽 Data Pelanggan			
Stock Akhir	2021 - Administrasi Bisnis Politek	sik Negeri Banjarmasin. All rights reserved.	Version 1.0

Gambar 5. 10 Menambah Data Master Kota

## 5.3 Mengubah Data Master Kota

Berikut langkah-langkah dalam mengubah data master kota menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Kota



- a. Memanggil fungsi get\_byid() yang ada pada model
   Kota\_model dan memasukkan hasilnya ke dalam variable \$kota
- b. Jika variable \$kota ada nilainya maka, memasukkan nilai ke dalam array 'kotaid' pada variable \$data pada form yang memiliki name kotaid yang berasal dari *field* kotaid.
- c. Jika variable \$kota ada nilainya maka, memasukkan nilai ke dalam array 'nmkota' pada variable \$data pada form yang memiliki name **nmkota** yang berasal dari *field* kotaid.

- d. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_kota.php pada folder views/content/kota
- e. Jika variable **\$kota** tidak ada nilainya, maka memanggil *controller* **Kota**
- 2. Tambahkan fungsi berikut pada Controller Kota



- a. Memanggil fungsi \_rules() yang ada pada Controller
- b. Mengambil nilai dari form yang memiliki name **kotaid** kemudian dimasukkan ke dalam variable **\$kotaid**
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi edit(\$kotaid)
- d. Mengambil nilai pada form dengan name 'kotaid' dan dimasukkan ke array pada variable \$data
- e. Mengambil nilai pada form dengan name 'nmkota' dan dimasukkan ke array pada variable \$data

- f. Memanggil fungsi update(\$kotaid, \$data) yang ada pada *model* Kota\_model
- g. Memanggil Controller Kota
- 3. Tambahkan fungsi berikut pada Model Kota\_model

```
function get_byid($kotaid) {
    $sql = "SELECT kota.* FROM kota
    WHERE kota.kotaid = '" . $kotaid . "'";
    return $this->db->query($sql)->row();
}
```

Gambar 5. 13 function get\_byid() pada Kota\_model

Keterangan :

Fungsi memanggil *query* untuk mengambil salah satu data didalam tabel **kota** berdasarkan **kotaid** yang ingin diubah

4. Tambahkan fungsi berikut pada Model Kota\_model

```
public function update($kotaid, $data) {
    $this->db->where('kotaid', $kotaid);
    $this->db->update('kota', $data);
}
```

Gambar 5. 14 function update() pada Kota\_model

Keterangan :

Fungsi ini berfungsi untuk mengubah nilai dalam variable \$data ke dalam tabel **kota** dimana kuncinya adalah **kotaid**.

5. Simpan dan jalankan program

SI Penjualan	=		💩 Nama Karyawan
<ul> <li>Online</li> </ul>	Data Kota		🏟 Beranda > Kota > Form Kota
NAVIGASI UTAMA	🕈 Form Kota		
🙆 Beranda	Nama Kota	Banjarmasin	
DATA MASTER		Simpan	
📥 Data Karyawan			
🚚 Data Supplier			
🖉 Data Pelanggan			
📍 Data Kota			
🗅 Data Barang			
DATA TRANSAKSI			
🖸 Data Pembelian			
🗣 Data Penjualan			
LAPORAN			
🕍 Laporan Pembelian			
🛃 Laporan Penjualan			
Stock Akhir	2021 - Administrasi Bisnis Politekr	ik Negeri Banjarmasin. All rights reserved.	Version 1.0

Gambar 5. 15 Mengubah Data Master Kota

# 5.4 Menghapus Data Master Kota

Berikut langkah-langkah dalam menghapus data master kota menggunakan CodeIgniter 3 :

## 1. Tambahkan fungsi berikut pada Controller Kota

```
public function delete($kotaid) {
    $kota = $this->Kota_model->get_byid($kotaid);
    if($kota) {
        $this->Kota_model->delete($kotaid);
        redirect(site_url('Kota'));
    } else {
        redirect(site_url('Kota'));
    }
}
```

## Gambar 5. 16 function delete() pada controller Kota

Keterangan :

a. Memanggil fungsi get\_byid() yang ada pada model
 Kota\_model dan memasukkan hasilnya ke dalam variable \$kota

- b. Jika variable \$kota ada nilainya maka memanggil fungsi delete dari **Kota\_model**.
- c. Memanggil controller Kota
- d. Jika variable **\$kota** tidak ada nilainya, maka memanggil *controller* **Kota**
- 2. Tambahkan fungsi berikut pada Model Kota\_model

```
public function delete($kotaid) {
    $this->db->where('kotaid', $kotaid);
    $this->db->delete('kota');
}
```

Gambar 5. 17 function delete pada Kota\_model

Fungsi ini berfungsi untuk menghapus baris di dalam tabel **kota** dimana kuncinya adalah **kotaid** 

3. Simpan dan jalankan program

## 5.5 Menampilkan Data Master Supplier

Berikut langkah-langkah dalam menampilkan data master supplier menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Supplier.php** pada folder **sipenjualan/application/controllers**/
- 2. Tambahkan fungsi berikut pada *Controller* **Supplier**, letakkan di atas fungsi *index()*



Gambar 5. 18 function index() pada controller Supplier

Fungsi tersebut digunakan untuk memuat file **Supplier\_model dan Kota\_model.php** di dalam folder model

 Buatlah sebuah file *model* baru di dalam folder sipenjualan/application/model/, beri nama Supplier\_model.php

```
class Supplier_model extends CI_Model {
    function __construct() {
        parent::__construct();
    }
    function get_supplier_all() {
        $sql = "SELECT vsupplier.* FROM vsupplier";
        return $this->db->query($sql)->result();
    }
```

Gambar 5. 19 class Model pada Supplier\_model

Keterangan :

Fungsi yang berfungsi untuk memanggil *query* untuk mengambil semua data didalam view **vsupplier** 

4. Tambahkan code program berikut pada file **Supplier.php** yang ada di dalam folder *Controllers* 

```
public function index() {
    $supplier_data = $this->Supplier_model->get_supplier_all();
    $data = array(
        'supplier_data' => $supplier_data
    );
    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('content/supplier/daftar_supplier', $data);
    $this->load->view('template/footer');
}
```

Gambar 5. 20 function index() pada controller Supplier

- a. Memanggil fungsi get\_supplier\_all() pada file
   Supplier\_model.php yang ada di dalam folder *model*,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$supplier\_data
- b. Array 'supplier\_data' pada variable \$data dimasukkan nilai yang sudah ada dari variable \$supplier\_data
- c. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_supplier.php pada folder views/content/supplier
- 5. Tambahkan kode program pada file **daftar\_supplier.php** yang ada di dalam folder **views/content/supplier**

```
<h3 class="box-title">Daftar Supplier</h3>
<div class="box-tools pull-right">
<div class="box tools pull-right">
</div class="box tools pull-right"</div class="box tools pull-right">
</div class="box tools pull-right">
</div class="box tools pull-right"</div class="box tools pull-right">
</div class="box tools pull-right"</div class="box tools
```

```
<div class="box-body";</pre>
 <thead>
   No.
    Supplier ID
    Nama Supplier
    Alamat Supplier
    Kota Supplier
    Aksi
   </thead>
  <?php $no = 1;</pre>
   foreach($supplier_data as $supplier) { ?>
    <?php echo $no++; ?>
     <?php echo $supplier->supplierid; ?>
     <?php echo $supplier->nmsupplier; ?>
     <?php echo $supplier->almtsupplier; ?>
     <?php echo $supplier->kotasupplier; ?>
      <a href="<?php echo site_url('Supplier/edit/' . $supplier->supplierid); ?>"
      class="btn btn-sm btn-warning">Edit</a>
      <a href="<?php echo site_url('Supplier/delete/' . $supplier->supplierid); ?>"
       class="btn btn-sm btn-danger"
       onclick="return confirm('Apakah Anda Yakin ?')">Hapus</a>
     <?php } ?>
  -/div>
```

Gambar 5. 21 antarmuka daftar\_supplier

- a. Membuat tombol di tampilan program untuk menambahkan data
- Merupakan perulangan sesuai dengan jumlah data yang ada di dalam variable \$supplier\_data. Isi nilai pada variable \$supplier disamakan dengan isi nilai pada variable \$supplier\_data

- Kolom pertama diisi dengan nilai yang ada dalam variable **\$no**. Dimulai dari angka 1 dan terus ditambahkan 1 sesuai dengan jumlah perulangan
- Kolom kedua, diisi dengan isi variable \$supplier dalam field supplierid (sesuaikan dengan nama field di database masing – masing)
- Kolom ketiga, diisi dengan isi variable \$supplier dalam field namasupplier(sesuaikan dengan nama field di database masing – masing)
- Kolom keeempat, diisi dengan isi variable
   \$supplier dalam field alamatsupplier(sesuaikan dengan nama field di database masing masing)
- Kolom kelima, diisi dengan isi variable **\$supplier** dalam field **kotasupplier**(sesuaikan dengan nama field di database masing – masing)
- Kolom keenam, membuat tampilan tombol untuk mengubah dan menghapus data

#### 6. Simpan dan jalankan program

SI Penjualan	=					🍐 Ramadhani
Online	Data Sup	Data Supplier 🔹 Beranda - Supplier				
NAVIGASI UTAMA	🚚 Daftar S	🚚 Daftar Supplier 🗕 + Tambah				+ Tambah
🏚 Beranda	No.	Supplier ID	Nama Supplier	Alamat Supplier	Kota Supplier	Aksi
DATA MASTER	1	1	Dandoh	Kelayan	Barabai	Edit Hapus
📥 Data Karyawan						
🚚 Data Supplier						
🐮 Data Pelanggan						
🕈 Data Kota						
🕒 Data Barang						
DATA TRANSAKSI						
🖸 Data Pembelian						
🗣 Data Penjualan						
LAPORAN						
📕 Laporan Pembelian						
🛃 Laporan Penjualan						

Gambar 5. 22 menampilkan data supplier

# 5.6 Menambahkan Data Master Supplier

Berikut langkah-langkah dalam menambahkan data master supplier menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Supplier

```
public function create() {
    $kota_data = $this->Kota_model->get_kota_all();

    $data = array(
        'action' => site_url('Supplier/create_action'),
        'supplierid' => set_value('supplierid'),
        'nmsupplier' => set_value('nmsupplier'),
        'almtsupplier' => set_value('almtsupplier'),
        'kotaid' => set_value('kotaid'),
        'kota_data' => $kota_data
    );

    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('template/footer');
}
```

Gambar 5. 23 function create() pada controller Supplier

- a. Memanggil fungsi get\_kota\_all() pada file
   Kota\_model.php yang ada di dalam folder model,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$kota\_data
- b. Action yang ada pada form nanti akan diproses oleh fungsi create\_action pada Controller Supplier
- c. Memasukkan nilai ke dalam array 'supplierid' pada variable \$data yang berasal dari form yang memiliki name supplierid.
- d. Memasukkan nilai ke dalam array 'nmsupplier' pada variable \$data yang berasal dari form yang memiliki name **nmsupplier**.

