

2020

Pemrograman Berbasis Objek

Pemrograman Mobile dengan Android Studio

Wanvy Arifha Saputra



Diterbitkan Atas Kerjasama
Deepublish dengan Politeknik Banjarmasin



Pemrograman Berbasis Objek
Pemrograman *Mobile* dengan
Android Studio

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

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

Pembatasan Pelindungan Pasal 26

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

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

Sanksi Pelanggaran Pasal 113

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

Pemrograman Berbasis Objek

Pemrograman *Mobile* dengan Android Studio

Wanvy Arifha Saputra



**PEMROGRAMAN BERBASIS OBJEK
PEMROGRAMAN *MOBILE* DENGAN ANDROID STUDIO**

Penulis :
Wanvy Arifha Saputra

ISBN :
978-623-7694-18-2

ISBN Elektronis :
978-623-7694-28-1

Editor dan Penyunting :
Faris Ade Irawan

Desain Sampul dan Tata Letak :
Rahma Indera; Eko Sabar Prihatin

Penerbit :
POLIBAN PRESS
Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia)
no.004.098.1.06.2019
Cetakan Pertama, 2020

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, Oktober 2020

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

Katalog Dalam Terbitan (KDT)

**Wanvy Arifha Saputra —Cet. 1.— Pemrograman Berbasis Objek, Pemrograman *Mobile*
dengan Android Studio.** Banjarmasin: Poliban Press, Oktober 2020.

x; 81 hlm.; 15.5x23 cm

Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Poliban Press karena telah mempercayakan proses percetakan buku *Pemrograman Berbasis Objek, Pemrograman Mobile dengan Android Studio* kepada Penerbit Deepublish. Semoga buku ini dapat memberikan manfaat kepada seluruh pembaca dan kerja sama ini dapat terus terjalin.



Kata Pengantar

Puji syukur ke hadirat Allah Swt. atas limpahan rahmat dan karunianya sehingga buku *Pemrograman Berbasis Objek, Pemrograman Mobile dengan Android Studio* tahun 2020 telah dapat diselesaikan. Buku ini merupakan pengantar bagi mahasiswa Diploma di Politeknik Negeri Banjarmasin.

Terima kasih disampaikan kepada Joni Riadi S.S.T., M.T. selaku Direktur Politeknik Negeri Banjarmasin dan Nurmahaludin, S.T., M.T. selaku Ketua Pusat Penelitian dan Pengabdian Masyarakat beserta sekretaris dan staf. Terima kasih 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 manfaat bagi semua pihak.

Banjarmasin, September 2020

Poliban Press

Prakata

Puji syukur ke hadirat Allah Swt., karena atas limpahan rahmat-Nya, sehingga kami dapat menyelesaikan bahan ajar *Pemrograman Berbasis Objek, Pemrograman Mobile dengan Android Studio*. Bahan ajar ini disusun berdasarkan topik tingkatan pemahaman. Bahan ajar ini juga dilengkapi dengan latihan soal untuk menguji pemahaman mahasiswa terkait dengan materi.

Kami menyadari masih banyak kekurangan dalam penyusunan bahan ajar ini. Oleh karena itu, kami sangat mengharapkan kritik dan saran demi perbaikan dan kesempurnaan bahan ajar ini.

Kami mengucapkan terima kasih kepada berbagai pihak yang telah membantu proses penyelesaian bahan ajar ini, terutama unit P3M Politeknik Negeri Banjarmasin yang telah mendukung penuh atas penerbitan bahan ajar ini. Semoga bahan ajar ini dapat bermanfaat bagi kita semua, khususnya para mahasiswa.

Banjarmasin, 30 Juni 2020

Penyusun



Wanvy Arifha Saputra, M. Kom.

Daftar Isi

Ucapan Terima Kasih.....	v
Kata Pengantar.....	vi
Prakata.....	vii
Daftar Isi.....	viii
BAB 1 PENGENALAN ANDROID	1
1.1. Teori Sistem Operasi Android.....	1
1.1.1. Sejarah	1
1.1.2. Versi Android.....	2
1.1.3. Fitur Android.....	3
1.1.4. Publikasi Android.....	4
1.2. Teori Arsitektur Android.....	4
1.3. Praktik Android Studio.....	6
1.3.1. Kebutuhan Minimum Perangkat.....	6
1.3.2. Persiapan Instalasi	7
BAB 2 DASAR-DASAR ANDROID	15
2.1. Teori Java JDK dan JRE	15
2.2. Teori Android Studio IDE.....	15
2.3. Teori Android SDK.....	15
2.4. Teori Android Virtual Devices (AVDs).....	16
2.5. Praktik Pengaturan AVDs	16
2.6. Praktik Pengaturan SDK <i>Manager</i>	18
2.7. Praktik Pembuatan Hello World	20

BAB 3	ACTIVITY DAN INTERNET	26
3.1.	Teori <i>Activity</i>	26
3.1.1.	Latihan Soal <i>Activity</i>	28
3.1.2.	Jawaban Soal <i>Activity</i>	29
3.2.	Teori <i>Intent</i>	31
3.2.1.	Latihan Soal <i>Intent</i>	31
3.2.2.	Jawaban Soal <i>Intent</i>	32
BAB 4	FRAGMENTS DAN USER INTERFACE	37
4.1.	Teori <i>Fragments</i>	37
4.1.1.	Latihan Soal <i>Fragment</i>	38
4.1.2.	Jawaban Soal <i>Fragments</i>	38
4.2.	Teori <i>User Interface</i>	49
4.2.1.	Latihan Soal <i>User Interface</i>	50
4.2.2.	Jawaban Soal <i>User Interface</i>	51
BAB 5	DATA PERSISTENCE	57
5.1.	Teori <i>Data Persistence</i>	57
5.1.1.	Shared Preferences	57
5.1.2.	File System	58
5.1.3.	Static Resources	58
5.1.4.	Android Backup Service	58
5.1.5.	Latihan Soal <i>Data Persistence</i>	58
5.1.6.	Jawaban Soal <i>Data Persistence</i>	59
5.2.	Teori <i>SQLite Database</i>	62
5.2.1.	Cursor <i>SQLite</i>	62
5.2.2.	ContentValues <i>SQLite</i>	64
5.2.3.	Mengimplementasikan <i>Database SQLite</i>	65
BAB 6	ANDROID SENSOR	71
6.1.	Teori <i>Android Sensor</i>	71
6.2.	Latihan Soal <i>Android Sensor</i>	73
6.3.	Jawaban Soal <i>Android Sensor</i>	74

Glosarium	79
Daftar Pustaka.....	81

BAB 1

PENGENALAN ANDROID

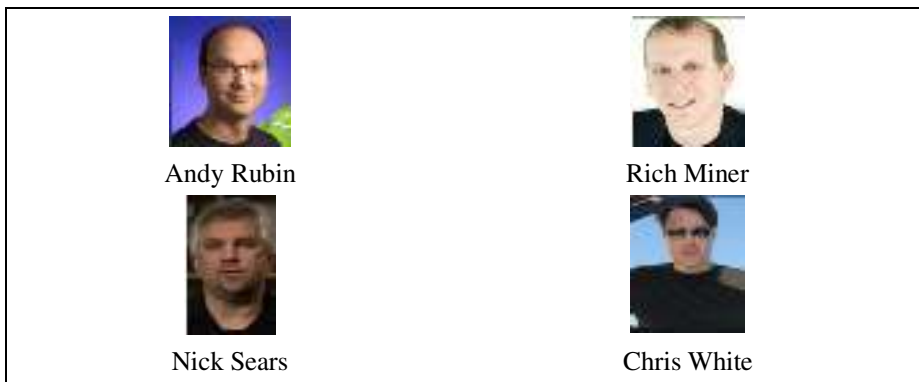
Capaian Pembelajaran

1. Mampu melakukan instalasi Android Studio.
2. Mampu menjalankan Android Studio.
3. Mampu mengetahui konsep dasar sistem operasi Android.

1.1. Teori Sistem Operasi Android

1.1.1. Sejarah

Android merupakan sistem operasi Linux yang bersifat *Open source* untuk perangkat *mobile* atau telepon genggam. Android, Inc. didirikan di Palo Alto California pada bulan Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi yang khusus bagi kamera digital. Kemudian terlihat permintaan pasar untuk perangkat tersebut kurang diminati, lalu dialihkan ke *smartphone* untuk bersaing dengan Symbian (Nokia) dan Windows Mobile (Microsoft).



Gambar 1. Pendiri Android, Inc

Bab 1 Pengenalan Android

Pada Tahun 2005, Google membeli Android dan mengambil alih proses pengembangannya hingga saat ini (2019). Pada tahun 2007, dirilislah sistem operasi pertama Android dengan ditandai juga didirikannya Open Handset Alliance (OHA) yang bertujuan untuk melakukan pengembangan standar terbuka bagi perangkat seluler. Android menawarkan pendekatan yang menyeluruh dalam pengembangan aplikasinya, sehingga aplikasi yang dibangun dapat berjalan pada berbagai perangkat sistem operasi Android seperti *smartphone*, *smartwatch*, tablet dan lainnya. Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.



Gambar 2. HTC Dream

1.1.2. Versi Android

Pada tahun 2008 merupakan awal dari sistem operasi pertama Android produk *alpha*, produk beta pada tahun 2009, dan versi komersial yaitu Cupcake pada tahun 2009.

Bab 1 Pengenalan Android

Tabel 1. Versi Sstem Operasi Android

Versi	API	Tanggal	Codename
1.0	1	23 September 2008	Alpha
1.1	2	9 Februari 2009	Beta
1.5	3	30 April 2009	Cupcake
1.6	4	15 September 2009	Donut
2.0 sd 2.1	5/6/7	26 Oktober 2009	Éclair
2.2 sd 2.2.3	8	20 Mei 2010	Froyo
2.3 sd 2.3.7	9/10	6 Desember 2010	Gingerbread
3.0 sd 3.2.6	11/12/13	22 Februari 2011	Honeycomb
4.0 sd 4.0.4	14/15	19 Oktober 2011	Ice Cream
4.1 sd 4.3.1	16/17/18	27 Juni 2012	Jelly Bean
4.4 sd 4.4.4	19/20	31 Oktober 2013	Kitkat
5.0 sd 5.1.1	21/22	12 November 2014	Lollipop
6.0 sd 6.0.1	23	5 Oktober 2015	Marshmellow
7.0	24	22 Agustus 2016	Nougat
8.0	26	21 Agustus 2017	Oreo
9.0	28	6 Agustus 2018	Pie
10.0	29	7 Agustus 2019	Q

1.1.3. Fitur Android

Sejak pertama kali diperkenalkan Android memiliki beberapa fitur utama yang wajib dimiliki oleh *smartphone* maupun perangkat lainnya dengan sistem operasi Android. Berikut adalah beberapa fitur yang disediakan dan dimanfaatkan oleh pengembang yaitu sebagai berikut.

1. *Storage, storage internal* menggunakan SQLite, sedangkan pendukung *storage external* yaitu SD Card.
2. *Connectivity*, mendukung koneksi jaringan GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, dan WiMAX.
3. *Messaging*, mendukung SMS dan MMS.
4. *Web Browser*, mendukung aplikasi *browser* yaitu Chrome.
5. *Media Support*, mendukung beragam jenis media H.263, H.264, MPEG-4, AMR, AAC, HE-AAC, MP3, MIDI, OGG, Vorbis, WAV, JPEG, PNG, GIF, BMP.

Bab 1 Pengenalan Android

6. *Hardware Support*, mendukung beragam *hardware accelerometer sensor, gyroscopes, barometers, magnetometers, thermometers, camera, digital compass, proximity and pressure sensors, GPS*.
7. *Multitouch*, mendukung *multitouch screen*.
8. *Multitasking*, mendukung sistem *multitasking*.
9. *Flash Support*, mendukung *flash 10.1*.
10. *Tethering*, mendukung *wireless hotspot*.
11. *Voice based*, mendukung pencarian berbasis suara.
12. *Screen Capture*, mendukung *screenshot* tampilan layar.
13. *Handset Layouts*, mendukung beragam layar tampilan dan dapat terkoneksi ke *external* seperti HDMI atau MiraCast.

1.1.4. Publikasi Android

Pada Agustus 2008, Google mengumumkan Android Market yang pada waktu itu diakses melalui laman <http://Android-market.co> sebagai toko aplikasi *online* yang dikhususkan bagi pengguna Android. Namun saat ini, Android Market, Google Music, dan Google Ebookstore telah menjadi satu wadah yaitu Google Playstore. Di Playstore terdapat berbagai kebutuhan pengguna dari yang bersifat *open source* atau gratis maupun berbayar.

1.2. Teori Arsitektur Android

Android adalah sekumpulan *software* berbasis Linux *open source* yang dibuat untuk berbagai perangkat. Pada diagram arsitektur menunjukkan komponen besar dari *platform* Android.

Berdasarkan Gambar 3, fondasi *platform* atau *layer* terdalam Android adalah kernel Linux. *Layer* tersebut mencakup 2 aspek yaitu driver dari *hardware* suatu perangkat yang digunakan seperti *display driver, WiFi Driver, Audio Driver, IPC driver, camera driver*. Kedua, merupakan kemampuan manajemen sistem operasi secara optimal yaitu *management power, management process* dan *management memory*. Sehingga dapat dikatakan bahwa Linux kernel merupakan layanan level dasar inti sistem.

Bab 1 Pengenalan Android



Gambar 3. Arsitektur Android

Kemudian pada *layer* kedua merupakan *Hardware Abstraction Layer* (HAL) yang berfungsi menyediakan standar *interface* untuk mengekspos atau membantu memaksimalkan kemampuan *hardware* pada *framework* Java API yang lebih tinggi. HAL terdiri atas beberapa modul *library*, setiapnya mengimplementasikan *interface* untuk komponen *hardware* tertentu, seperti modul kamera atau *bluetooth*. Bila *framework* API memanggil suatu akses *hardware*, maka sistem Android memuat modul *library* untuk komponen *hardware* tersebut.

Selanjutnya *layer* pada *Library* dan *Android Runtime* (ART). Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang

lebih tinggi, setiap aplikasi memiliki kontrol masing-masing di ART. ART dibuat untuk menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi *file* DEX, format *bytecode* yang didesain khusus untuk Android yang dioptimalkan dengan skala memori yang minimal. Beberapa fitur utama ART yaitu sebagai berikut.

1. *Ahead-of-Time* (AOT) dan *Just-in-Time* (JIT).
2. *Garbage Collector* (GC) yang dioptimalkan.
3. Dukungan *debug* yang lebih baik, mencakup *sampling profiler*, *crash reporting* dan *watchpoints* pada *line sourcecode*.

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL yang memerlukan *library* dalam C dan C++. *Platform* Android memungkinkan framework Java API mengekspos fungsionalitas beberapa *library* pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui *framework* Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D.

Framework Java API merupakan *tools* untuk mengakses Tampilan Sistem, *Resources Manager* dan *Content Provider*. Seperti dijelaskan sebelumnya bahwa *Framework* Java API sangat berguna sebagai jembatan antara layar teratas dengan HAL pada *layer* kedua.

Layer teratas merupakan *system apps*, di mana berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh *developer* dari aplikasi mereka sendiri. Misalnya, jika aplikasi ingin mengirimkan pesan SMS, tidak perlu membangun fungsionalitas tersebut sendiri sebagai gantinya Pengguna bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang telah ditetapkan.

1.3. Praktik Android Studio

1.3.1. Kebutuhan Minimum Perangkat

Persyaratan minimum sebelum instalasi perlu diketahui agar pada saat menjalankan Android Studio secara optimum. Untuk persyaratan minimum PC atau laptop tersebut ialah sebagai berikut:

Bab 1 Pengenalan Android

1. Windows 2003 (32-bit atau 64-bit),
2. Windows Vista (32-bit atau 64-bit),
3. Windows 7 (32-bit atau 64-bit),
4. Windows 8 (32-bit atau 64-bit),
5. Windows 8.1 (32-bit atau 64-bit),
6. Windows 10 (32-bit atau 64-bit),
7. Mac OS X 10.5.8 atau yang lebih baru,
8. sistem Linux dengan versi 2.11 atau yang lebih baru dari GNU C *Library* (glibc),
9. minimal 2GB RAM (direkomendasikan 4 GB),
10. 1.5 GB *storage* yang tersedia (direkomendasikan *storage* tipe *Solid State Drive* atau SSD), dan
11. *processor* Minimal Intel Core i3 atau sederajat (direkomendasikan Intel Core i5).

1.3.2. Persiapan Instalasi

1. Instalasi *Java Development Kit* (JDK)

Java Development Kit (JDK) adalah salah satu kit yang digunakan Android Studio, karena Android Studio menggunakan bahasa Java. Pertama *download* JDK terbaru pada alamat <http://www.oracle.com/technetwork/java/javase/downloads/>.



Gambar 4. Instalasi JDK

Bab 1 Pengenalan Android

2. Kemudian jalankan setup JDK tersebut, dan klik instal seperti pada Gambar 4.
3. Setelah selesai, klik *finish*, dan dilanjutkan dengan instalasi Android Studio.
4. Instalasi Android Studio dengan mengunduh Android Studio pada *link* <https://developer.Android.com/studio/index.html?hl=id#windows-bundle>, kemudian jalankan sehingga muncul tampilan seperti Gambar 5.



Gambar 5. Instalasi Android Studion (1)

5. Setelah muncul tampilan seperti di atas lalu klik *next* dan hasilnya seperti Gambar 6.



Gambar 6. Instalasi Android Studion (2)

Bab 1 Pengenalan Android

6. hasil tersebut menunjukkan bahwa yang akan di instal Android Studio yaitu SDK dan *Virtual Device* lalu klik *next* (SDK digunakan untuk mengembangkan atau membuat aplikasi untuk paket *software* tertentu, *software framework*, *hardware platform*, sistem komputer, konsol video game, sistem operasi atau *platform* sejenis lainnya, *Virtual Device* untuk *emulator* Android).



Gambar 7. Instalasi Android Studion (3)

7. Hasil tersebut adalah perjanjian dan lisensi Android Studio klik *next* saja jika tidak mau membacanya (disarankan membaca sebelum bertindak).



Gambar 8. Instalasi Android Studion (4)

Bab 1 Pengenalan Android

8. Hasil di atas menunjukkan lokasi penyimpanan untuk *install* Android, namun disarankan lokasi di C:\ lalu klik *next*.



Gambar 9. Instalasi Android Studion (5)

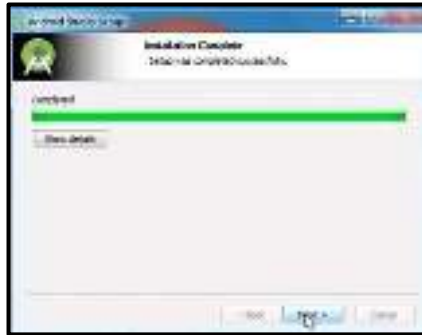
9. Hasil tersebut menunjukkan bahwa aplikasi Android Studio akan terlihat pada *desktop* Windows klik *install*.



Gambar 10. Instalasi Android Studion (6)

Bab 1 Pengenalan Android

10. Tunggu instalasi sampai selesai.



Gambar 11. Instalasi Android Studion (7)

11. Jika telah selesai klik *finish*.



Gambar 12. Instalasi Android Studion (8)

12. Ketika klik *finish* akan keluar aplikasi Android Studio seperti ini.



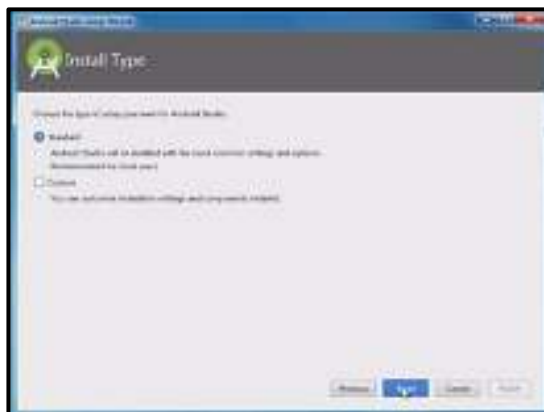
Gambar 13. Instalasi Android Studion (9)

Bab 1 Pengenalan Android



Gambar 14. Instalasi Android Studio (10)

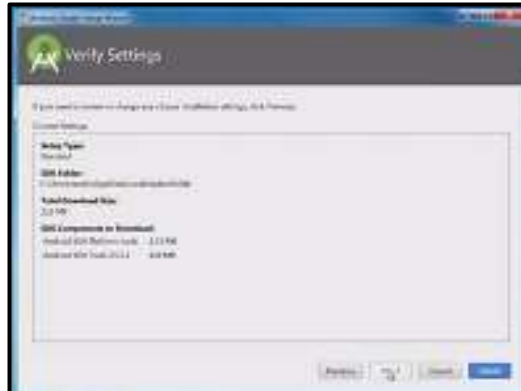
13. Tampilan di atas akan membawa kita ke instalasi selanjutnya dan disarankan untuk terhubung ke internet agar instalasi selanjutnya berhasil dan lancar dijalankan setelah instalasi, lalu klik *next*.



Gambar 15. Instalasi Android Studio (11)

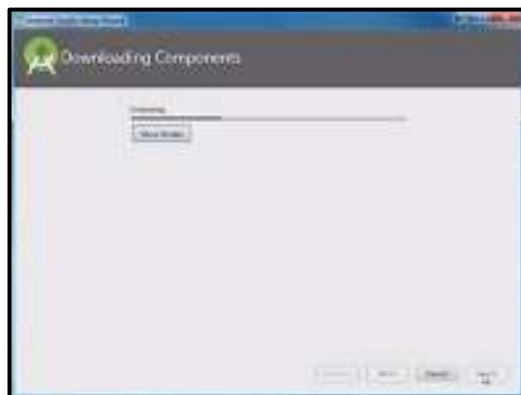
14. Gambar di atas adalah jenis instalasi, jika kita memilih standar maka akan terinstal secara *default*, lalu klik *next*.