- e. Memasukkan nilai ke dalam array 'almtsupplier' pada variable \$data yang berasal dari form yang memiliki name **almtsupplier**.
- f. Memasukkan nilai ke dalam array 'kotaid pada variable \$data yang berasal dari form yang memiliki name kotaid.
- g. Array **'kota\_data'** pada variable **\$data** dimasukkan nilai yang sudah ada dari variable **\$kota\_data**
- h. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_supplier.php pada folder views/content/supplier
- 2. Tambahkan fungsi berikut pada Controller Supplier

Gambar 5. 24 function create\_action pada controller Supplier

- a. Memanggil fungsi \_rules() yang ada pada Controller
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi create()
- Mengambil nilai pada form dengan name 'nmsupplier',
   'almtsupplier', 'kotaid' dan dimasukkan ke array pada variable \$data

- Memanggil fungsi insert(\$data) yang ada pada model
   Supplier\_model
- e. Memanggil Controller Supplier
- 3. Tambahkan fungsi berikut pada Controller Supplier

```
public function _rules() {
    $this->form_validation->set_rules('nmsupplier', 'Nama Supplier', 'required');
    $this->form_validation->set_rules('kotaid', 'Kota Pelanggan', 'required');
    $this->form_validation->set_error_delimiters('<span class="label label-danger">', '</span>');
}
```

Gambar 5. 25 function \_rules pada controller Supplier

- a. Form yang memiliki name nmsupplier, kotaid harus diisi (*required*)
- b. Memunculkan pesan error jika tidak memenuhi syarat validasi

Fungsi yang diawali dengan *underscore*, artinya hanya bisa dipanggil melalui proses pemanggilan pada kode program, tidak bisa dipanggil melalui *address bar*.

4. Tambahkan fungsi berikut pada model Supplier\_model

```
public function insert($data) {
    $this->db->insert('supplier', $data);
}
```

Gambar 5. 26 function insert pada Supplier\_model

Keterangan :

Fungsi ini juga berfungsi untuk memasukkan nilai dalam variable **\$data** ke dalam tabel **supplier**.

5. Buat file baru dalam folder **views/content/supplier** dengan nama **form\_supplier.php** 

```
<div class="content-wrapper">
  <section class="content-header">
   <h1>Data Supplier</h1>
     <a href="<?php echo base url(); ?>"><i</a>
class="fa fa-dashboard"></i> Beranda</a>
       Supplier
       Form Supplier
     </01>
   </section>
  <section class="content">
   <div class="row">
     <section class="col-lg-12 connectedSortable">
       <div class="box box-info">
         <div class="box-header">
           <i class="fa fa-truck"></i>
           <h3 class="box-title">Form Supplier</h3>
         </div>
         <div class="box-body">
           <?php echo form open($action) ?>
           <div class="form-group">
             <div class="row">
             <div class="col-md-2 col-sm-4 col-xs-4">
             <?php echo form label('Nama Supplier'); ?>
               </div>
             <div class="col-md-10 col-sm-8 col-xs-8">
                 <?php echo form input('nmsupplier',
              'placeholder="Masukkan Nama Supplier"
$nmsupplier,
class="form-control"'); ?>
               </div>
             </div>
           </div>
           <div class="form-group">
             <div class="row">
              <div class="col-md-2 col-sm-4 col-xs-4">
                                   form label('Alamat
                <?php
                          echo
Supplier'); ?>
               </div>
             <div class="col-md-10 col-sm-8 col-xs-8">
   echo
                 <?php
form textarea('almtsupplier',
  $almtsupplier,
'placeholder="Masukkan Alamat Supplier" class="form-
control"'); ?>
               </div>
             </div>
           </div>
           <div class="form-group">
             <div class="row">
              <div class="col-md-2 col-sm-4 col-xs-4">
                <?php
                                     form label('Kota
                           echo
Supplier'); ?>
```

# 6. Simpan dan jalankan program

SI Penjualan		🔒 Ramadhani
Online	Data Supplier	Beranda -> Supplier -> Form Supplier
	🛲 Form Supplier	
🍘 Beranda	Nama Supplier	Masukkan Nama Supplier
	Alamat Supplier	Masukkan Alamat Supplier
📥 Data Karyawan		
🚚 Data Supplier		
🖀 Data Pelanggan		
	Kota Supplier	
		Banjarmasin v
		Simpan
🗣 Data Penjualan		

Gambar 5. 27 menambah data master supplier

## 5.7 Mengubah Data Master Supplier

Berikut langkah-langkah dalam mengubah data master supplier menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Supplier

```
public function edit($supplierid)
    $supplier = $this->Supplier_model->get_byid($supplierid);
    if($supplier){
        $kota data = $this->Kota model->get kota all();
        $data = array(
            'action' => site_url('Supplier/edit_action'),
            'supplierid' => set_value('supplierid', $supplier->supplierid),
            'nmsupplier' => set_value('nmsupplier', $supplier->nmsupplier),
            'almtsupplier' => set_value('almtsupplier', $supplier->almtsupplier),
            'kotaid' => set_value('kotaid', $supplier->kotaid),
            'kota_data' => $kota_data
        $this->load->view('template/header');
        $this->load->view('template/sidebar');
        $this->load->view('content/supplier/form_supplier', $data);
        $this->load->view('template/footer');
      else
        redirect(site_url('Supplier'));
```

Gambar 5. 28 function edit pada controller Supplier

- a. Memanggil fungsi get\_byid() yang ada pada *model* Supplier\_model dan memasukkan hasilnya ke dalam variable \$supplier
- b. Jika variable \$supplier ada nilainya maka, memasukkan nilai ke dalam array 'supplierid' pada variable \$data pada form yang memiliki name supplierid yang berasal dari *field* supplierid.
- c. Jika variable \$supplier ada nilainya maka, memasukkan nilai ke dalam array 'nmsupplier' pada variable \$data pada form yang memiliki name nmsupplier yang berasal dari *field* nmsupplier.

- d. Jika variable \$supplier ada nilainya maka, memasukkan nilai ke dalam array 'almtsupplier' pada variable \$data pada form yang memiliki name almtsupplier yang berasal dari *field* almtsupplier.
- e. Jika variable \$supplier ada nilainya maka, memasukkan nilai ke dalam array 'kotaid' pada variable \$data pada form yang memiliki name kotaid yang berasal dari *field* kotaid.
- f. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_supplier.php pada folder views/content/supplier
- g. Jika variable **\$supplier** tidak ada nilainya, maka memanggil *controller* **Supplier**

#### 2. Tambahkan fungsi berikut pada Controller Supplier

Gambar 5. 29 function edit\_action pada controller Supplier

Keterangan :

a. Memanggil fungsi \_rules() yang ada pada Controller

- Mengambil nilai dari form yang memiliki name supplierid kemudian dimasukkan ke dalam variable \$supplierid
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi edit(\$supplierid)
- Mengambil nilai pada form dengan name 'supplierid', 'nmsupplier', 'almtsupplier', 'kotaid' dan dimasukkan ke array pada variable \$data
- e. Memanggil fungsi **update(\$supplierid, \$data)** yang ada pada *model* **Supplier\_model**
- f. Memanggil Controller Supplier
- 3. Tambahkan fungsi berikut pada Model Supplier\_model

```
function get_byid($supplierid) {
    $sql = "SELECT supplier.* FROM supplier
    WHERE supplier.supplierid = '" . $supplierid . "'";
    return $this->db->query($sql)->row();
}
```

Gambar 5. 30 function get\_byid pada Supplier\_model

Keterangan :

Fungsi memanggil *query* untuk mengambil salah satu data didalam tabel **supplier** berdasarkan **supplierid** yang ingin diubah

4. Tambahkan fungsi berikut pada Model Supplier\_model

```
public function update($supplierid, $data) {
    $this->db->where('supplierid', $supplierid);
    $this->db->update('supplier', $data);
}
```

Gambar 5. 31 function update pada Supplier\_model

Fungsi ini berfungsi untuk mengubah nilai dalam variable \$data ke dalam tabel **supplier** dimana kuncinya adalah **supplierid**.

5. Simpan dan jalankan program

SI Penjualan			🍐 Ramadhani
<ul> <li>Online</li> </ul>	Data Supplier		Beranda > Supplier > Form Supplier
	🛲 Form Supplier		
🍘 Beranda	Nama Supplier	Dandoh	
	Alamat Supplier	Kelayan	
📥 Data Karyawan			
🚚 Data Supplier			
嶜 Data Pelanggan			
	Kota Supplier	Banjarmasin	~
		Simpan	
🔖 Data Penjualan			

Gambar 5. 32 mengubah data master supplier

# 5.8 Menghapus Data Master Supplier

Berikut langkah-langkah dalam menghapus data master supplier menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Supplier



Gambar 5. 33 function delete pada controller Supplier

- a. Memanggil fungsi get\_byid() yang ada pada model
   Supplier\_model dan memasukkan hasilnya ke dalam variable \$supplier
- b. Jika variable \$supplier ada nilainya maka memanggil fungsi delete dari **Supplier\_model**.
- c. Memanggil *controller* **Supplier**
- d. Jika variable **\$supplier** tidak ada nilainya, maka memanggil *controller* **Supplier**
- 2. Tambahkan fungsi berikut pada Model Supplier\_model

```
public function delete($supplierid) {
    $this->db->where('supplierid', $supplierid);
    $this->db->delete('supplier');
}
```

Gambar 5. 34 function delete pada Supplier\_model

Keterangan :

Fungsi ini berfungsi untuk menghapus baris di dalam tabel **supplier** dimana kuncinya adalah **supplierid** 

3. Simpan dan jalankan program

# 5.9 Latihan

Buatlah fungsi CRUD (*Create, Read, Update, Delete*) pada tabel – tabel master yang lainnya seperti :

- Pelanggan
- Karyawan, dan
- Barang

## Capaian Pembelajaran:

Mahasiswa mampu mengimplementasikan CRUD (*Create Read Update Delete*) pada pemrograman web

Bab ini membahas bagaimana membuat aplikasi menggunakan framework CodeIgniter 3 dan Bootstraps sebagai antarmuka pada menu Data Transaksi.