Bab 1 Pengenalan Android



Gambar 16. Instalasi Android Studio (12)

15. Tampilan di atas menunjukkan *file* yang harus di-*download*, lalu klik *finish*.



Gambar 17. Instalasi Android Studio (13)

16. Tunggu hingga benar-benar selesai mengunduh, proses ini merupakan tahapan yang paling lama, maka dari itu persiapkan internet yang stabil dan kuota yang mencukupi untuk menjamin kelancaran instalasi.

Bab 1 Pengenalan Android



Gambar 18. Instalasi Android Studio (14)

17. Klik *finish* dan Android Studio telah siap digunakan.



Gambar 19. Instalasi Android Studio (15)

BAB 2

DASAR-DASAR ANDROID

Capaian Pembelajaran

1. Mampu membuat aplikasi *hello world* dari Android Studio.
2. Mampu mengetahui struktur proyek, *layout*, AVDs dan SDK Android.
3. Mampu menjalankan aplikasi Hello World yang dibuat.
4. Mampu mengenal *editor* IDE Android Studio.

2.1. Teori Java JDK dan JRE

Pengembangan aplikasi Android berdasarkan bahasa Java, untuk itu perlu dipastikan komputer yang digunakan sudah terpasang dengan *Java Development Kit* (JDK) dan *Java Runtime Environment* (JRE). Untuk versi minimal dalam Android Studio ialah JDK versi 7 dan JRE versi 6.

2.2. Teori Android Studio IDE

Pada 16 Mei 2013, Google meluncurkan Android Studio sebagai lingkungan pengembangan terintegrasi atau IDE (*Integrated Development Environment*) khusus pada Android. Didasarkan pada IntelliJ IDEA, Android Studio memiliki fitur tambahan dari IDE sebelumnya yaitu Eclipse ADT. Versi stabil pertama dan resmi dirilis pada Desember 2014.

2.3. Teori Android SDK

Android *system development kit* (Android SDK) ini telah disediakan secara otomatis pada instalasi Android Studio. Fungsi Android SDK ini sebagai wadah *libraries* dan *tools* pendukung dalam pembuatan aplikasi.

2.4. Teori Android Virtual Devices (AVDs)

Dalam mengembangkan suatu aplikasi Android tentunya diperlukan sebuah mesin virtual yang dapat menjalankan beragam sistem operasi dan spesifikasi pada perangkatnya. Hal ini diperlukan agar dapat melihat performa aplikasi yang dibuat dapat bekerja secara baik tanpa memiliki seluruh *smartphone* / perangkat Android lainnya secara fisik. Android Studio menyediakan secara otomatis AVDs untuk menunjang kebutuhan tersebut. AVDs dapat dilakukan konfigurasi RAM, ukuran layar, dan sistem operasi yang digunakan.

2.5. Praktik Pengaturan AVDs

1. Pada tampilan *welcome screen* Android Studio, terdapat fitur untuk melakukan pengaturan terhadap mesin *virtual* Android yang digunakan. Untuk melakukan pengaturan tersebut dapat mengklik tombol *configure* pada sisi bawah *welcome screen*, kemudian pilih *AVD manager*.



Gambar 20. Pengaturan AVDs (1)

2. Kemudian akan muncul tampilan *Android virtual device manager*, di sana terdapat tabel yang berisikan informasi mesin *virtual* yang telah dibuat, pada gambar di bawah ini sudah terpasang dua mesin *virtual* yaitu Nexus 5X dan OPPO A3S. untuk menambahkan mesin *virtual* baru klik *create virtual devices*.

Bab 2 Dasar-Dasar Android



Gambar 21. Pengaturan AVDs (2)

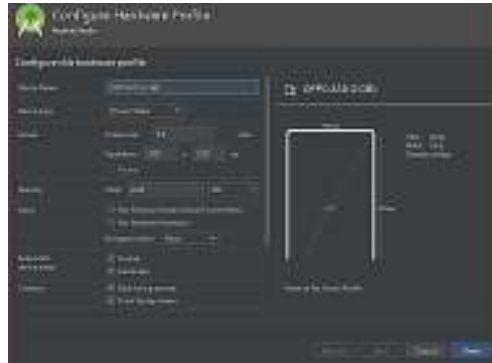
3. Di tampilan ini merupakan pemilihan mesin *virtual* perangkat yang diinstalasi. Terdapat 4 kategori pilihan yaitu TV, Phone, Wear OS, dan Tablet. Di sini kita akan fokus membuat mesin *virtual smartphone* dengan kategori *phone*.
4. Pada tampilan ini juga terdapat *template* pilihan *smartphone* yang sudah disediakan, jika terdapat *smartphone* yang diinginkan pada *list* tersebut, maka klik *Next* untuk melanjutkan. Namun, jika tidak terdapat *smartphone* yang diinginkan, maka klik *new hardware profile* untuk melanjutkan ke tahapan berikutnya.



Gambar 22. Pengaturan AVDs (3)

5. Pada tampilan ini tinggal mengisi informasi spesifikasi *smartphone* yang ingin dibuat. Informasi pada tampilan ini mencakup *device name*, *device type*, *screen size*, *screen resolution*, *memory / RAM*, *input button*, *input keyboard*, *input navigation*,

supported device states, camera, sensor, dan skin pada virtual mesin tersebut.



Gambar 23. Pengaturan AVDs (4)

2.6. Praktik Pengaturan SDK Manager

1. Pada tampilan *welcome screen* Android Studio, terdapat fitur untuk melakukan pengaturan terhadap SDK Android yang digunakan. Untuk melakukan pengaturan tersebut dapat mengklik tombol *configure* pada sisi bawah *welcome screen*, kemudian pilih *SDK manager*.
2. Pada tampilan ini terdapat 4 komponen yang penting, yaitu *Android SDK location*, *SDK platforms*, *SDK tools*, dan *SDK update sites*. Untuk *Android SDK location* merupakan lokasi instalasi Android SDK pada *folder* yang dipilih. *SDK platforms* merupakan versi sistem operasi Android pada perangkat fisik atau mesin virtual. *SDK tools* merupakan *add-ons*/fitur tambahan yang ingin digunakan. Dan terakhir yaitu *SDK update sites* yang merupakan kumpulan situs *online* dan *offline* dalam melakukan *download update* pada Android Studio di komputer.
3. Berikan tanda centang untuk menambahkan sistem operasi perangkat yang ingin digunakan, di sini akan memilih sistem operasi Android 4.4 KitKat.

Bab 2 Dasar-Dasar Android



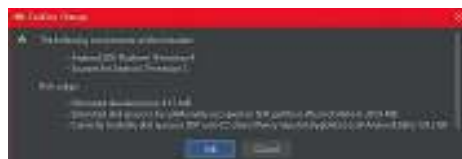
Gambar 24. Pengaturan SDK Manager (1)

4. Kemudian klik *apply* untuk melanjutkan proses selanjutnya.



Gambar 25. Pengaturan SDK Manager (2)

5. Pada pesan dialog konfirmasi, tekan OK untuk meneruskan proses *download*, dan siapkan jaringan internet dan pastikan terdapat kuota yang mencukupi bagi jaringan internet *volume-based*.



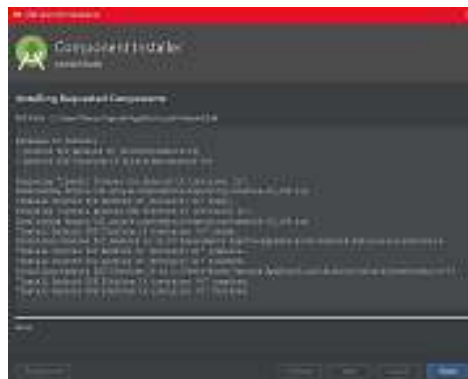
Gambar 26. Pengaturan SDK Manager (3)

Bab 2 Dasar-Dasar Android

6. Tunggu proses hingga selesai dan tekan *finish* untuk mengakhirinya.



Gambar 27. Pengaturan SDK Manager (4)



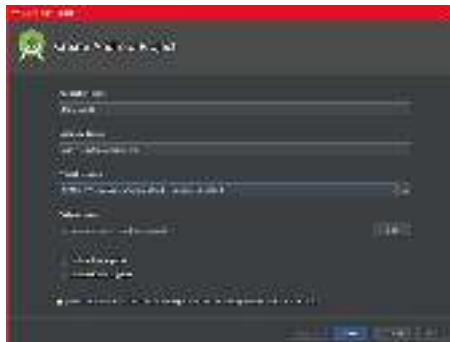
Gambar 28. Pengaturan SDK Manager (5)

2.7. Praktik Pembuatan Hello World

1. Langkah awal dari membuat aplikasi sederhana ialah dengan membuat sebuah *project* terlebih dahulu. Dapat dikatakan 1 (satu) *project* menghasilkan 1 (satu) *output* aplikasi. Namun, dalam 1 *project* terdapat lebih dari satu *Activity* ataupun *form*.

Bab 2 Dasar-Dasar Android

2. Untuk membuat *form*, pada *welcome screen* klik *start a new Android Studio project*.
3. Kemudian akan tampilan *form create Android project*. Pada tampilan ini isikan informasi berupa nama aplikasinya “Hello World”, dan *company details* isikan “wanvysaputra.example.com”, *project location* pada direktori HelloWorld, biarkan yang lainnya dalam keadaan *default* kemudian klik *next*.



Gambar 29. Aplikasi Hello World (1)

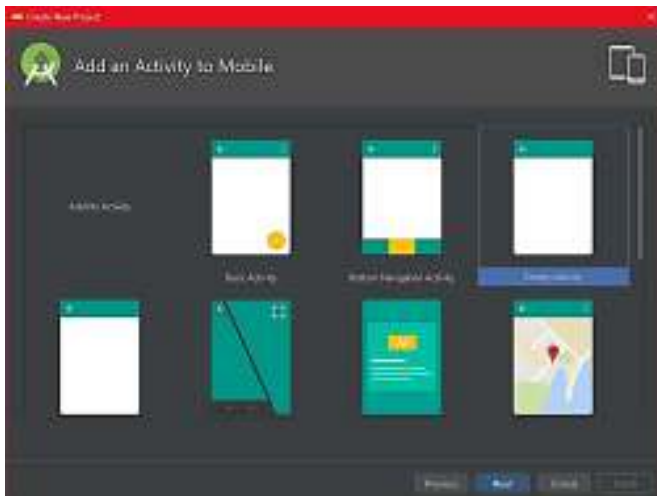
4. Pada tampilan *form target Android devices*, centang *phone and tablet*. Pilih versi sistem operasi yang digunakan minimalnya yaitu API 15: Android 4.0.3 (IceCreamSandwich), Kemudian klik *next*.



Gambar 30. Aplikasi Hello World (2)

Bab 2 Dasar-Dasar Android

5. Pada *form add an Activity mobile*, terdapat 8 pilihan *template Activity*, yaitu *Add No Activity*, *Basic Activity*, *Bottom Navigation Activity*, *Empty Activity*, dan 4 tambahan *Activity* lainnya. *Activity* merupakan modul fungsionalitas aplikasi yang biasanya berhubungan langsung dengan *layer user interface* beserta fungsionalitas di dalamnya.
6. Pada *form* ini, pilih *Empty Activity* kemudian klik *next* untuk melanjutkan ke tahapan selanjutnya



Gambar 31. Aplikasi Hello World (3)

7. Pada *form configure Activity*, masukkan *Activity name* dengan “*MainActivity*” dan centang “*generate layout file*”. Kemudian pada *layout name* masukkan dengan “*Activity_Main*” dan centang “*backwards compatibility*”
8. Kemudian klik *next* untuk melanjutkan ke tahapan selanjutnya yaitu *component installer*. Biarkan saja progress bar sampai mencapai 100% dan klik *finish* untuk memulai *project* baru tersebut.

Bab 2 Dasar-Dasar Android



Gambar 32. Aplikasi Hello World (4)



Gambar 33. Aplikasi Hello World (5)

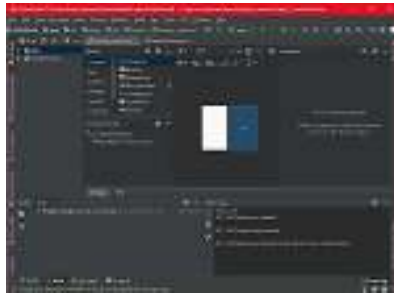
9. Setelah pembuatan *project*, maka akan muncul tampilan editor Android Studio dengan membuka *file* MainActivity.java



Gambar 34. Aplikasi Hello World (6)

Bab 2 Dasar-Dasar Android

10. Juga terdapat MainActivity.xml untuk *preview layout* dari aplikasi yang dibuat, terdapat 2 mode yaitu *mode design* dan *mode text*.



Gambar 35. Aplikasi Hello World (7)

11. Selanjutnya menjalankan program dengan mengklik tombol play ► pada editor bagian atas atau dapat juga menekan tombol SHIFT + F10
12. Siapkan kabel USB dan *smartphone* yang digunakan, pada *smartphone* pastikan *developer option* dan *USB debugging* diaktifkan. Untuk cara aktivasi fitur tersebut dapat dibuka pada pengaturan di *smartphone*. Setiap *smartphone* mempunyai pengaturan tersendiri dalam aktivasinya.
13. Jika sudah aktif, maka akan muncul pilihan *connected devices*, pilih *smartphone* yang digunakan dan tekan OK.



Gambar 36. Aplikasi Hello World (8)

Bab 2 Dasar-Dasar Android



Gambar 37. Aplikasi Hello World (9)

BAB 3

ACTIVITY DAN INTERNET

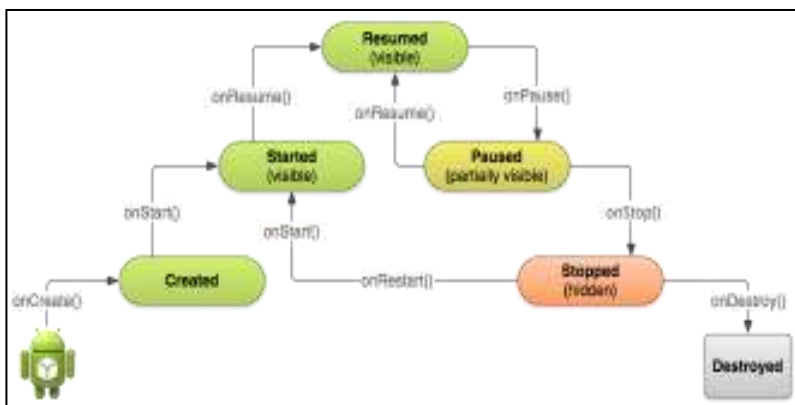
Capaian Pembelajaran

1. Mampu mengetahui *Activities* pada Android.
2. Mampu mengetahui *Intent* pada Android.
3. Mampu membuat aplikasi menggunakan *Activities* dan *Intent*.
4. Mampu menjalankan aplikasi menggunakan *Activities* dan *Intent*.

3.1. Teori Activity

Activity (aktivitas) adalah sebuah komponen pada aplikasi yang menyediakan layar untuk digunakan pengguna sehingga dapat berinteraksi seperti melakukan sesuatu hal, misalnya memilih nomor ponsel, mengambil foto, mengirim email, atau menampilkan peta.

Tiap *Activity* diberi sebuah jendela untuk menggambar antarmuka pengguna. Satu aplikasi bisa memiliki lebih dari satu *Activity*, dan setiap *Activity* memiliki siklus hidup.



Gambar 38. Siklus hidup dan *state* pada Activity

Bab 3 Activity dan Internet

Siklus hidup aplikasi Android diatur oleh sistem, berdasarkan kebutuhan pengguna, sumber daya yang tersedia, dan sebagainya. Adapun siklus hidup Android yaitu:

- *onCreate*

Method ini dieksekusi saat pertama kali *Activity* dijalankan dan melakukan pengaturan semua komponen *Activity*, seperti pengaturan *interface*. Jadi ketika *OnCreate* dijalankan, aplikasi mulai membangun dan menampilkan *interface* yang diatur.

- *onStart*

Method ini dieksekusi setelah *OnCreate* selesai dijalankan dan juga pada saat sebuah *Activity* akan terlihat oleh *user*. Pada bagian ini umumnya diisi dengan kode program, contohnya seperti pengambilan data dari *database*.

- *onRestart*

Method ini dipanggil ketika *Activity* yang berjalan pada saat itu dihentikan sementara (*paused*) dan *Activity* sebelumnya dijalankan kembali (*resume*).

- *onResume*

Method ini dieksekusi setelah *OnStart* selesai dijalankan, atau pada saat *user* kembali membuka aplikasi yang masih dalam keadaan dibelakang layar (*background state*). Pada *method* ini biasanya untuk menampilkan data. Misalnya pada saat *OnPause* terdapat proses penyimpanan inputan, dan saat *OnResume* ini kita tampilkan datanya kembali.

- *onPause*

Method ini dijalankan pada saat *Activity* berada dibelakang layar, yaitu ketika *user* keluar dari aplikasi. Biasanya *OnPause* ini untuk menyimpan data inputan, contoh kasus ketika *user* melakukan inputan teks yang sudah banyak dan panjang pada suatu komponen, kemudian tiba-tiba *user* mendapatkan telepon. Dari kondisi tersebut, data inputan tidak akan tersimpan jika tidak ditambahkan kode program penyimpanan pada *state* ini.

Bab 3 Activity dan Internet

- *onStop*

Method ini dipanggil ketika *Activity* tidak lagi ditampilkan kepada pengguna

- *onDestroy*

Method ini dipanggil sebelum *Activity* dihancurkan (*destroy*) oleh sistem (baik secara manual maupun untuk kepentingan pelonggaran memori).

Penggunaan siklus hidup aplikasi dapat diterapkan misalnya untuk membuat aplikasi yang terdapat *splash screen* sehingga saat aplikasi baru dimulai, akan menampilkan suatu gambar terlebih dahulu disertai dengan musik lalu masuk ke aplikasi utama. Siklus di atas pun berlangsung terus menerus hingga aplikasi atau *Activity* akhirnya di terminasi.

Pada contoh lainnya saat membuat aplikasi game dengan skenario. Game yang dilengkapi dengan waktu, skor dan riwayat. Apabila game sedang dijalankan, tiba-tiba ada panggilan telepon maka aplikasi harus menyimpan posisi terakhir, skor terakhir dan waktu. Maka dari itu dibutuhkan siklus hidup dalam sebuah aplikasi.

3.1.1. Latihan Soal *Activity*

Buatlah *project* baru pada Android Studio dengan nama siklus hidup. Ketentuannya ialah terdapat satu label dan nama *project* pada *header* seperti gambar di bawah ini.

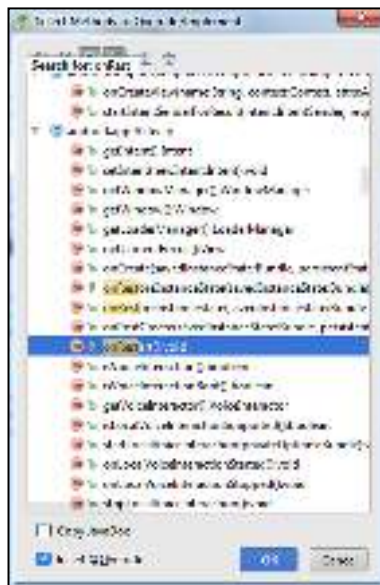


Gambar 39. Latihan Soal *Activity*

3.1.2. Jawaban Soal Activity

Jika *workspace* Android Studio sudah terbuka klik tab MainActivity.java dan tambahkan beberapa *method* dari siklus hidup di atas. Kita akan menggunakan *overriding method* onStart(), onResume(), onPause(), dan onStop().

Untuk menambahkan *method*-nya tekan CTRL + O pada MainActivity.java dan pilih 4 *method* yang ada di atas. Jika sudah tambahkan *toast* agar kita bisa melihat proses siklus hidup pada aplikasi tersebut.