#### 6.1 Menampilkan Data Transaksi Pembelian

Berikut langkah-langkah dalam menampilkan data transaksi pembelian menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Pembelian.php** pada folder **sipenjualan/application/controllers**/
- Tambahkan fungsi berikut pada *Controller* Pembelian, letakkan di atas fungsi *index()*

```
class Pembelian extends CI_Controller {
    function __construct(){
        parent::__construct();
        sthis->load->model('Pembelian_model');
        sthis->load->model('Supplier_model');
        sthis->load->model('Karyawan_model');
        sthis->load->model('Barang_model');
    }
}
```

Gambar 6. 1 function \_\_construct() pada transaksi pembelian

Keterangan :

Fungsi tersebut digunakan untuk memuat filePembelian\_model.php,Supplier\_model,Karyawan\_model, Barang\_model di dalam folder model

 Buatlah sebuah file *model* baru di dalam folder sipenjualan/application/model/, beri nama Pembelian\_model.php

```
class Pembelian_model extends CI_Model {
    function __construct() {
        parent::__construct();
    }
    function get_pembelian_all() {
        $sql = "SELECT vnotapembelian.* FROM vnotapembelian";
        return $this->db->query($sql)->result();
    }
```

Gambar 6. 2 model pada transaksi pembelian

Keterangan :

Fungsi yang berfungsi untuk memanggil *query* untuk mengambil semua data didalam *view* **vnotapembelian** 

4. Tambahkan code program berikut pada file **Pembelian.php** yang ada di dalam folder *Controllers* 

```
public function index() {
    $pembelian_data = $this->Pembelian_model->get_pembelian_all();

    $data = array(
        'pembelian_data' => $pembelian_data
    );

    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('template/footer');
    $this->load->view('template/footer');
}
```

## Gambar 6. 3 function index() pada transaksi pembelian

- a. Memanggil fungsi get\_pembelian\_all() pada file
   Pembelian\_model.php yang ada di dalam folder
   *model*, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$pembelian\_data
- b. Array 'pembelian\_data' pada variable \$data dimasukkan nilai yang sudah ada dari variable \$pembelian\_data
- c. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_pembelian.php pada folder views/content/pembelian
- 5. Tambahkan kode program pada file **daftar\_pembelian.php** yang ada di dalam folder **views/content/pembelian**

```
<h3 class="box-title">Daftar Pembelian</h3>
<div class="box-tools pull-right">
<div class="btn-group">
<a href="<?php echo site_url('Pembelian/create_pembelian'); ?>" class="btn btn-primary">
<i class="fa fa-plus"></i> Tambah</a>
</div>
```


Gambar 6. 4 antarmuka daftar\_pembelian

- a. Membuat tombol di tampilan program untuk menambahkan data
- Merupakan perulangan sesuai dengan jumlah data yang ada di dalam variable **\$pembelian\_data**. Isi nilai pada variable **\$pembelian** disamakan dengan isi nilai pada variable **\$pembelian\_data**

SI Penjualan	≡	=									
Online	Data F	Data Pembelian									Beranda ⊨ Pembelian
	🖸 Dat	(III) Daftar Pembelian									+ Tambah
😰 Beranda	No.	Pembelian ID	Tanggal	Supplier	Penerima	Pengirim	Jih. Barang	Total	PPN	Grand Total	Aksi
	1	14	29-07-2021	Dandoh	Ramadhani	lko	43	107.500,00	10.750,00	118.250,00	Edit Hapus
📥 Data Karyawan	2	16	29-07-2021	Dandoh	Suanang	Riyadi	18	1.225.000,00	122.500,00	1.347.500,00	Edit Hapus
🚚 Data Supplier	3	17	30-07-2022	Dandoh	Ramadhani	2021	200	500.000,00	50.000,00	550.000,00	Edit Harves
📽 Data Pelanggan											
Stock Akhir											
	2021 - Adr	ministrasi Bisnis Poli	teknik Negeri Ba	njarmasin. All	rights reserved.						Version 1.0

#### 6. Simpan dan jalankan program

Gambar 6. 5 Menampilkan Data Transaksi Pembelian

#### 6.2 Menambahkan Data Transaksi Pembelian

Berikut langkah-langkah dalam menambahkan data transaksi pembelian menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Pembelian

```
public function create pembelian() {
   $penerima_data = $this->Karyawan_model->get_penerima();
   $supplier_data = $this->Supplier_model->get_supplier_all();
   $data = array(
        'action' => site_url('Pembelian/create_pembelian_action'),
        'pembelianid' => set_value('pembelianid'),
        'supplierid' => set_value('supplierid'),
        'penerimaid' => set_value('penerimaid'),
        'pengirim' => set_value('pengirim'),
        'penerima data' => $penerima data,
        'supplier_data' => $supplier_data
    );
   $this->load->view('template/header');
   $this->load->view('template/sidebar');
   $this->load->view('content/pembelian/form_pembelian', $data);
   $this->load->view('template/footer');
```

Gambar 6. 6 function create\_pembelian() pada controller Pembelian

- a. Memanggil fungsi get\_penerima() pada file
   Karyawan\_model.php yang ada di dalam folder
   *model*, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$penerima\_data
- b. Memanggil fungsi get\_supplier\_all() pada file
   Supplier\_model.php yang ada di dalam folder *model*,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$supplier\_data
- c. Action yang ada pada form nanti akan diproses oleh fungsi create\_pembelian\_action pada Controller Pembelian
- d. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_pembelian.php pada folder views/content/pembelian
- 2. Tambahkan fungsi berikut pada Controller Pembelian

Gambar 6. 7 function create\_pembelian\_action() pada controller Pembelian

- a. Memanggil fungsi **\_rules\_pembelian**() yang ada pada *Controller*
- b. Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi create\_pembelian()
- Memanggil fungsi insert\_pembelian(\$data) yang ada pada *model* Pembelian\_model
- Memanggil fungsi get\_lastid() pada file
   Pembelian\_model.php yang ada di dalam folder
   model, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$lastid
- e. Memanggil fungsi \_create\_itempembelian() pada controller **Pembelian**
- 3. Tambahkan fungsi berikut pada Controller Pembelian

```
public function _rules_pembelian() {
    $this->form_validation->set_rules('supplierid', 'Supplier', 'required');
    $this->form_validation->set_rules('penerimaid', 'Penerima', 'required');
    $this->form_validation->set_error_delimiters('<span class="label label-danger">', '</span>');
}
```

Gambar 6. 8 function \_rules\_pembelian() pada controller Pembelian

- c. Form yang memiliki name **supplierid** dan **penerimaid** harus diisi (*required*)
- d. Memunculkan pesan error jika tidak memenuhi syarat validasi
- 4. Tambahkan fungsi berikut pada Controller Pembelian



Gambar 6. 9 function \_create\_itempembelian() pada controller Pembelian

5. Tambahkan fungsi berikut pada Controller Pembelian



Gambar 6. 10 function create\_itempembelian\_action() pada controller Pembelian

#### 6. Tambahkan fungsi berikut pada Controller Pembelian

```
public function _rules_itempembelian() {
    $this->form_validation->set_rules('barangid', 'Barang', 'required');
    $this->form_validation->set_rules('pembelianid', 'Pembelian ID', 'required');
    $this->form_validation->set_rules('qty', 'Qty', 'required');
    $this->form_validation->set_error_delimiters('<span class="label label-danger">', '</span>');
}
```

Gambar 6. 11 function \_rules\_itempembelian() pada controller

Pembelian

7. Tambahkan fungsi berikut pada model Karyawan\_model

```
function get_penerima() {
    $sql = "SELECT vpenerima.* FROM vpenerima";
    return $this->db->query($sql)->result();
}
function get_pembuat() {
    $sql = "SELECT vpembuat.* FROM vpembuat";
    return $this->db->query($sql)->result();
}
```



8. Tambahkan fungsi berikut pada model Pembelian\_model

public function insert\_pembelian(\$data) {
 \$this->db->insert('pembelian', \$data);
}

Gambar 6. 13 function insert\_pembelian() pada Pembelian\_model

Keterangan :

Fungsi ini juga berfungsi untuk memasukkan nilai dalam variable **\$data** ke dalam tabel **pembelian**.

9. Tambahkan fungsi berikut pada model Pembelian\_model

```
function get_lastid() {
    $sql = "SELECT LAST_INSERT_ID() AS lastid";
    return $this->db->query($sql)->row();
}
```

Gambar 6. 14 function get\_lastid() pada Pembelian\_model

Keterangan :

Fungsi ini juga berfungsi untuk mendapatkan nilai dari pembelianid yang paling akhir dari tabel **pembelian**.

10. Tambahkan fungsi berikut pada model Pembelian\_model

```
function get_pembelian_byid($pembelianid) {
    $sql = "SELECT *
    FROM pembelian
    LEFT JOIN supplier ON supplier.supplierid = pembelian.supplierid
    LEFT JOIN karyawan ON karyawan.karyawanid = pembelian.penerimaid
    WHERE pembelian.pembelianid = '" . $pembelianid . "'";
    return $this->db->query($sql)->row();
}
```

Gambar 6. 15 function get\_pembelian\_byid() pada Pembelian\_model

11. Tambahkan fungsi berikut pada model Pembelian\_model



Gambar 6. 16 function get\_itempembelian() pada Pembelian\_model

12. Tambahkan fungsi berikut pada model Pembelian\_model

```
public function insert_itempembelian($barangid, $qty, $harga, $pembelianid) {
    $sql = "INSERT INTO itempembelian VALUES (".$barangid.", ".$qty.", ".$harga.", ".$pembelianid.")
    ON DUPLICATE KEY UPDATE qty = qty + ".$qty."";
    $this->db->query($sql);
}
```

# Gambar 6. 17 function insert\_itempembelian() pada Pembelian\_model

SI Penjualan	🗉 🔒 Мата Кагуа										
Online	Data Perr	Data Pembelian 🐽 Branda - Pembelian - Form Pembelian									
	D Form It	I Form Item Pembelian									
🕫 Beranda	Tanggal	2021-07-29 23:32:31									
	Supplier	Dandoh									
🏝 Data Karyawan	Penerima	Suanang									
🚚 Data Supplier	Pengirim	Riyadi									
🖀 Data Pelanggan											
	No.	Barang Sahua Lifahuau Rataan	Harga 2 500.00	Qty	Subtotal	Aksi					
		auton circutoy batang	2.300,00	10	23,000,00	Hapus					
	2	Keyboard Wireless	150.000,00	8	1.200.000,00	Hapus					
		Keyboard Wireless	۷	1		Tambah					
🗣 Data Penjualan											
Stock Akhir											
	2021 - Adminis	trasi Bisnis Politeknik Negeri Banjarmasin. All rights ret	erved.			Version 1.0					

13. Simpan dan jalankan program

Gambar 6. 18 Menambahkan Data Transaksi Pembelian

# 6.3 Mengubah Data Transaksi Pembelian

Berikut langkah-langkah dalam mengubah data transaksi pembelian menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Pembelian

```
public function edit pembelian($pembelianid) {
   $pembelian = $this->Pembelian model->qet pembelian byid($pembelianid);
   if($pembelian){
       $penerima data = $this->Karyawan model->get penerima();
       $supplier_data = $this->Supplier_model->get_supplier_all();
       $data = array(
            'action' => site_url('Pembelian/edit_pembelian_action'),
            'pembelianid' => set_value('pembelianid', $pembelian->pembelianid),
            'supplierid' => set_value('supplierid', $pembelian->supplierid),
            'penerimaid' => set_value('penerimaid', $pembelian->penerimaid),
            'pengirim' => set_value('pengirim', $pembelian->pengirim),
            'penerima_data' => $penerima_data,
            'supplier_data' => $supplier_data
        $this->load->view('template/header');
        $this->load->view('template/sidebar');
        $this->load->view('content/pembelian/form_pembelian', $data);
        $this->load->view('template/footer');
     else {
        redirect(site url('Pembelian'));
```

Gambar 6. 19 function edit\_pembelian() pada controller Pembelian

- a. Memanggil fungsi get\_pembelian\_byid() yang ada pada *model* Pembelian\_model dan memasukkan hasilnya ke dalam variable \$pembelian
- b. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_pembelian.php pada folder views/content/pembelian
- c. Jika variable **\$pembelian** tidak ada nilainya, maka memanggil *controller* **Pembelian**

2. Tambahkan fungsi berikut pada Controller Pembelian

```
public function edit_pembelian_action() {
    $this->_rules_pembelian();
    $pembelianid = $this->input->post('pembelianid', TRUE);

    if($this->form_validation->run() == FALSE) {
        $this->edit_pembelian($pembelianid);
    } else {
        $data = array(
            'supplierid' => $this->input->post('supplierid', TRUE),
            'penerimaid' => $this->input->post('penerimaid', TRUE),
            'pengirim' => $this->input->post('pengirim', TRUE)
        );
        $this->Pembelian_model->update($pembelianid, $data);
        $this->_create_itempembelian($pembelianid);
    }
}
```



- a. Memanggil fungsi **\_rules\_pembelian**() yang ada pada *Controller*
- Mengambil nilai dari form yang memiliki name pembelianid kemudian dimasukkan ke dalam variable \$pembelianid
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi edit\_pembelian(\$pembelianid)
- Mengambil nilai pada form dengan name 'supplierid', 'penerimaid', 'pengirim' dan dimasukkan ke array pada variable \$data
- e. Memanggil fungsi **update(\$pembelianid, \$data**) yang ada pada *model* **Kota\_model**
- f. Memanggil fungsi \_create\_itempembelian(\$pembelianid)

3. Tambahkan fungsi berikut pada Model Pembelian\_model

```
public function update($pembelianid, $data) {
    $this->db->where('pembelianid', $pembelianid);
    $this->db->update('pembelian', $data);
}
```

Gambar 6. 21 function update() pada Pembelian\_model

Keterangan :

Fungsi ini berfungsi untuk mengubah nilai dalam variable \$data ke dalam tabel **pembelian** dimana kuncinya adalah **pembelianid**.