Gambar 40. Jawaban Soal Activity (1)

Untuk lebih lengkapnya dapat dilihat seperti ini:

```
package wany SAPUTRA . example . com . siklushidup ;  
  
import Android . support . v7 . app . AppCompatActivity ;  
import Android . os . Bundle ;  
import Android . widget . Toast ;
```


Bab 3 Activity dan Internet

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Activity_Main);
    }

    @Override
    protected void onStart() {
        super.onStart();
        Toast.makeText(this, "Siklus hidup
onStart", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onResume() {
        super.onResume();
        Toast.makeText(this, "Siklus hidup
onResume", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onPause() {
        super.onPause();
        Toast.makeText(this, "Siklus hidup
onPause", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onStop() {
        super.onStop();
        Toast.makeText(this, "Siklus hidup
onStop", Toast.LENGTH_SHORT).show();
    }
}
```

Gambar 41. Jawaban Soal Activity (2)

Jika sudah silahkan jalankan aplikasinya dan metode akan dieksekusi secara berurutan!

3.2. Teori Intent

Intent adalah sebuah *class* dalam *programming* Android yang berfungsi untuk perpindahan *activity* / halaman. *Intent* juga merupakan suatu objek yang terdapat dalam suatu *activity* di mana objek tersebut dapat komunikasi dengan *activity* yang lain, baik *activity* pada fungsi internal Android misal seperti memanggil *activity* dalam satu *package* atau beda *package* yang masih berada dalam satu *project*.



Gambar 42. Ilustrasi *Intent*

Intent merupakan objek tipe “*Android.content.intent*”. Melalui metode *startActivity()* yang digunakan untuk memulai sebuah *activity* lain. *Intent* dibagi menjadi 2, yaitu:

- *Explicit Intent* berfungsi untuk mengaktifkan komponen-komponen dalam satu aplikasi yang sama. Misalnya seperti: *Berpindah Activity*.
- *Implicit Intent* berfungsi untuk memanggil fungsi *activity* yang sudah ada di fungsi internal Android seperti *Dial Number*, *Open Browser* dan lainnya.

Contoh kasus dapat dilihat saat membuat aplikasi dengan 3 *Activity*. *Activity* utama berisi tombol informasi dan *login*. Ketika tombol informasi di klik akan tampil *activity* informasi begitupun dengan tombol *login* ketika di klik akan menampilkan *activity login*.

3.2.1. Latihan Soal *Intent*

Buatlah Sebuah *activity* (On/Off Lampu) dengan menggunakan komponen *image* dan *button* seperti gambar di bawah ini:



Gambar 43. Latihan Soal *Intent*

3.2.2. Jawaban Soal *Intent*

Tutorial pertama belajar membuat aplikasi sederhana dengan menggunakan animasi transisi pada Android Studio. Dengan animasi transisi bisa mengatur pergantian/perpindahan dari satu *activity* ke *activity* lainnya dengan sentuhan efek animasi supaya terlihat lebih menarik. Untuk lebih jelasnya silahkan ikuti langkah-langkah berikut:

- Buat Project Baru

Silahkan Buka Android Studionya dan buat *project* baru dengan nama “Transition”, tentukan SDK, *activity* dan lainnya sampai dengan *finish* dan *workspace* Android Studio tampil.

- Tambahkan Gambar

Tambahkan dua gambar yaitu gambar lampu dalam keadaan off dan juga gambar lampu dalam keadaan *on* pada *drawable*. Gambar tersebut dapat diunduh pada mesin pencari dengan *browser*. Format gambar yang digunakan ialah png dengan *channel* transparan.

- Buat *Layout* Baru

Buat dua *layout* baru pada *directory drawable* dengan cara klik kanan pada *drawable* → New → Xml → Nama *Layout*

- 1) *Layout trans_off.xml*
- 2) *Layout trans_on.xml*

Bab 3 Activity dan Internet

Ketikkan kode program berikut pada `trans_off.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<transition
xmlns:android="http://schemas.android.com/apk/res/android">
<item android:drawable="@drawable/on"/>
<item android:drawable="@drawable/off"/>
</transition>
```

Gambar 44. Jawaban Soal *Intent* (1)

Sedangkan pada `trans_on.xml` ketikkan juga kode-kode berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<transition
xmlns:android="http://schemas.android.com/apk/res/android">
<item android:drawable="@drawable/off"/>
<item android:drawable="@drawable/on"/>
</transition>
```

Gambar 45. Jawaban Soal *Intent* (2)

- Pengaturan *Layout* Utama

Buka `Activity_Main.xml` dan ubah *background color* menjadi warna hitam dengan kode-kode sebagai berikut:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/Activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#000000"
android:paddingBottom="@dimen/Activity_vertical_margin"
android:paddingLeft="@dimen/Activity_horizontal_margin"
android:paddingRight="@dimen/Activity_horizontal_margin"
android:paddingTop="@dimen/Activity_vertical_margin"
tools:context="wanvysaputra.example.com.transition.MainActivity">
```

Gambar 46. Jawaban Soal *Intent* (3)

Bab 3 Activity dan Internet

Tambahkan *ImageView*, untuk src-nya pilih *drawable layout trans_on.xml* dengan kode-kode berikut:

```
<ImageView
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    app:srcCompat="@drawable/trans_on"
    Android:scaleType="centerInside"
    Android:layout_centerVertical="true"
    Android:layout_centerHorizontal="true"
    Android:id="@+id/imageView" />
```

Gambar 47. Jawaban Soal *Intent* (4)

Tambahkan *button* dengan kode-kode berikut:

```
<Button
    Android:text="ON / OFF"
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    Android:layout_alignParentBottom="true"
    Android:layout_centerHorizontal="true"
    Android:layout_marginBottom="34dp"
    Android:id="@+id/button" />
```

Gambar 48. Jawaban Soal *Intent* (5)

Untuk kode versi lengkapnya yaitu:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:Android="http://schemas.Android.com/apk/res/Android"
    xmlns:app="http://schemas.Android.com/apk/res-auto"
    xmlns:tools="http://schemas.Android.com/tools"
    Android:id="@+id/Activity_Main"
    Android:layout_width="match_parent"
    Android:layout_height="match_parent"
    Android:background="#000000"
    Android:paddingBottom="@dimen/Activity_vertical_margin"
    Android:paddingLeft="@dimen/Activity_horizontal_margin"
    Android:paddingRight="@dimen/Activity_horizontal_margin"
    Android:paddingTop="@dimen/Activity_vertical_margin"
    tools:context="wanvysaputra.example.com.transition.MainActivi
```

Bab 3 Activity dan Internet

```
ty">

<ImageView
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    app:srcCompat="@drawable/trans_on"
    Android:scaleType="centerInside"
    Android:layout_centerVertical="true"
    Android:layout_centerHorizontal="true"
    Android:id="@+id/imageView" />

<Button
    Android:text="ON / OFF"
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    Android:layout_alignParentBottom="true"
    Android:layout_centerHorizontal="true"
    Android:layout_marginBottom="34dp"
    Android:id="@+id/button" />

</RelativeLayout>
```

Gambar 49. Jawaban Soal *Intent* (5)

- Pengaturan MainActivity

Pada MainActivity.java kita akan memasukan kode-kode Java untuk memberikan *action* pada *button* (*on/off*) yang telah dibuat pada *activityMain.xml* dan juga pengaturan animasi transisi termasuk waktu berapa lama efek transisi tampil.

```
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(!turnOn){
            imageView.setImageResource(R.drawable.trans_on);
            ((TransitionDrawable)imageView.getDrawable()).startTransition(3000);
            turnOn=true;
        }else {
            imageView.setImageResource(R.drawable.trans_off);
            ((TransitionDrawable)imageView.getDrawable()).startTransition(3000);
            turnOn=false;
        }
    }
});
```

Bab 3 Activity dan Internet

```
}  
}  
});
```

Gambar 50. Jawaban Soal *Intent* (6)

- Pengaturan Nama Aplikasi

Buka `string.xml` masukan kode-kode berikut:

```
<resources>  
<string name="app_name">Animasi Transisi</string>  
</resources>
```

Gambar 51. Jawaban Soal *Intent* (7)

Jika sudah diikuti langkah-langkah di atas, silahkan jalankan aplikasi menggunakan emulator atau langsung ke *device* Android dan lihat hasilnya.

BAB 4

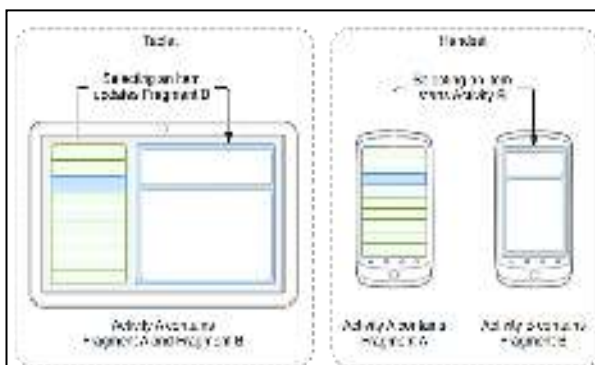
FRAGMENTS DAN USER INTERFACE

Capaian Pembelajaran

1. Mampu mengetahui *fragments* dan *user interface* pada Android.
2. Mampu membuat aplikasi menggunakan *fragments* dan *user interface*.
3. Mampu menjalankan aplikasi dengan *fragments* dan *user interface*.

4.1. Teori *Fragments*

Fragment adalah sebuah *reusable class* yang mengimplementasikan beberapa fitur sebuah *activity*. *Fragment* biasanya dibuat sebagai bagian dari suatu antarmuka. Sebuah *fragment* harus berada di dalam sebuah *activity*, mereka tidak dapat berjalan sendiri tanpa adanya *activity* tempat mereka menempel.



Gambar 52. Ilustrasi *Fragments*

Sebuah *Fragment* merupakan kombinasi sebuah *layout XML* dan kelas *java* yang mirip dengan sebuah *activity*. Dengan menggunakan

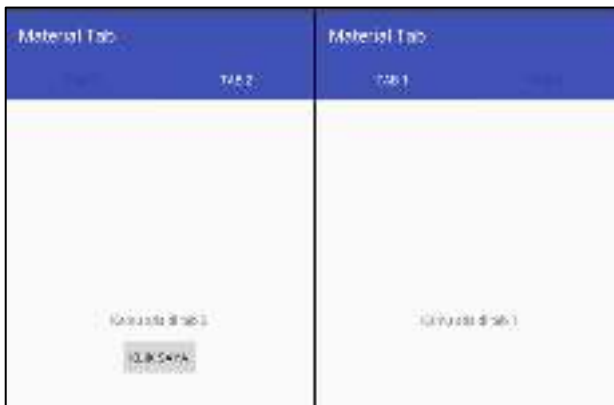
Bab 4 Fragments dan User Interface

support library, *fragment* dapat mendukung hampir semua versi Android. *Fragment* juga dapat dipakai berulang kali di dalam *Activity*. *Fragment* merupakan komponen utuh yang memiliki *view*, *event*, dan *logic* (meskipun tetap membutuhkan sebuah *fragment* agar dapat bekerja).

Ada banyak kasus yang dapat diselesaikan menggunakan *fragment*, namun yang paling umum adalah penggunaan komponen *view* dan *logic* berulang kali, menampilkan data, melakukan *event* tertentu di beberapa *activity* berbeda, membuat antarmuka sesuai dengan perangkat yang membukanya, dan menggunakan orientasi layar tampilan (*portrait/landscape*) yang berbeda menggunakan elemen yang sama.

4.1.1. Latihan Soal *Fragment*

Buatlah sebuah *Fragment* seperti gambar di bawah ini:



Gambar 53. Latihan Soal *Fragments*

4.1.2. Jawaban Soal *Fragments*

1. Buat *Project* dengan Start a new Android Studio *Project*
2. Kemudian *Configure your new project*, isikan *Application Name* dengan Belajar *Fragment* dan *Company domain* sesuai dengan nama yang kamu inginkan. Kemudian klik *Next* hingga *Finish*. Biarkan *gradle* meng-generate *project* kita.

Bab 4 Fragments dan User Interface



Gambar 54. Jawaban Soal *Fragments* (1)

3. Tambahkan *Library Design* dengan cara pilih menu *File* → *Project Structure*, kemudian pilih *Modules app* → *tab Dependencies* → klik “+”, kemudian pilih *Library Dependency* → *design*, dalam hal ini untuk versi yang dipakai adalah: *compile* ‘com.Android.support:design:24.0.0’, lalu klik OK.



Gambar 55. Jawaban Soal *Fragments* (2)

4. Pada *file styles.xml* (folder *res/values*) modifikasi kode di dalamnya sehingga menjadi seperti ini:

```
<resources>
<!-- Base application theme. -->
<style name="AppTheme"
parent="Theme.AppCompat.Light.NoActionBar">
<!-- Customize your theme here. -->
<item name="android:windowActionBar">false</item>
</style>

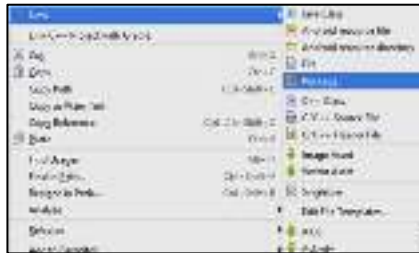
<!-- mendefinisikan style untuk TabLayout -->
```

Bab 4 Fragments dan User Interface

```
<style name="MyCustomTabLayout"
parent="Widget.Design.TabLayout">
<item
name="tabIndicatorColor">@color/colorPrimaryDark</item>
</style>
</resources>
```

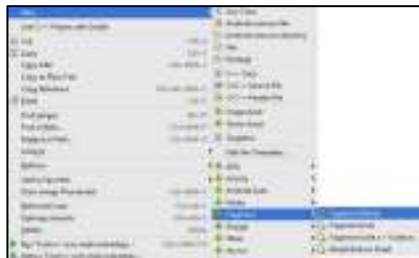
Gambar 56. Jawaban Soal *Fragments* (3)

5. Selanjutnya membuat *content* dari tab-tab yang akan ditampilkan. Buat *Package* atau direktori baru dengan nama *fragment* dengan cara: klik kanan pada *package* utama → *New* → *Package* dan beri nama *fragment*.



Gambar 57. Jawaban Soal *Fragments* (4)

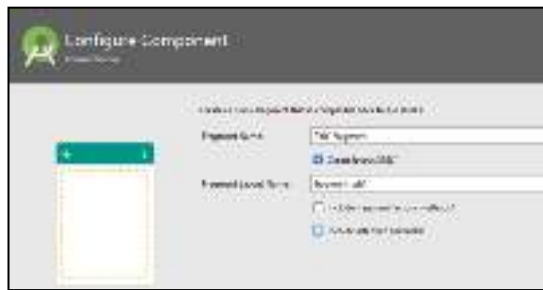
6. Kemudian buat dua *class fragment* di dalamnya masing-masing *Tab1 Fragment* dan *Tab2 Fragment* dengan cara Klik kanan pada *package fragment* → *New* → *Fragment* → *Blank Fragment*.



Gambar 58. Jawaban Soal *Fragments* (5)

Bab 4 Fragments dan User Interface

7. Muncul dialog *form* seperti berikut, beri nama *Tab1Fragment* dan *uncheck Include fragment factory methods*. Dan *uncheck Include interface callbacks*. Lalu klik *finish*.



Gambar 59. Jawaban Soal *Fragments* (6)

8. Ulangi langkah 6 di atas untuk membuat *Tab2Fragment*
9. Buka *file strings.xml* (folder *res/values*) dan tambahkan kode sebagai berikut:

```
<resources>
  <string name="app_name">Belajar Fragment</string>

  <!-- TODO: Remove or change this placeholder text -->
  <string name="hello_blank_fragment">Hello blank
  fragment</string>

  <string name="tab_1">Kamu ada di tab 1</string>
  <string name="tab_2">Kamu ada di tab 2</string>
</resources>
```

Gambar 60. Jawaban Soal *Fragments* (7)

10. Pada *fragment_tab1.xml* lengkapi kodenya seperti di bawah ini:

```
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
```

Bab 4 Fragments dan User Interface

```
        Android:layout_height="match_parent"

tools:context="com.wanvysaputra.example.belajarfragment.fragment.Tab1Fragment">

    <TextView
        Android:layout_width="wrap_content"
        Android:layout_height="wrap_content"
        Android:text="@string/tab_1"
        Android:layout_centerHorizontal="true"
        Android:layout_centerVertical="true"/>
</RelativeLayout>
```

Gambar 61. Jawaban Soal *Fragments* (8)

11. Pada `Tab1Fragment.java` kodenya seperti di bawah ini:

```
import Android.os.Bundle;
import Android.support.v4.app.Fragment;
import Android.view.LayoutInflater;
import Android.view.View;
import Android.view.ViewGroup;
import com.wanvysaputra.example.belajarfragment.R;

/**
 * A simple {@link Fragment} subclass.
 */
public class Tab1Fragment extends Fragment {

    public Tab1Fragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
        container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_tab1, container,
            false);
    }
}
```

Gambar 62. Jawaban Soal *Fragments* (9)

Bab 4 Fragments dan User Interface

12. Pada `fragment_tab2.xml` silakan lengkapi kodenya seperti di bawah ini, Sedikit berbeda dengan `fragment_tab1.xml`, kita menambahkan sebuah *button* di `fragment_tab2.xml`.

```
<RelativeLayout
xmlns:android="http://schemas.Android.com/apk/res/Android"
xmlns:tools="http://schemas.Android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context="com.wanvysaputra.example.belajarfragment.fragment
ent.Tab2Fragment">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/tab_2"
android:id="@+id/txt_content_tab_2"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"/>

<Button
android:id="@+id/btn_click_me"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Klik saya"
android:layout_below="@+id/txt_content_tab_2"
android:layout_centerHorizontal="true"
android:layout_marginTop="10dp"/>

</RelativeLayout>
```

Gambar 63. Jawaban Soal *Fragments* (10)

13. Pada `Tab2Fragment.java` silakan lengkapi kodenya seperti di bawah ini. Untuk melakukan *import Android support* tekan ALT+Enter

```
import Android.os.Bundle;
import Android.support.v4.app.Fragment;
import Android.view.LayoutInflater;
import Android.view.View;
import Android.view.ViewGroup;
import Android.widget.Button;
```

Bab 4 Fragments dan User Interface

```
import Android.widget.Toast;
import com.wanvysaputra.example.belajarfragment.R;

/**
 * A simple {@link Fragment} subclass.
 */
public class Tab2Fragment extends Fragment {

    private Button btnClickMe;

    public Tab2Fragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
    container,
    Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_tab2,
    container, false);
        btnClickMe = (Button) view.findViewById(R.id.btn_click_me);
        return view;
    }

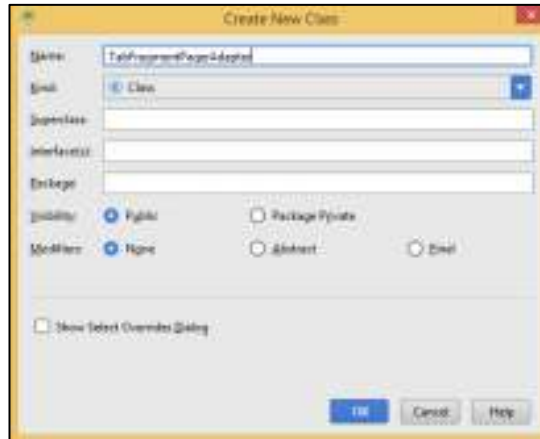
    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        btnClickMe.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(getActivity(), "Halo",
                Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Gambar 64. Jawaban Soal *Fragments* (11)

14. Sekarang kita akan membuat *Adapter* untuk *Fragment-fragment* yang telah kita buat. Buat *package* baru dengan cara klik kanan di *package* utama → *New* → *Package* → dan beri nama *adapter*. Di dalam *package directory* buat *class Adapter* dengan cara Klik kanan

Bab 4 Fragments dan User Interface

pada *package adapter* → *New* → *Java Class* dan beri nama “TabFragmentPagerAdapter”.



Gambar 65. Jawaban Soal *Fragments* (12)

15. Pada `TabFragmentPagerAdapter.java` lengkapi kodenya sebagai berikut dan sesuaikan baris-baris yang memuat *Package name* dan *Fragment Name*.

```
import Android.support.v4.app.Fragment;
import Android.support.v4.app.FragmentManager;
import Android.support.v4.app.FragmentPagerAdapter;
import
com.wanvysaputra.example.belajarfragment.fragment.Tab1Fragment;
import
com.wanvysaputra.example.belajarfragment.fragment.Tab2Fragment;

/**
 * Created by Windows on 06/06/2020.
 */
public class TabFragmentPagerAdapter extends
FragmentPagerAdapter {
    //nama tab nya
    String[] title = new String[] {
```


Bab 4 Fragments dan User Interface

```
"Tab 1", "Tab 2"
};

public TabPagerAdapter(FragmentManager fm) {
    super(fm);
}

//method ini yang akan memanipulasi penampilan Fragment
dilayar
@Override
public Fragment getItem(int position) {
    Fragment fragment = null;
    switch (position){
        case 0:
            fragment = new Tab1Fragment();
            break;
        case 1:
            fragment = new Tab2Fragment();
            break;
        default:
            fragment = null;
            break;
    }
    return fragment;
}

@Override
public CharSequence getPageTitle(int position) {
    return title[position];
}

@Override
public int getCount() {
    return title.length;
}
}
```

Gambar 66. Jawaban Soal *Fragments* (13)

16. Pada direktori *layout* buat *file resource layout* baru dengan cara klik kanan pada direktori *layout* → *New* → *Layout resource file* dan beri nama “*toolbar.xml*”. Dan kemudian lengkapi kodenya sebagai berikut. *Toolbar* ini akan menjadi pengganti dari *Action Bar* pada umumnya.

Bab 4 Fragments dan User Interface

```
<Android.support.v7.widget.Toolbar
  Android:layout_height="wrap_content"
  Android:layout_width="match_parent"
  Android:background="@color/colorPrimary"
  Android:elevation="0dp"
  Android:theme="@style/Base.ThemeOverlay.AppCompat.Dark"
  xmlns:Android="http://schemas.Android.com/apk/res/Android"
/>
```

Gambar 67. Jawaban Soal *Fragments* (14)

17. Pada Activity_Main.xml lengkapi kodenya sebagai berikut.

```
<LinearLayout
  xmlns:Android="http://schemas.Android.com/apk/res/Android"
  xmlns:tools="http://schemas.Android.com/tools"
  Android:layout_width="match_parent"
  Android:layout_height="match_parent"
  Android:orientation="vertical"

  tools:context="com.wanvysaputra.example.belajarfragment.MainA
  ctivity">
  <include
    Android:id="@+id/tool_bar"
    layout="@layout/toolbar"
    Android:layout_height="wrap_content"
    Android:layout_width="match_parent" />
  <Android.support.design.widget.TabLayout
    Android:id="@+id/tabs"
    Android:layout_width="match_parent"
    Android:layout_height="wrap_content"
    Android:elevation="0dp"
    style="@style/MyCustomTabLayout"
    Android:background="@color/colorPrimary"/>
  <Android.support.v4.view.ViewPager
    Android:id="@+id/pager"
    Android:layout_height="match_parent"
    Android:layout_width="match_parent"
    Android:layout_weight="1">
  </Android.support.v4.view.ViewPager>
</LinearLayout>
```

Gambar 68. Jawaban Soal *Fragments* (15)

Bab 4 Fragments dan User Interface

18. Dapat dilihat pada baris kode di atas kita menggunakan beberapa komponen UI seperti *TabLayout* dan *ViewPager* yang dibungkus dengan *Container* utama yaitu *LinearLayout* dengan konfigurasi *Vertical* pada orientasinya.
19. Terakhir pada *MainActivity.java* lengkapi kodenya sebagai berikut.

```
import Android.support.design.widget.TabLayout;
import Android.support.v4.view.ViewPager;
import Android.support.v7.app.AppCompatActivity;
import Android.os.Bundle;
import Android.support.v7.widget.Toolbar;
import
com.wanvysaputra.example.belajarfragment.adapter.TabFragmentP
agerAdapter;

public class MainActivity extends AppCompatActivity {
    //deklarasi semua komponen View yang akan digunakan
    private Toolbar toolbar;
    private ViewPager pager;
    private TabLayout tabs;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Activity_Main);

        //set up toolbar
        toolbar = (Toolbar)findViewById(R.id.tool_bar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("Material Tab");

        //inisialisasi tab dan pager
        pager = (ViewPager)findViewById(R.id.pager);
        tabs = (TabLayout)findViewById(R.id.tabs);

        //set object adapter kedalam ViewPager
        pager.setAdapter(new
        TabFragmentPagerAdapter(getSupportFragmentManager()));

        //Manipulasi sedikit untuk set TextColor pada Tab

        tabs.setTabTextColors(getResources().getColor(R.color.colorPr
        imaryDark),
        getResources().getColor(Android.R.color.white));
```

Bab 4 Fragments dan User Interface

```
//set tab ke ViewPager
tabs.setupWithViewPager (pager) ;

//konfigurasi Gravity Fill untuk Tab berada di posisi yang
proposional
tabs.setTabGravity(TabLayout.GRAVITY_FILL) ;
}
}
```

Gambar 69. Jawaban Soal *Fragments* (16)

20. Silakan jalankan dengan tombol *Run* ke *emulator* atau *device* atau *export* ke Apk untuk instal secara manual di *device*. Tampilan *apps* hasil implementasi membuat *fragment* di Android Studio adalah sebagai berikut.



Gambar 70. Jawaban Soal *Fragments* (17)

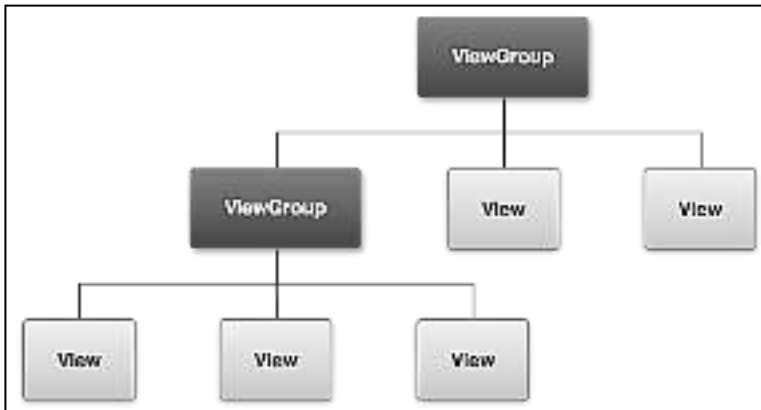
4.2. Teori *User Interface*

Semua elemen antarmuka pengguna (*user interface*) dalam aplikasi Android dibangun menggunakan objek *View* dan *ViewGroup*. *View* adalah objek yang menarik sesuatu di layar dan dapat berinteraksi dengan pengguna. *ViewGroup* merupakan sebuah objek yang menyimpan objek *View* lainnya (dan *ViewGroup*) untuk mendefinisikan *layout* antarmuka.

Bab 4 Fragments dan User Interface

Android menyediakan sekumpulan subkelas *View* dan *ViewGroup* yang menawarkan kontrol masukan umum (seperti *button* dan *layout*) serta berbagai model *layout* (seperti *layout* linier atau relatif).

Setiap kelompok *user interface* merupakan kontainer tersembunyi (*invisible*) yang mengelola tampilan *child*. Pohon hierarki pada gambar di bawah ini dapat menjadi sesuatu yang sederhana atau menjadi kompleks sesuai kebutuhan (namun yang sederhana merupakan paling baik untuk kinerja Android).



Gambar 71. Ilustrasi *User Interface*

Untuk mendeklarasikan *layout* dapat menyediakan objek *View* dalam kode dan mulai membangun *tree*, namun cara termudah dan terefektif untuk mendefinisikan *layout* adalah dengan file XML. XML menawarkan struktur *layout* yang dapat dibaca manusia, serupa dengan HTML.

4.2.1. Latihan Soal User Interface

Buatlah *dashboard* seperti tampilan gambar di bawah dengan menggunakan *Relative Layout*, *Frame Layout* ditambah dengan *GridLayout*.

Bab 4 Fragments dan User Interface



Gambar 72. Latihan Soal *User Interface*

4.2.2. Jawaban Soal *User Interface*

1. Buat *Project* Baru dengan nama aplikasi *dashboard*.
2. Tambahkan *Library*, Setelah *workspace* Android Studio terbuka pilih *build.gradle* (Module:app) tambahkan *script library gridlayout* seperti berikut, lalu klik *sync now*:

```
compile 'com.android.support:gridlayout-v7:24.2.1'
```

Gambar 73. Jawaban Soal *User Interface* (1)



Gambar 74. Jawaban Soal *User Interface* (2)

3. Tambahkan *icon* pada *drawable*. *Icon* yang saya gunakan diunduh dari www.flaticon.com.

Bab 4 Fragments dan User Interface



Gambar 75. Jawaban Soal *User Interface* (3)

4. Edit `Activity_Main.xml`, Jika sudah buka `Activity_Main.xml` lalu tambahkan kode-kode di bawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:columnCount="2"
    app:rowCount="2"
    app:orientation="horizontal"
    >
    <FrameLayout
        app:layout_columnWeight="1"
        app:layout_rowWeight="1"
        android:background="#5F9EA0"
        >
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:id="@+id/user"
                android:src="@drawable/user"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_marginTop="10dp"
```

Bab 4 Fragments dan User Interface

```
        Android:textSize="20sp"
        Android:textColor="#FFFFFF"
        Android:layout_height="wrap_content"
        Android:layout_below="@+id/user"
        Android:layout_centerHorizontal="true"
        Android:text="Profile"/>
    </RelativeLayout>
</FrameLayout>

<FrameLayout
app:layout_columnWeight="1"
app:layout_rowWeight="1"
Android:background="#D2691E">
    <RelativeLayout
        Android:layout_width="match_parent"
        Android:layout_height="match_parent">

        <ImageView
            Android:layout_width="wrap_content"
            Android:layout_height="wrap_content"
            Android:layout_centerInParent="true"
            Android:id="@+id/job"
            Android:src="@drawable/pin"/>

        <TextView
            Android:layout_width="wrap_content"
            Android:layout_marginTop="10dp"
            Android:textSize="20sp"
            Android:textColor="#FFFFFF"
            Android:layout_height="wrap_content"
            Android:layout_below="@+id/job"
            Android:layout_centerHorizontal="true"
            Android:text="Search Job"/>
    </RelativeLayout>
</FrameLayout>

<FrameLayout
app:layout_columnWeight="1"
app:layout_rowWeight="1"
Android:background="#CD5C5C">
    <RelativeLayout
        Android:layout_width="match_parent"
        Android:layout_height="match_parent">
        <ImageView
            Android:layout_width="wrap_content"
            Android:layout_height="wrap_content"
            Android:layout_centerInParent="true"
```


Bab 4 Fragments dan User Interface

```
        Android:id="@+id/notif"
        Android:src="@drawable/bell"/>
    <TextView
        Android:layout_width="wrap_content"
        Android:layout_marginTop="10dp"
        Android:textSize="20sp"
        Android:textColor="#FFFFFF"
        Android:layout_height="wrap_content"
        Android:layout_below="@+id/notif"
        Android:layout_centerHorizontal="true"
        Android:text="Notification"/>
    </RelativeLayout>
</FrameLayout>

<FrameLayout
    app:layout_columnWeight="1"
    app:layout_rowWeight="1"
    Android:background="#B8860B"
    >
    <RelativeLayout
        Android:layout_width="match_parent"
        Android:layout_height="match_parent">
        <ImageView
            Android:layout_width="wrap_content"
            Android:layout_height="wrap_content"
            Android:layout_centerInParent="true"
            Android:id="@+id/mail"
            Android:src="@drawable/mail"/>
        <TextView
            Android:layout_width="wrap_content"
            Android:layout_marginTop="10dp"
            Android:textColor="#FFFFFF"
            Android:textSize="20sp"
            Android:layout_height="wrap_content"
            Android:layout_below="@+id/mail"
            Android:layout_centerHorizontal="true"
            Android:text="Messages"/>
        </RelativeLayout>
    </FrameLayout>

<FrameLayout
    app:layout_columnWeight="1"
    app:layout_rowWeight="1"
    Android:background="#2E8B57"
    >
    <RelativeLayout
        Android:layout_width="match_parent"
```

Bab 4 Fragments dan User Interface

```
        Android:layout_height="match_parent">
<ImageView
    Android:layout_width="wrap_content"
    Android:layout_height="wrap_content"
    Android:layout_centerInParent="true"
    Android:id="@+id/rate"
    Android:src="@drawable/star"/>
<TextView
    Android:layout_width="wrap_content"
    Android:layout_marginTop="10dp"
    Android:textColor="#FFFFFF"
    Android:textSize="20sp"
    Android:layout_height="wrap_content"
    Android:layout_below="@+id/rate"
    Android:layout_centerHorizontal="true"
    Android:text="Rate Card"/>
</RelativeLayout>
</FrameLayout>
<FrameLayout
    app:layout_columnWeight="1"
    app:layout_rowWeight="1"
    Android:background="#BC8F8F">
    <RelativeLayout
        Android:layout_width="match_parent"
        Android:layout_height="match_parent">
        <ImageView
            Android:layout_width="wrap_content"
            Android:layout_height="wrap_content"
            Android:layout_centerInParent="true"
            Android:id="@+id/rev"
            Android:src="@drawable/like"/>
        <TextView
            Android:layout_width="wrap_content"
            Android:layout_marginTop="10dp"
            Android:textSize="20sp"
            Android:layout_height="wrap_content"
            Android:layout_below="@+id/rev"
            Android:layout_centerHorizontal="true"
            Android:textColor="#FFFFFF"
            Android:text="Review "/>
        </RelativeLayout>
    </FrameLayout>
</Android.support.v7.widget.GridLayout>
```

Gambar 76. Jawaban Soal *User Interface* (4)

Bab 4 Fragments dan User Interface

5. *Edit string.xml* ubah *app-name* menjadi *home*:

```
<resources>  
<string name="app_name">Home</string>  
</resources>
```

Gambar 77. Jawaban Soal *User Interface* (5)

6. *Edit color.xml* ubah warna utama (*primary*):

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
<color name="colorPrimary">#4682B4</color>  
<color name="colorPrimaryDark">#303F9F</color>  
<color name="colorAccent">#FF4081</color>  
</resources>
```

Gambar 78. Jawaban Soal *User Interface* (6)

7. Jika sudah diubah semua *Activity* di atas, jalankan aplikasi dan akan tampil seperti di bawah ini.



Gambar 79. Jawaban Soal *User Interface* (7)

BAB 5

DATA PERSISTENCE

Capaian Pembelajaran

1. Mampu mengetahui data *persistence* dan SQLite pada Android.
2. Mampu membuat aplikasi menggunakan data *persistence* dan SQLite.
3. Mampu menjalankan aplikasi menggunakan data *persistence* SQLite.