4. Simpan dan jalankan program

SI Penjualan	=	3 NJ	ama Karyawan
Online	Data Pembelian	🚯 Bennda - Pembelan -	Form Pembelian
NAVIGASI UTAMA	E Form Pembelian		
🔹 Beranda	Supplier	Dandoh	~
DATA MASTER	Penerima	Suanang	~
🍐 Data Karyawan	Pengirim		
🛲 Data Supplier		Riyaal	
🗑 Data Pelanggan		Simpan	
📍 Data Kota			
DATA TRANSAKSI			
3 Data Pembelian			
🗣 Data Penjualan			
LAPORAN			
🕍 Laporan Pembelian			
🛃 Laporan Penjualan			
Stock Akhir			
	2021 - Administrasi Bisnis Politer	knik Negeri Banjarmasin. All rights reserved.	Version 1.0

Gambar 6. 22 Mengubah Data Transaksi Pembelian

## 6.4 Menghapus Data Transaksi Pembelian

Berikut langkah-langkah dalam menghapus data transaksi pembelian menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Pembelian

```
public function delete($pembelianid) {
    $pembelian = $this->Pembelian_model->get_pembelian_byid($pembelianid);
    if($pembelian) {
        $this->Pembelian_model->delete($pembelianid);
        redirect(site_url('Pembelian'));
    } else {
        redirect(site_url('Pembelian'));
    }
}
```

Gambar 6. 23 function delete() pada Controller Pembelian

Keterangan :

- a. Memanggil fungsi get\_pembelian\_byid() yang ada pada *model* Pembelian\_model dan memasukkan hasilnya ke dalam variable \$pembelian
- b. Jika variable \$pembelian ada nilainya maka memanggil fungsi delete dari **Pembelian\_model**.
- c. Memanggil controller Pembelian
- d. Jika variable **\$kota** tidak ada nilainya, maka memanggil *controller* **Pembelian**
- 2. Tambahkan fungsi berikut pada Controller Pembelian

```
public function deleteitem($barangid, $pembelianid) {
    $itempembelian = $this->Pembelian_model->get_itempembelianid($barangid, $pembelianid);
    if($itempembelian] {
        $this->Pembelian_model->deleteitem($barangid, $pembelianid);
        $this->_create_itempembelian($pembelianid);
        $ else {
            $this->_create_itempembelian($pembelianid);
        }
}
```

Gambar 6. 24 function deleteitem() pada controller Pembelian

Keterangan :

 Memanggil fungsi get\_itempembelian\_byid() yang ada pada *model* Pembelian\_model dan memasukkan hasilnya ke dalam variable \$itempembelian

- b. Jika variable \$itempembelian ada nilainya maka memanggil fungsi deleteitem dari **Pembelian\_model**.
- c. Memanggil fungsi \_create\_itempembelian
- d. Jika variable **\$itempembelian** tidak ada nilainya, maka memanggil fungsi **\_create\_itempembelian**
- 3. Tambahkan fungsi berikut pada Model Pembelian\_model

```
public function delete($pembelianid) {
    $this->db->where('pembelianid', $pembelianid);
    $this->db->delete('pembelian');
}
```

Gambar 6. 25 function delete() pada Pembelian\_model

Keterangan :

Fungsi ini berfungsi untuk menghapus baris di dalam tabel **pembelian** dimana kuncinya adalah **pembelianid** 

4. Tambahkan fungsi berikut pada Model Pembelian\_model

```
public function deleteitem($barangid, $pembelianid) {
    $this->db->where('pembelianid', $pembelianid);
    $this->db->where('barangid', $barangid);
    $this->db->delete('itempembelian');
}
```

Gambar 6. 26 function deleteitem() pada Pembelian\_model

Keterangan :

Fungsi ini berfungsi untuk menghapus baris di dalam tabel item**pembelian** dimana kuncinya adalah **pembelianid dan barangid** 

5. Simpan dan jalankan program

## 6.5 Menampilkan Data Transaksi Penjualan

Berikut langkah-langkah dalam menampilkan data transaksi penjualan menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Penjualan.php** pada folder **sipenjualan/application/controllers**/
- 2. Tambahkan fungsi berikut pada *Controller* **Penjualan**, letakkan di atas fungsi *index()*

```
class Penjualan extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('Penjualan_model');
        $this->load->model('Pelanggan_model');
        $this->load->model('Karyawan_model');
        $this->load->model('Barang_model');
    }
}
```

Gambar 6. 27 class controller Penjualan

## Keterangan :

Fungsi tersebut digunakan untuk memuat filePenjualan\_model.php,Pelanggan\_model,Karyawan\_model, Barang\_modeldi dalam folder model

 Buatlah sebuah file *model* baru di dalam folder sipenjualan/application/model/, beri nama Penjualan\_model.php

```
class Penjualan_model extends CI_Model {
    function __construct() {
        parent::__construct();
    }
    function get_penjualan_all() {
        $sql = "SELECT vnotapenjualan.* FROM vnotapenjualan";
        return $this->db->query($sql)->result();
}
```

Gambar 6. 28 class model Penjualan\_model

Keterangan :

Fungsi yang berfungsi untuk memanggil *query* untuk mengambil semua data didalam *view* **vnotapenjualan** 

4. Tambahkan code program berikut pada file **Penjualan.php** yang ada di dalam folder *Controllers* 

```
public function index() {
    $penjualan_data = $this->Penjualan_model->get_penjualan_all();
    $data = array(
        'penjualan_data' => $penjualan_data
    );
    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('content/penjualan/daftar_penjualan', $data);
    $this->load->view('template/footer');
}
```

Gambar 6. 29 function index pada controller Penjualan

- a. Memanggil fungsi get\_penjualan\_all() pada file
   Penjualan\_model.php yang ada di dalam folder
   model, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$penjualan\_data
- b. Array 'penjualan\_data' pada variable \$data dimasukkan nilai yang sudah ada dari variable \$penjualan\_data

- c. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_penjualan.php pada folder views/content/penjualan
- 5. Tambahkan kode program pada file **daftar\_penjualan.php** yang ada di dalam folder **views/content/penjualan**

```
<h3 class="box-title">Daftar Penjualan</h3>
<div class="box-tools pull-right">
<div class="btn-group">
<a href="<?php echo site_url('Penjualan/create_penjualan'); ?>"
class="btn btn-primary"><i class="fa fa-plus"></i> Tambah</a>
</div>
```

```
<div class="box-body">
 <thead>
   No.
    Penjualan ID
    Tanggal
    Pelanggan
    Pembuat
    Penerima
    Jlh. Barang
    Total
    PPN
    Grand Total
    Aksi
   </theads
  <?php $no = 1;
   foreach($peniualan data as $peniualan) { ?>
    <?php echo $no++; ?>
     <?php echo $penjualan->penjualanid; ?>
     <?php echo date('d-m-Y',strtotime($penjualan->tanggal)); ?>
     <?php echo $penjualan->nmpelanggan; ?>
     <?php echo $peniualan->pembuat: ?>
     <?php echo $penjualan->penerima; ?>
     <?php echo $penjualan->jmlbarang; ?>
     <?php echo number_format($penjualan->total,2,',','.'); ?>
     <?php echo number_format($penjualan->ppn,2,',','.'); ?>
     <?php echo number_format($penjualan->grandtotal,2,',','.'); ?>
     <a href="<?php echo site_url('Penjualan/edit_penjualan/' . $penjualan->penjualanid); ?>"
      class="btn btn-sm btn-warning">Edit</a>
      <a href="<?php echo site_url('Penjualan/delete/' . $penjualan->penjualanid); ?>"
      class="btn btn-sm btn-danger"
      onclick="return confirm('Apakah Anda Yakin ?')">Hapus</a>
     <?php } ?>
  </div>
```



- a. Membuat tombol di tampilan program untuk menambahkan data
- Merupakan perulangan sesuai dengan jumlah data yang ada di dalam variable \$penjualan\_data. Isi nilai pada variable \$penjualan disamakan dengan isi nilai pada variable \$penjualan\_data
- 6. Simpan dan jalankan program

SI Penjualan											🍐 Ramadhani	
Coline	Data	Data Penjualan 🔹 Penjuala										
	🗣 Da	iftar Penjuala	n								+ Tambah	
🍘 Beranda	No.	Penjualan ID	Tanggal	Pelanggan	Pembuat	Penerima	Jlh. Barang	Total	PPN	Grand Total	Aksi	
	1	4	30-07-2021	Ramadhani	Anang Walut	Abang	27	2.132.500,00	213.250,00	2.345.750,00	Edit Hapus	
🛔 Data Karyawan												
🛲 Data Supplier												
👹 Data Pelanggan												
🗣 Data Penjualan												
Stock Akhir												
	2021 - A	dministrasi Bisni	s Politeknik N	egeri Banjarma	sin. All rights res	erved.					Version 1.0	

Gambar 6. 31 menampilkan data transaksi penjualan

## 6.6 Menambahkan Data Transaksi Penjualan

Berikut langkah-langkah dalam menambahkan data transaksi penjualan menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Penjualan

```
public function create_penjualan() {
    $pembuat_data = $this->Karyawan_model->get_pembuat();
    $pelanggan_data = $this->Pelanggan_model->get_pelanggan_all();
    $data = array(
        'action' => site_url('Penjualan/create_penjualan_action'),
        'penjualanid' => set_value('penjualanid'),
        'pelangganid' => set_value('penbuatid'),
        'pembuatid' => set_value('penerima'),
        'penerima' => set_value('penerima'),
        'penerima' => spembuat_data,
        'pelanggan_data' => $pelanggan_data
);
    $this->load->view('template/header');
    $this->load->view('content/penjualan/form_penjualan', $data);
    $this->load->view('template/footer');
    }
}
```



- a. Memanggil fungsi get\_pembuat() pada file
   Karyawan\_model.php yang ada di dalam folder
   model, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$pembuat\_data
- Memanggil fungsi get\_pelanggan\_all() pada file
   Pelanggan\_model.php yang ada di dalam folder
   model, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$pelanggan\_data
- c. Action yang ada pada form nanti akan diproses oleh fungsi create\_penjualan\_action pada Controller Penjualan
- d. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_penjualan.php pada folder views/content/penjualan

2. Tambahkan fungsi berikut pada Controller Penjualan

```
public function create_penjualan_action() {
    $this->_rules_penjualan();
    if($this->form_validation->run() == FALSE) {
        $this->create_penjualan();
    } else {
        $data = array(
            'pelangganid' => $this->input->post('pelangganid', TRUE),
            'penerima' => $this->input->post('penerima', TRUE)
        );
        $this->Penjualan_model->insert_penjualan($data);
        $lastid = $this->Penjualan_model->get_lastid()->lastid;
        $this->_create_itempenjualan($lastid);
    }
}
```

```
Gambar 6. 33 function create_penjualan_action pada controller
Penjualan
```

- a. Memanggil fungsi **\_rules\_penjualan**() yang ada pada *Controller*
- Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi create\_penjualan()
- Memanggil fungsi insert\_penjualan(\$data) yang ada pada *model* Penjualan\_model
- Memanggil fungsi get\_lastid() pada file
   Penjualan\_model.php yang ada di dalam folder
   model, kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$lastid
- e. Memanggil fungsi \_create\_itempenjualan() pada controller **Penjualan**

3. Tambahkan fungsi berikut pada Controller Penjualan

```
public function _rules_penjualan() {
    $this->form_validation->set_rules('pelangganid', 'Pelanggan', 'required');
    $this->form_validation->set_rules('pembuatid', 'Pembuat', 'required');
    $this->form_validation->set_error_delimiters('<span class="label label-danger">', '</span>');
}
```



Keterangan :

- a. Form yang memiliki name **pelangganid** dan **pembuatid** harus diisi (*required*)
- b. Memunculkan pesan error jika tidak memenuhi syarat validasi
- 4. Tambahkan fungsi berikut pada Controller Penjualan

```
public function _create_itempenjualan($penjualanid) {
   $barang_data = $this->Barang_model->get_barang_all();
   $penjualan_data = $this->Penjualan_model->get_penjualan_byid($penjualanid);
   $itempenjualan_data = $this->Penjualan_model->get_itempenjualan_byid($penjualanid);
   $data = arrav(
       'action' => site_url('Penjualan/create_itempenjualan_action'),
        'penjualanid' => $penjualanid,
        'barangid' => set_value('barangid'),
        'qty' => set_value('qty'),
        'barang_data' => $barang_data,
        'penjualan_data' => $penjualan_data,
        'itempenjualan data' => $itempenjualan data
   ):
   $this->load->view('template/header');
   $this->load->view('template/sidebar');
   $this->load->view('content/penjualan/form_itempenjualan', $data);
   $this->load->view('template/footer');
```

Gambar 6. 35 function \_create\_itempenjualan pada controller Penjualan

5. Tambahkan fungsi berikut pada Controller Penjualan



Gambar 6. 36 function create\_itempenjualan\_action pada controller Penjualan

6. Tambahkan fungsi berikut pada Controller Penjualan

```
public function _rules_itempenjualan() {
    $this=>form_validation=>set_rules('barangid', 'Barang', 'required');
    $this=>form_validation=>set_rules('penjualanid', 'Penjualan ID', 'required');
    $this=>form_validation=>set_rules('qty', 'Qty', 'required');
    $this=>form_validation=>set_error_delimiters('<span class="label label-danger">', '</span>');
}
```

Gambar 6. 37 function \_rules\_itempenjualan pada controller Penjualan

7. Tambahkan fungsi berikut pada model Penjualan\_model

```
public function insert_penjualan($data) {
    $this->db->insert('penjualan', $data);
}
```

Gambar 6. 38 function insert pada Penjualan\_model

Fungsi ini juga berfungsi untuk memasukkan nilai dalam variable **\$data** ke dalam tabel **penjualan**.

8. Tambahkan fungsi berikut pada *model* **Penjualan\_model** 

```
function get_lastid() {
    $sql = "SELECT LAST_INSERT_ID() AS lastid";
    return $this->db->query($sql)->row();
}
```

Gambar 6. 39 function get\_lastid pada Penjualan\_model

Keterangan :

Fungsi ini juga berfungsi untuk mendapatkan nilai dari pembelianid yang paling akhir dari tabel **penjualan**.

9. Tambahkan fungsi berikut pada model Penjualan\_model



Gambar 6. 40 function get\_penjualan\_byid pada Penjualan\_model

10. Tambahkan fungsi berikut pada model Penjualan\_model

```
function get_itempenjualanid($barangid, $penjualanid) {
    $sql = "SELECT *
        FROM itempenjualan
        WHERE itempenjualan.barangid = '" . $barangid . "'
        AND itempenjualan.penjualanid = '" .$penjualanid. "' ";
    return $this->db->query($sql)->row();
}
```

Gambar 6. 41 function get\_itempenjualanid pada Penjualan\_model

11. Tambahkan fungsi berikut pada model Penjualan\_model

```
public function insert_itempenjualan($barangid, $qty, $harga, $penjualanid) {
    $sql = "INSERT INTO itempenjualan VALUES (".$barangid.", ".$qty.", ".$harga.", ".$penjualanid.")
    ON DUPLICATE KEY UPDATE qty = qty + ".$qty."";
    $this->db->query($sql);
```

Gambar 6. 42 function insert\_itempenjualan pada Penjualan\_model

12. Simpan dan jalankan program

SI Penjualan		≡ 4 Ramadhani								
<ul> <li>Online</li> </ul>	Data Per	Data Penjualan 🔹 Beranda = Penjualan - Torm Penjualan								
NAVIGASI UTAMA	🗣 Form I	tem Penjualan								
🏚 Beranda	Tanggal	2021-09-08 14:28:10	2021-09-08 14:28:10							
DATA MASTER	Pelanggan	Ramadhani	Ramadhani							
👗 Data Karyawan	Pembuat	Anang Walut								
🚚 Data Supplier	Penerima	Penerima fajar								
😁 Data Pelanggan										
📍 Data Kota	No.	Barang	Harga	Qty	Subtotal	Aksi				
Ch. Data Ramon	1	Sabun Lifebuoy Batang	2.500,00	10	25.000,00	Hapus				
	2	Keyboard Wireless	150.000,00	2	300.000,00	Hapus				
Data Pembelian		Sabun Lifebuoy Batang	~	✓ 1						
🗣 Data Penjualan										
LAPORAN										
Lat Laporan Pembelian										
🛃 Laporan Penjualan										

Gambar 6. 43 menambah data transaksi penjualan

## 6.7 Mengubah Data Transaksi Penjualan

Berikut langkah-langkah dalam mengubah data transaksi penjualan menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Penjualan

```
public function edit_penjualan($penjualanid) {
   $penjualan = $this->Penjualan_model->get_penjualan_byid($penjualanid);
    if($peniualan){
        $pembuat data = $this->Karvawan model->get pembuat():
        $pelanggan_data = $this->Pelanggan_model->get_pelanggan_all();
        $data = arrav(
            'action' => site_url('Penjualan/edit_penjualan_action'),
            'penjualanid' => set_value('penjualanid', $penjualan->penjualanid),
            'pelangganid' => set_value('pelangganid', $penjualan->pelangganid),
            'pembuatid' => set_value('pembuatid', $penjualan->pembuatid),
            'penerima' => set_value('penerima', $penjualan->penerima),
            'pembuat_data' => $pembuat_data,
            'pelanggan_data' => $pelanggan_data
        $this->load->view('template/header');
        $this->load->view('template/sidebar');
        $this->load->view('content/penjualan/form_penjualan', $data);
        $this->load->view('template/footer');
     else {
        redirect(site url('Penjualan'));
```



- Memanggil fungsi get\_penjualan\_byid() yang ada pada model Penjualan\_model dan memasukkan hasilnya ke dalam variable \$penjualan
- b. Semua nilai pada array \$data disertakan pada saat pemanggilan form\_penjualan.php pada folder views/content/penjualan
- c. Jika variable **\$penjualan** tidak ada nilainya, maka memanggil *controller* **Penjualan**
- 2. Tambahkan fungsi berikut pada Controller Penjualan

```
public function edit_penjualan_action() {
    $this->_rules_penjualan();
    $penjualanid = $this->input->post('penjualanid', TRUE);

    if($this->edit_penjualan($penjualanid);
    } else {
        $data = array(
            'pelangganid' => $this->input->post('pelangganid', TRUE),
            'penerima' => $this->input->post('penerima', TRUE)
        );
        $this->Penjualan_model->update($penjualanid, $data);
        $this->_create_itempenjualan($penjualanid);
    }
}
```



- a. Memanggil fungsi **\_rules\_penjualan**() yang ada pada *Controller*
- Mengambil nilai dari form yang memiliki name penjualanid kemudian dimasukkan ke dalam variable \$penjualanid
- c. Melakukan pengecekan validasi, jika terdapat kesalahan validasi maka proses akan dikembalikan ke fungsi edit\_penjualan(\$penjualanid)
- d. Mengambil nilai pada form dengan name 'pelangganid', 'pembuatid', 'penerima' dan dimasukkan ke array pada variable \$data
- e. Memanggil fungsi **update(\$penjualanid, \$data)** yang ada pada *model* **Penjualan\_model**
- f. Memanggil fungsi \_create\_itempenjualan(\$penjualanid)

3. Tambahkan fungsi berikut pada Model Penjualan\_model

```
public function update($penjualanid, $data) {
    $this->db->where('penjualanid', $penjualanid);
    $this->db->update('penjualan', $data);
}
```

Gambar 6. 46 function update pada Penjualan\_model

Keterangan :

Fungsi ini berfungsi untuk mengubah nilai dalam variable \$data ke dalam tabel **penjualan** dimana kuncinya adalah **penjualanid**.

4. Simpan dan jalankan program

SI Penjualan	=		<ul> <li>Ramadhani</li> </ul>
© Online	Data Penjualan		8 Beranda > Penjualan > Form Penjualan
	🗣 Form Penjualan		
🆚 Beranda	Pelanggan	Ramadhani	v
	Pembuat	Anang Walut	~
📥 Data Karyawan	Penerima	faire	
🚚 Data Supplier		10001	
🔮 Data Pelanggan		Simpan	
💊 Data Penjualan			

Gambar 6. 47 mengubah data transaksi penjualan

#### 6.8 Menghapus Data Transaksi Penjualan

Berikut langkah-langkah dalam menghapus data transaksi penjualan menggunakan CodeIgniter 3 :

1. Tambahkan fungsi berikut pada Controller Penjualan

```
public function delete($penjualanid) {
    $penjualan = $this->Penjualan_model->get_penjualan_byid($penjualanid);
    if($penjualan) {
        $this->Penjualan_model->delete($penjualanid);
        redirect(site_url('Penjualan'));
    } else {
        redirect(site_url('Penjualan'));
    }
}
```



- Memanggil fungsi get\_penjualan\_byid() yang ada pada model Penjualan\_model dan memasukkan hasilnya ke dalam variable \$penjualan
- b. Jika variable \$penjualan ada nilainya maka memanggil fungsi delete dari **Penjualan\_model**.
- c. Memanggil controller Penjualan
- d. Jika variable **\$penjualan** tidak ada nilainya, maka memanggil *controller* **Penjualan**
- 2. Tambahkan fungsi berikut pada Controller Penjualan



Gambar 6. 49 function deleteitem pada controller Penjualan

Keterangan :

 a. Memanggil fungsi get\_itempenjualan\_byid() yang ada pada *model* Penjualan\_model dan memasukkan hasilnya ke dalam variable \$itempenjualan

- b. Jika variable \$itempenjualan ada nilainya maka memanggil fungsi deleteitem dari **Penjualan\_model**.
- c. Memanggil fungsi \_create\_itempenjualan
- d. Jika variable **\$itempenjualan** tidak ada nilainya, maka memanggil fungsi **\_create\_itempenjualan**
- 3. Tambahkan fungsi berikut pada Model Penjualan\_model

```
public function delete($penjualanid) {
    $this->db->where('penjualanid', $penjualanid);
    $this->db->delete('penjualan');
}
```

Gambar 6. 50 function delete pada controller Penjualan

Keterangan :

Fungsi ini berfungsi untuk menghapus baris di dalam tabel **penjualan** dimana kuncinya adalah **penjualanid** 

4. Tambahkan fungsi berikut pada Model Penjualan\_model

```
public function deleteitem($barangid, $penjualanid) {
    $this->db->where('penjualanid', $penjualanid);
    $this->db->where('barangid', $barangid);
    $this->db->delete('itempenjualan');
}
```

Gambar 6. 51 function deleteitem pada Penjualan\_model

Keterangan :

Fungsi ini berfungsi untuk menghapus baris di dalam tabel itempenjualan dimana kuncinya adalah penjualanid dan barangid

5. Simpan dan jalankan program

# 6.9 Latihan

Buatlah fungsi CRUD (*Create, Read, Update, Delete*) pada tabel transaksi pembelian dan penjualan.

## Capaian Pembelajaran:

Mahasiswa mampu mengimplementasikan CRUD (*Create Read Update Delete*) pada pemrograman web

Bab ini membahas bagaimana membuat aplikasi menggunakan framework CodeIgniter 3 dan Bootstraps sebagai antarmuka pada menu Data Transaksi.

## 7.1 Membuat Laporan Pembelian

Berikut langkah-langkah dalam membuat laporan pembelian menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Laporan.php** pada folder **sipenjualan/application/controllers**/
- Tambahkan fungsi berikut pada Controller Laporan, letakkan di atas fungsi index()

```
class Laporan extends CI_Controller {
    function __construct(){
        parent::_construct();
        $this->load->model('Laporan_model');
    }
```

Gambar 7. 1 class controller Laporan

Keterangan :

Fungsi tersebut digunakan untuk memuat file **Laporan\_model.php** dari dalam folder model

 Buatlah sebuah file *model* baru di dalam folder sipenjualan/application/model/, beri nama Laporan\_model.php

```
class Laporan_model extends CI_Model {
    function __construct() {
        parent::__construct();
    }
}
```

Gambar 7. 2 class model Laporan\_model

4. Tambahkan code program berikut pada file **Laporan.php** yang ada di dalam folder *Controllers* 

```
public function pembelian() {
    $tahun = $this->input->post('tahun');
    $laporan_pembelian = $this->Laporan_model->get_laporan_pembelian($tahun);
    $get_tahun = $this->Laporan_model->get_tahun_laporan_pembelian($tahun);
    $data = array(
        'action' => site_url('Laporan/pembelian'),
        'tahun' => set_value('tahun', $tahun),
        'get_tahun' => $get_tahun,
        'laporan_pembelian' => $laporan_pembelian
);
    $this->load->view('template/header');
    $this->load->view('template/footer');
    $this->load->view('template/footer');
}
```

Gambar 7. 3 function pembelian pada controller Laporan

- a. Mengisi nilai variable **\$tahun** dari form dengan name **tahun**
- Memanggil fungsi get\_laporan\_pembelian() pada file
   Laporan\_model.php yang ada di dalam folder *model*,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$laporan\_pembelian

- c. Memanggil fungsi get\_tahun\_laporan\_pembelian() pada file Laporan\_model.php yang ada di dalam folder *model*, kemudian hasil pemanggilan fungsi tersebut dimasukkan ke dalam variable \$get\_tahun
- d. Array 'get\_tahun' pada variable \$data dimasukkan nilai yang sudah ada dari variable \$get\_tahun
- e. Array **'laporan\_pembelian'** pada variable **\$data** dimasukkan nilai yang sudah ada dari variable **\$laporan\_pembelian**
- f. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_laporan\_pembelian.php pada folder views/content/laporan
- 5. Tambahkan kode program pada file **Laporan\_model.php** yang ada di dalam folder *model*

```
function get_laporan_pembelian($tahun=null) {
    $sql = "SELECT vrpembelian.* FROM vrpembelian";
    if(!empty($tahun)) {
        $sql = "SELECT vrpembelian.* FROM vrpembelian
        WHERE vrpembelian.tahun = '" . $tahun . "'";
    }
    return $this->db->query($sql)->result();
}
```



Keterangan : Mengambil data dari *view* vrpembelian sesuai dengan tahun yang ditentukan

6. Tambahkan kode program pada file **Laporan\_model.php** yang ada di dalam folder *model* 

```
function get_tahun_laporan_pembelian() {
    $sql = "SELECT DISTINCT YEAR(pembelian.tanggal) as tahun
    FROM pembelian
    ORDER BY (SELECT tahun) DESC";
    return $this->db->query($sql)->result();
}
```

```
Gambar 7. 5 function get_tahun_laporan_pembelian pada
Laporan_model
```

Mengambil data tahun dari tanggal data dari tahun pembelian

 Tambahkan kode program pada file daftar\_laporan\_pembelian.php yang ada di dalam folder views/content/laporan

```
<div class="box-header">
<i class="fa fa-bar-chart"></i>
<h3 class="box-title">Laporan Pembelian</h3>
<div class="box-tools pull-right">
</div class="box-tools pull-right"</div class="box-tools pull-right">
</div class="box-tools
```

```
<div class="box-body">
 <?php echo form_open($action) ?>
 <div class="row">
   <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
     <div class="col-lg-1 col-md-1 col-sm-1 col-xs-1">
      <?php echo form_label('Filter', 'class="form-control"'); ?>
     </div>
     <div class="col-lg-9 col-md-9 col-sm-9 col-xs-9">
      <?php
           $opt_tahun[null] = 'Semua';
           foreach($get_tahun as $cbtahun) {
          $opt_tahun[$cbtahun->tahun] = $cbtahun->tahun;
         echo form_dropdown('tahun', $opt_tahun, $tahun,
         'class="form-control" style="width:100%"'); ?>
     </div>
     <div class="col-lq-2 col-md-2 col-sm-2 col-xs-2">
     <?php echo form_submit('submit', 'Filter', 'class="btn btn-success"'); ?>
     </div>
   </div>
 </div>
</div>
```

<pre><div class="box-body"></div></pre>
<thead></thead>
<pre>No.</pre>
Tahun
<pre>Nama Barang</pre>
Jan
Feb
Mar
Apr
Mei
Jun
Jul
Ags
Sep
Okt
Nop
Des

php \$no = 1;</th
<pre>foreach(\$laporan_pembelian as \$pembelian) { ?&gt;</pre>
>
php echo \$no++; ?
php echo \$pembelian- tahun; ?>
php echo \$pembelian- nmbarang; ?>
php echo \$pembelian- jan; ?>
php echo \$pembelian- feb; ?>
php echo \$pembelian- mar; ?>
php echo \$pembelian- apr; ?>
php echo \$pembelian- mei; ?>
php echo \$pembelian- jun; ?>
php echo \$pembelian- jul; ?>
php echo \$pembelian- ags; ?>
php echo \$pembelian- sep; ?>
php echo \$pembelian- okt; ?>
php echo \$pembelian- nop; ?>
php echo \$pembelian- des; ?>
php } ?

Gambar 7. 6 antarmuka daftar\_laporan\_pembelian

7.	Simpan	dan	jalankan	program
	1		5	1 0

SI Penjualan	=													۵	Ramadhani
Online	Lapora	Laporan & terr										nda > Lapo	ran Pembelian		
NAVIGASI UTAMA	🔟 Lapo	ran Pemb	elian												🕀 Cetak
🍰 Beranda	-														
DATA MASTER	Filter	St	mua									~	Filter		
🛓 Data Karyawan	No	Tahua	Nama Parang	1.00	Fab	Max	4.04	Mei	lun	tul.		f	014	Nee	Des
💭 Data Supplier	1	2022	Sabun Lifebuoy Batang	0	0	0	0	0	0	200	~£* 0	0	0	0	0
🔡 Data Pelanggan	2	2021	Sabun Lifebuoy Batang	0	0	0	0	0	0	53	0	0	0	0	0
🕈 Data Kota	3	2021	Keyboard Wireless	0	0	0	0	0	0	8	0	0	0	0	0
🗅 Data Barang															
DATA TRANSAKSI															
Data Pembelian															
🗣 Data Penjualan															
LAPORAN															
Laporan Pembelian															
🛃 Laporan Penjualan															
Stock Akhir															
	2021 - Admir	nistrasi Bisni	s Politeknik Negeri Banjarmasin. All rigi	hts reserved.											Version 1.0

Gambar 7. 7 menampilkan laporan pembelian

#### 7.2 Membuat Laporan Penjualan

Berikut langkah-langkah dalam membuat laporan penjualan menggunakan CodeIgniter 3 :

1. Tambahkan code program berikut pada file **Laporan.php** yang ada di dalam folder *Controllers* 

```
public function penjualan() {
    $tahun = $this->input->post('tahun');
    $laporan_penjualan = $this->Laporan_model->get_laporan_penjualan($tahun);
    $get_tahun = $this->Laporan_model->get_tahun_laporan_penjualan($tahun);
    $data = array(
        'action' => site_url('Laporan/penjualan'),
        'tahun' => set_value('tahun', $tahun),
        'get_tahun' => $get_tahun,
        'laporan_penjualan' => $laporan_penjualan
);
    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('template/footer');
}
```

Gambar 7. 8 function penjualan pada controller Laporan

- a. Mengisi nilai variable **\$tahun** dari form dengan name **tahun**
- Memanggil fungsi get\_laporan\_penjualan() pada file
   Laporan\_model.php yang ada di dalam folder *model*,
   kemudian hasil pemanggilan fungsi tersebut
   dimasukkan ke dalam variable \$laporan\_penjualan
- c. Memanggil fungsi get\_tahun\_laporan\_penjualan() pada file Laporan\_model.php yang ada di dalam folder *model*, kemudian hasil pemanggilan fungsi tersebut dimasukkan ke dalam variable \$get\_tahun
- d. Array 'get\_tahun' pada variable \$data dimasukkan nilai yang sudah ada dari variable \$get\_tahun
- e. Array **'laporan\_penjualan'** pada variable **\$data** dimasukkan nilai yang sudah ada dari variable **\$laporan\_penjualan**
- f. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_laporan\_pembelian.php pada folder views/content/laporan
- 2. Tambahkan kode program pada file **Laporan\_model.php** yang ada di dalam folder *model*



Gambar 7. 9 function get\_laporan\_penjualan pada Laporan\_model

Keterangan :

Mengambil data dari *view* vrpenjualan sesuai dengan tahun yang ditentukan

3. Tambahkan kode program pada file **Laporan\_model.php** yang ada di dalam folder *model* 

Gambar 7. 10 function get\_tahun\_laporan\_penjualan pada Laporan\_model

Keterangan : Mengambil data tahun dari tanggal data dari tahun penjualan 4. Tambahkan kode program pada file daftar\_laporan\_penjualan.php yang ada di dalam folder views/content/laporan

```
<div class="box-header">
<i class="fa fa-line-chart"></i>
<h3 class="box-title">Laporan Penjualan</h3>
<div class="box-tools pull-right">
</div class="box-tools pull-right">
</di
```

```
<div class="box-body">
 <?php echo form_open($action) ?>
 <div class="row">
   <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
     <div class="col-lg-1 col-md-1 col-sm-1 col-xs-1">
      <?php echo form_label('Filter', 'class="form-control"'); ?>
     </div>
     <div class="col-lg-9 col-md-9 col-sm-9 col-xs-9">
       <?php
           $opt_tahun[null] = 'Semua';
           foreach($get tahun as $cbtahun) {
          $opt_tahun[$cbtahun->tahun] = $cbtahun->tahun;
         echo form_dropdown('tahun', $opt_tahun, $tahun,
         'class="form-control" style="width:100%"');
         ?>
     </div>
     <div class="col-lg-2 col-md-2 col-sm-2 col-xs-2">
       <?php echo form_submit('submit', 'Filter', 'class="btn btn-success"'); ?>
     </div>
   </div>
 </div>
</div>
```

```
<div class="box-body">
<thead>
 No.
 Tahun
 Nama Barang
 Jan
 Feb
 Mar
 Apr
 Mei
 Jun
 Jul
 Ags
 Sep
 Okt
 Nop
 Des
 </thead>
<?php $no = 1;</pre>
foreach($laporan_penjualan as $penjualan) { ?>
-tro
```

php echo \$no++; ?
php echo \$penjualan- tahun; ?>
php echo \$penjualan- nmbarang; ?>
php echo \$penjualan- jan; ?>
php echo \$penjualan- feb; ?>
php echo \$penjualan- mar; ?>
php echo \$penjualan- apr; ?>
php echo \$penjualan- mei; ?>
php echo \$penjualan- jun; ?>
php echo \$penjualan- jul; ?>
php echo \$penjualan- ags; ?>
php echo \$penjualan- sep; ?>
php echo \$penjualan- okt; ?>
php echo \$penjualan- nop; ?>
php echo \$penjualan- des; ?>
php } ?
table>

Gambar 7. 11 antarmuka daftar\_laporan\_penjualan

</

S Penjualan	=													۵	Ramadhani
Online	Laporar	ı											(B) Ber	anda > Lapo	ran Penjualan
	🗠 Lapo	ran Penju:	alan												🕀 Cetak
😰 Beranda															
	Filter		Semua 🗸									Filter	Filter		
📥 Data Karyawan	No	Tahua	Name Parang	in a	Fab	Mar	4.04	Mai	hun.	and a	4.00	644	Old	Nee	Der
🚚 Data Supplier	1	2021	Sabun Lifebuoy Batang	0	0	0	- Apr 0	0	0	13	0	<i>зер</i> 0	0	0	0
🖀 Data Pelanggan	2	2021	Keyboard Wireless	0	0	0	0	0	0	14	0	0	0	0	0
🗣 Data Penjualan															
Stock Akhir															
	2021 - Admir	nistrasi Bisni	s Politeknik Negeri Banjarmasin. All righ	ts reserved.											Version 1.0

#### 5. Simpan dan jalankan program

Gambar 7. 12 menampilkan laporan penjualan

#### 7.3 Membuat Laporan Stock Akhir

Berikut langkah-langkah dalam membuat laporan stock akhir menggunakan CodeIgniter 3 :

1. Tambahkan code program berikut pada file **Laporan.php** yang ada di dalam folder *Controllers* 

```
public function stockakhir() {
    $stock_akhir = $this->Laporan_model->get_stock_akhir();
    $data = array(
        'stock_akhir' => $stock_akhir
    );
    $this->load->view('template/header');
    $this->load->view('template/sidebar');
    $this->load->view('content/laporan/daftar_stock_akhir', $data);
    $this->load->view('template/footer');
}
```

Gambar 7. 13 function stockakhir pada controller Laporan

Keterangan :

a. Memanggil fungsi get\_stock\_akhir() pada file
 Laporan\_model.php yang ada di dalam folder *model*,

kemudian hasil pemanggilan fungsi tersebut dimasukkan ke dalam variable **\$stock\_akhir** 

- b. Array **'stock\_akhir'** pada variable **\$data** dimasukkan nilai yang sudah ada dari variable **\$stock\_akhir**
- c. Semua nilai pada array \$data disertakan pada saat pemanggilan daftar\_stock\_akhir.php pada folder views/content/laporan
- 2. Tambahkan kode program pada file **Laporan\_model.php** yang ada di dalam folder *model*

```
function get_stock_akhir() {
    $sql = "SELECT vstockakhir.* FROM vstockakhir";
    return $this->db->query($sql)->result();
}
```

Gambar 7. 14 function get\_stock\_akhir pada Laporan\_model

Keterangan : Mengambil data dari *view* vstockakhir

 Tambahkan kode program pada file daftar\_stock\_akhir.php yang ada di dalam folder views/content/laporan

```
<div class="box-header">
  <i class="fa fa-list"></i>
  <i class="fa fa-list"></i>
  <h3 class="box-title">Laporan Stock Akhir</h3>
  <div class="box-tools pull-right">
    </div class="box-tools pull-right">
```

<div class="box-body">

```
No.
   Barang ID
   Nama Barang
   Harga
   Stock Akhir
   Status
  </thead>
 <?php $no = 1;
  foreach($stock_akhir as $stock) { ?>
   <?php echo $no++; ?>
    <?php echo $stock->barangid; ?>
    <?php echo $stock->nmbarang; ?>
    <?php echo number_format($stock->harga,2,',','.'); ?>
    <?php echo $stock->stockakhir; ?>
    <?php echo $stock->status; ?>
   <?php } ?>
 </div>
```

Gambar 7. 15 antarmuka daftar\_stock\_akhir

4. Simpan dan jalankan program

SI Penjualan	=					🍙 Ramadhani
• Online	Laporan					🏟 Beranda 🖂 Laporan Stock Akhir
NAMGASI UTAMA	I Laporan	Stock Akhir				🕀 Cetak
🍰 Beranda	No.	Barang ID	Nama Barang	Harga	Stock Akhir	Status
DATA MASTER	1	3	Keyboard Wireless	150.000,00	4	Pesan Ulang
🏝 Data Karyawan	2	1	Sabun Lifebuoy Batang	2.500,00	250	
🛲 Data Supplier	_					
😁 Data Pelanggan						
📍 Data Kota						
🗅 Data Barang						
DATA TRANSAKSI						
Data Pembelian						
🗣 Data Penjualan						
LAPORAN						
Lat. Laporan Pembelian						
🛃 Laporan Penjualan						
Stock Akhir						
	2021 - Administr	rasi Bisnis Politeknik Nege	ri Banjarmasin. All rights reserved.			Version 1.0

Gambar 7. 16 menampilkan laporan stock akhir

#### 7.4 Latihan

Buatlah modul Laporan (*Report*) pada Sistem Informasi Penjualan Anda.

#### Capaian Pembelajaran:

Mahasiswa mampu mengintegrasikan session dan autentikasi dengan framework CodeIgniter

Bab ini membahas tentang *session* dan langkah-langkah membuat halaman login pada CodeIgniter

#### 8.1 Session

Autentikasi adalah suatu mekanisme untuk mengatur hak akses suatu halaman web, biasanya diawali dengan adanya proses login. Dimana user diminta menginput data login. Jika berhasil login user berhak masuk ke suatu halaman, dan jika tidak, anda atau user tidak akan bisa masuk ke halaman tersebut. Session adalah sebuah varibel sementara yang diletakkan di server. Di mana PHP bisa mengambil nilai yang tersimpan di server walaupun kita membuka halaman baru. Biasanya session akan hilang jika anda menutup browser. Contoh penggunaan session yang dapat dilihat pada proses login, sebenarnya diwaktu menginputkan data login, pertama-tama data login akan dicek atau dicocokkan data yang disimpan dalam database, dan jika data login ditemukan dalam database maka program akan menyimpan data login pada server atau didalam browser yang anda gunakan.

Dalam PHP, session merupakan data yang disimpan dalam suatu server yang dapat digunakan secara global di server tersebut, dimana data tersebut spesifik merujuk ke user/client tertentu, contoh penggunaan session adalah ketika user telah login di halaman tertentu, maka ketika membuka halaman lain, php akan mengingat bahwa user tersebut telah login, contoh ketika kita telah login pada Google, maka setiap kali kita membuka layanan Google seperti GMail, Google Drive, dll di tab berbeda, kita akan selalu dalam keadaan login, kecuali kita buka dengan browser yang berbeda. Contoh lain ketika web ecommerce merekam keranjang belanja user, maka ketika pindah ke halaman pembayaran, daftar belanja masih terekam.

#### 8.2 Membuat Halaman Login

Berikut langkah-langkah dalam membuat halaman *login* menggunakan CodeIgniter 3 :

- 1. Buatlah satu file *controller* dengan nama **Auth.php** pada folder **sipenjualan/application/controllers/**
- 2. Tambahkan fungsi berikut pada *Controller* **Auth**, letakkan di atas fungsi *index()*

```
class Auth extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        sthis->load->model('Auth_model');
        sthis->load->model('Karyawan_model');
    }
}
```

Gambar 8. 1 Controller Auth

Keterangan :

Fungsi tersebut digunakan untuk memuat file **Auth\_model.php dan Karyawan\_model** di dalam folder model

3. Tambahkan fungsi berikut pada Controller Auth

```
public function index()
{
    if($this->is_logged_in())
    {
        redirect('Beranda');
    }
    else
    {
        $data['data_karyawan'] = $this->Karyawan_model->get_karyawan_all();
        $this->load->view('template/login',$data);
    }
}
```



- a. Memanggil fungsi is\_logged\_in() pada file
   Auth\_model.php yang ada di dalam folder *model*, jika
   hasil pemanggilan bernilai TRUE maka diarahkan ke
   *controller* Beranda
- b. Jika hasil pemanggilan fungsi is\_logged\_in() tidak bernilai TRUE, maka akan diarahkan ke halaman views/template/login
- 4. Tambahkan fungsi berikut pada controller Auth

```
public function proclogin() {
   $usr = $this->input->post('usr');
   $psw = $this->input->post('psw');
   $query = $this->db->query('SELECT * FROM vkaryawan
       WHERE karyawanid="'.$usr.'" AND password="'.$psw.'"');
   if($query->num_rows() > 0) {
       $row = $query->row();
       $karvawanid = $row->karvawanid;
       $nmkaryawan = $row->nmkaryawan;
       $hakakses = $row->hakakses;
       $datalogin = array(
            'karyawanid' => $karyawanid,
            'nmkaryawan' => $nmkaryawan.
            'hakakses' => $hakakses,
            'logged_in' => TRUE
        );
       $this->session->set_userdata($datalogin);
       redirect('Beranda');
     else {
       redirect('Auth/loginfailed');
```

Gambar 8. 3 function proclogin()

- a. Mencari data dari **vkaryawan** yang mana **karyawanid** dan **password** sesuai dengan isian form login
- b. Jika ada datanya, maka nilai karyawanid, nmkaryawan, hakakses, dan logged\_in disimpan ke dalam session dan diarahkan ke controller Beranda
- c. Jika tidak ada data, maka diarahkan ke fungsi loginfailed() pada *controller* Auth
- 5. Tambahkan fungsi berikut pada controller Auth

```
public function is_logged_in()
{
    if ($this->session->userdata('logged_in')==TRUE)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Gambar 8. 4 function is\_logged\_in()

Fungsi yang akan mengembalikan nilai **TRUE** jika **'logged\_in'** yang ada pada *session* bernilai **TRUE**, dan akan mengembalikan nilai **FALSE** jika **'logged\_in'** yang ada pada *session* bernilai **FALSE** 

6. Tambahkan fungsi berikut pada controller Auth



Gambar 8. 5 function loginfailed()

Keterangan :

a. Memanggil fungsi is\_logged\_in() pada file
 Auth\_model.php yang ada di dalam folder *model*, jika
 hasil pemanggilan bernilai TRUE maka diarahkan ke
 *controller* Beranda

- b. Jika hasil pemanggilan fungsi is\_logged\_in() tidak bernilai TRUE, maka akan diarahkan ke halaman views/template/loginfailed
- 7. Tambahkan fungsi berikut pada controller Auth

```
public function proclogout()
{
    $this->session->sess_destroy();
    redirect('Auth');
}
```

Gambar 8. 6 function proclogout()

Menghancurkan atau menghapus semua data yang sudah tersimpan di dalam session

 Tambahkan fungsi berikut di dalam function <u>construct()</u> pada setiap *controller* yang ada pada aplikasi



Gambar 8. 7 menambahkan dan memanggil function is\_logged\_in() pada function \_\_construct() di setiap controller

#### 8.3 Latihan

Buatlah proses login pada aplikasi sistem penjualan anda.

#### Capaian Pembelajaran:

Mahasiswa mampu mengembangkan *project* pemrograman web sederhana menjadi *project* pemrograman web yang sesungguhnya dengan studi kasusnya masing - masing

Bab ini membahas cara instalasi dan konfigurasi program pada website

#### 9.1 Instalasi dan Konfigurasi Program pada Website

Berikut langkah – langkah instalasi dan konfigurasi program pada website :

- 1. Siapkan folder CodeIgniter dan hasil import database yang siap diinstall di website (dalam format .zip)
- 2. Kunjungi halaman penyedia hosting dan domain
- 3. Login menggunakan username dan password untuk masuk ke halaman control panel



Gambar 9. 1 Halaman Login CPanel Website



4. Pilih file manager untuk unggah folder CodeIgniter

Gambar 9. 2 Menu File Manager pada halaman CPanel website

	· -	-			
P File Manager			Search All Your Files 🗸 fo	pr	Go Settings
+ File + Folder © Copy ⊕ Mo	we 🛓 Upload 🛓 Download 🗱 Delete 🖏 Restore 🗎 Rename	🖋 Edit 🛛 🖓 HTML	L Editor 🛛 👂 Permissions 🛛 🐵	View 🖉 Extract	⊮ Compress
+ .cphorde	Home ↑ Up One Level ← Back → Forward      Z Reloa	d 🖻 Select All 🛛	Unselect All	I Empty Trash	
+ ibm					
+ 🖿 .pki	Name	Size	Last Modified	Туре	Permissions
azor .	.well-known	4 KB	Jan 22, 2020, 10:25 PM	httpd/unix-directory	0755
💼 .sitepad	cai-bin	4 KB	Jan 22, 2020, 10:00 PM	httpd/unix-directory	0755
+ in .softaculous	historia	4 1/10	Aug 23, 2021, 7-21 BM	http://www.directory	0765
spamassassin	kinoga	4 KB	Pug 23, 2021, 7:31 PM	nupurumix-directory	0705
subaccounts	lab	4 KB	Aug 23, 2021, 7:31 PM	httpd/unix-directory	0755
🖿 .trash	idia sidia	4 KB	Aug 23, 2021, 7:31 PM	httpd/unix-directory	0750
🖿 cache	imata simata	4 KB	Jul 11, 2021, 1:10 AM	httpd/unix-directory	0755
+ 🖿 etc	sista	4 KB	Aug 23, 2021, 7:32 PM	httpd/unix-directory	0750
+ III laravel	un admia	4 KB	Aug 22, 2021, 7:31 DM	http://www.directopy	0765
logs	wpountin	4 KD	Plog 23, 2021, 7:31 PM	http://www.anectory	0133
+ mail	wp-content	4 KB	Aug 29, 2021, 10:08 AM	httpd/unix-directory	0755
+ public_ftp	wp-includes	12 KB	Feb 3, 2020, 9:28 PM	httpd/unix-directory	0755
– 🔤 public_html	htaccess	1.89 KB	Aug 13, 2021, 6:33 AM	text/x-generic	0644
+ 🖿 .well-known	Litespeed_flag	297 bytes	Feb 9, 2020, 1:07 AM	text/x-generic	0644
cgi-bin	Buterini	586 hidee	Sep 3, 2020, 0-16 PM	text/x-ceneric	0644
+ lab				tonin generie	
+ 🔤 sidia	configureset.pnp	0 bytes	Apr 19, 2021, 8:40 AM	text/x-generic	0644
+ 🔤 simata	defaultconfig.php	0 bytes	Apr 19, 2021, 8:40 AM	text/x-generic	0644
+ 🖿 sista	index.html_	77 bytes	Jan 27, 2020, 3:28 PM	text/x-generic	0644
+ 🔤 wp-admin	b index.php	420 bytes	Dec 1, 2017, 5:41 AM	text/x-generic	0644
+ wp-includes	Iang php	0 bytes	Jan 20, 2021, 12:51 AM	text/x-generic	0644
+ 🖿 ssl		10 17 10	1 0.0010.0.07.444	terdialaia	0044
+ 🖿 tmp	icense.ut	19.47 KB	Jan 2, 2019, 3:07 AM	texoplain	0044
	main_go.html	0 bytes	Jan 28, 2021, 12:45 AM	text/html	0644

5. Upload folder CodeIgniter pada folder public\_html

Gambar 9. 3 Halaman File Manager

 Setelah folder CodeIgniter berhasil diunggah, kemudian membuat database pada website pada menu MySQL Database



Gambar 9. 4 Menu MySQL Databases pada halaman CPanel Website

7. Kembali ke control panel, pilih phpMyAdmin



Gambar 9. 5 Menu phpMyAdmin pada halaman CPanel website

- 8. Pilih database kemudian import database
- Konfigurasi aplikasi pada public\_html/sipenjualan/application/config/database .php



Gambar 9. 6 konfigurasi database

Sesuaikan *hostname, username, password* dan *nama database* yang sudah diatur pada *website* 

10. Sistem dapat diakses menggunakan web browser dengan memasukkan alamat website

Gambar 9. 7 Mengakses aplikasi pada website

#### 9.2 Latihan

Ikuti langkah – langkah di atas untuk implementasi pada website

#### GLOSARIUM

Class : struktur data dari sebuah object.

**CodeIgniter** : sebuah suatu kerangka kerja yang terdiri dari *library* dan *tools* yang dipadukan untuk bahasa pemrograman PHP.

**Constructor** : method khusus yang akan dieksekusi pada saat pembuatan objek (instance).

**Controller** : class yang merespon HTTP request yang datang dari user dan dari request tersebut class ini akan menentukan apa yang akan dilanjutkan selanjutnya.

Method : aksi yang dijalankan atau dikerjakan oleh object.

**Model** : class yang berisi fungsi untuk membantu informasi dan operasi dalam database seperti *Create, Read, Update* dan *Delete* (CRUD).

**Object** : kumpulan dari variable dan fungsi yang digabung menjadi satu entitas.

**Object Oriented Programming** : pandangan pemrograman yang berbasis kepada object.

**View** : informasi yang disajikan kepada pengguna, tidak selalu menampilkan halaman web, tetapi juga dapat menampilkan fragmen halaman seperti *header*, *sidebar*, dan *footer*.

#### DAFTAR PUSTAKA

- Arief MR. Pemrograman web dinamis menggunakan PHP dan MySQL. Yogyakarta: Andi. 2011.
- Basuki AP. Proyek membangun website Berbasis PHP dengan Codeigniter. Yogyakarta: Lokomedia. 2014.
- Belajar Otodidak Framework CodeIgniter Teknik Pemrograman Web dengan PHP dan Framework CodeIgniter 3, Informatika Bandung
- Dokumentasi CodeIgniter, Website: https://www.codeigniter.com/docs
- Sianipar RH. Belajar Cepat Pemrograman Query dengan MySQL. Penerbit ANDI; 2017 Dec 27.

#### **BIOGRAFI PENULIS**



Penulis pernah mengambil Jurusan Ilmu Komputer di Universitas Lambung Mangkurat. Melanjutkan studi strata dua di Jurusan Magister Teknik Informatika di STMIK AMIKOM Yogyakarta. Pernah bekerja sebagai Dosen Tetap di Politeknik Hasnur mulai tahun 2013 sampai dengan tahun 2018. Mulai tahun 2018 hingga saat

ini berstatus sebagai dosen tetap di program studi Manajemen Informatika Politeknik Negeri Banjarmasin. Penulis aktif mengembangkan beberapa sistem informasi berbasis web baik untuk keperluan Institusi maupun diluar kampus.

# PENERAPAN FRAMEWORK

### **RAMADHANI NOOR PRATAMA**

Codelgniter merupakan sebuah *framework* berbasis *web* yang berguna untuk memudahkan para *web developer* membangun sebuah aplikasi berbasis *web. Framework* Codelgniter merupakan suatu kerangka kerja yang terdiri dari *library* dan *tools* yang dipadukan untuk bahasa pemrograman PHP. Untuk memudahkan pengembang *web* untuk membangun aplikasi dengan lebih cepat, Codelgniter menggunakan arsitektur *Model-View-Controller (MVC)* yang memisahkan kode berdasarkan penanganan proses bisnis dan untuk keperluan tampilan. Dengan menggunakan arsitektur MVC, memungkinkan para pengembang *web* dapat membangun aplikasi secara bersamaan sehingga lebih fokus pada bagian kode masingmasing tanpa mengganggu bagian kode yang lain.

## CodeIgniter



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