5.1. Teori Data Persistence

Data menurut masa hidup (daur hidup) dalam aplikasi komputer dapat dibedakan menjadi 2 jenis yaitu *persistent* data dan *non-persistent* data. *Persistent* data adalah data yang dapat digunakan kembali jika komputer di mana aplikasi kembali dinyalakan. Sedangkan *non-persistent* data adalah kebalikannya. Masa hidup sebuah *non-persistent* data hanya selama aplikasi tersebut digunakan. Sebagaimana komputer jika suatu aplikasi *down*, maka data tersebut akan hilang.

Terdapat 4 dasar yang dapat dilakukan sebagai layanan data, yakni menggunakan *shared preferences* untuk menyimpan paket data berukuran kecil, menggunakan *file systems*, menggunakan DBMS (*database management system*) seperti *database* SQLite di *static resources*, dan menggunakan Android *backup service* sebagai keamanan data.

5.1.1. Shared Preferences

Shared Preferences adalah salah satu *storage* di Android. Karakter penyimpanan ini adalah *key-value storage*, Sehingga hanya bisa menyimpan data bertipe primitif (*float, int, long, string, boolean*). Karenanya penyimpanan ini tergolong sangat minimalis dan sangat cocok untuk penyimpanan data yang sedikit. *Shared Preferences object* paling

cocok digunakan untuk menyimpan data yang berupa pasangan *name/value* seperti *userID*, tanggal lahir, jenis kelamin dan sebagainya.

5.1.2. File System

Untuk menyimpan data dapat juga menggunakan *file system*. Pada Android, dapat digunakan “*java.io*” *class* untuk melakukan penyimpanan dan pengolahan data dengan *file system* ini, yakni *internal storage* (tidak dapat dipindahkan) dan *external storage* (misalnya, melalui SD Card).

5.1.3. Static Resources

Selain dengan membuat dan menggunakan *files* secara dinamis selama *runtime*, anda juga dapat menambah *files* ke dalam *package project* anda pada waktu *design* atau perancangan program masih dilakukan. Caranya dapat dengan menambah *file* ke dalam folder *res/raw*.

5.1.4. Android Backup Service

Android menyediakan layanan untuk membantu proses *backup* aplikasi *user* dan *system settings* melalui *cloud*. Hal ini memungkinkan aplikasi untuk berpindah ke perangkat baru *user*, tetapi data aplikasi tidak langsung dibawa oleh proses ini secara *default*. *Developers* harus meakukan pendaftaran (*sign up*) *Android Backup Service* secara eksplisit melalui tautan <http://developer.Android.com/google/backup/signup.html> dan menyediakan penerapan *Android.app.backup.BackupAgent* dalam aplikasinya untuk berinteraksi dengan *Android Backup Service*.

5.1.5. Latihan Soal Data Persistence

Buatlah aplikasi menggunakan *SharedPreferences* seperti di bawah ini.



Gambar 80. Latihan Soal *Data Persistence*

5.1.6. Jawaban Soal *Data Persistence*

1. Buka aplikasi Android Studio, buat *Project* baru.
2. Buka *layout* *Activity_Main.xml* kalian, di sini kita akan menambahkan *Edit text* (untuk *Input*), *TextView* (untuk *Output Data*) dan *Button* (untuk mengeksekusi program).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="8dp"
android:orientation="vertical"
android:background="#FFF5F5F5"

tools:context="Android.cianjur.developer.net.basicAndroid.Mai
nActivity">

    <EditText
    android:id="@+id/input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Masukan Data"
    android:inputType="textPersonName" />

    <Button
    android:id="@+id/save"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Simpan" />

    <TextView
    android:id="@+id/output"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:gravity="center"
    android:text="Output Data:"
    android:textStyle="bold" />
</LinearLayout>
```

Gambar 81. Jawaban Soal *Data Persistence* (1)

Bab 5 Data Persistence

3. Selanjutnya buka file `MainActivity.java`, pada *sourcecode* berikut ini kita akan menambahkan *SharedPreferences*, serta beberapa fungsi / *method*-nya.

```
package Android.developer.net.basicAndroid;
import Android.content.Context;
import Android.content.SharedPreferences;
import Android.support.v7.app.AppCompatActivity;
import Android.os.Bundle;
import Android.view.View;
import Android.widget.Button;
import Android.widget.EditText;
import Android.widget.TextView;
import Android.widget.Toast;

public class MainActivity extends AppCompatActivity{

    //Deklarasi Variable Pendukung
    private TextView Hasil;
    private EditText Masukan;
    private Button Eksekusi;

    //Deklarasi dan Inisialisasi SharedPreferences
    private SharedPreferences preferences;

    //Digunakan Untuk Konfigurasi SharedPreferences
    private SharedPreferences.Editor editor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Activity_Main);
        Masukan = findViewById(R.id.input);
        Hasil = findViewById(R.id.output);
        Eksekusi = findViewById(R.id.save);

        //Membuat File Baru Beserta Modifiernya
        Preferences =
            getSharedPreferences("Belajar_SharedPreferences",
                Context.MODE_PRIVATE);

        Eksekusi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getData();
            }
        });
    }
}
```

Bab 5 Data Persistence

```
Toast.makeText(getApplicationContext(), "Data Tersimpan",
    Toast.LENGTH_SHORT).show();
}
});
}

private void getData(){
//Mendapatkan Input dari user
String getKonten = Masukan.getText().toString();
//Digunakan Untuk Pengaturan Konfigurasi SharedPreferences
editor = preferences.edit();
//Memasukan Data Pada Editor SharedPreferences dengan key
(data_saya)
editor.putString("data_saya", getKonten);
//Menjalankan Operasi
editor.apply();
//Menampilkan Output
Hasil.setText("Output Data:
"+preferences.getString("data_saya", null));
}
}
```

Gambar 82. Jawaban Soal Data Persistence (2)

Pertama mendeklarasikan beberapa variabel pendukung, seperti *TextView*, *Edittext* dan *Button*, lalu kita juga menginisialisasi *SharedPreferences* dan juga *SharedPreferences*, *Editor* yang digunakan konfigurasi data pada *SharedPreferences*.

Fungsi *getSharedPreferences()* gunakan jika kita ingin membuat *file SharedPreferences* yang baru dengan parameter yang spesifik dan juga nama *file*-nya spesifik dan gunakan *Context* untuk mengakses *file* pada aplikasi.

Didalam *method getSharedPreferences()* *file* yang akan dibuat mempunyai mode privat, yang artinya, data pada *file* tersebut hanya untuk aplikasi itu saja. Untuk memasukan data dapat menggunakan fungsi “*putString()*,” dengan parameter *key* dan nilai. Dan untuk menampilkan *output*, kita dapat menggunakan fungsi “*getString()*,” untuk menampilkan data tersebut dapat dengan memanggil *key*-nya saya.

5.2. Teori SQLite Database

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* (bersifat gratis penggunaannya) yang dikerjakan oleh D. Richard Hipp. Tipe data yang didukung SQLite ialah *Numeric (integer, float, double)*, *Text (char, varchar, text)*, *DateTime*, dan *BLOB (Binary Large Object)*.

Jadi dapat dikatakan, SQLite ini adalah mesin database SQL yang tertanam pada sistem yang kita gunakan. Tidak seperti pada paradigma *client-server* umumnya, *Core SQLite* bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman.

Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah *file*. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

Database SQLite merupakan solusi penyimpanan yang baik jika anda memiliki data terstruktur yang perlu diakses dan disimpan secara persisten serta sering ditelusuri dan diubah. SQLite juga bisa digunakan sebagai media penyimpanan utama untuk data aplikasi atau pengguna, atau juga bisa menggunakannya untuk proses *caching* serta menyediakan data yang diambil dari *cloud*. Ada dua tipe data yang dikaitkan secara khusus dengan penggunaan database SQLite, yaitu *Cursor* dan *ContentValues*.

5.2.1. Cursor SQLite

SQLiteDatabase selalu menyajikan hasil berupa *cursor* dalam format tabel yang menyerupai *database SQL*. Anda bisa menganggap data sebagai larik baris dan *cursor* adalah *pointer* ke dalam satu baris data

terstruktur. Kelas *cursor* menyediakan metode untuk menggerakkan *cursor* melalui struktur data, dan metode untuk mendapatkan data dari bidang-bidang setiap baris.

Kelas *cursor* memiliki sejumlah subkelas yang mengimplementasikan *cursor* untuk tipe data tertentu, yaitu:

- *SQLiteCursor* untuk mengekspos hasil *query* dari sebuah *SQLiteDatabase*. *SQLiteCursor* tidak disinkronkan secara internal, sehingga kode yang menggunakan *SQLiteCursor* dari beberapa *thread* harus melakukan sinkronisasi sendiri saat menggunakan *SQLiteCursor*.
- *MatrixCursor* adalah implementasi *cursor* lengkap dan tidak tetap, yang didukung oleh larik objek yang secara otomatis meluaskan kapasitas internal bila diperlukan.

Beberapa operasi umum yang ada pada *cursor* adalah:

- *getCount()*, mengembalikan jumlah baris dalam *cursor*.
- *getColumnNames()*, mengembalikan larik *string* yang berisi nama semua kolom dalam rangkaian hasil dalam urutan pencantumannya dalam hasil/*result*.
- *getPosition()*, mengembalikan posisi *cursor* saat ini dalam rangkaian baris.
- Getter tersedia untuk tipe data tertentu, seperti *getString(int column)* dan *getInt(int column)*.
- Operasi seperti *moveToFirst()* dan *moveToNext()* akan menggerakkan *cursor*.
- *close()*, membebaskan semua sumber daya dan membuat *cursor* menjadi tidak valid.

Jika panggilan metode mengembalikan ulangi pada hasil, ekstrak data, lakukan sesuatu dengan data, dan terakhir harus menutup *cursor* untuk membebaskan memori. Jika tidak dilakukan, maka aplikasi bisa saja terhenti (*freeze*) saat kehabisan memori.

Cursor dimulai sebelum baris hasil pertama, sehingga pada pengulangan pertama gerakkan *cursor* ke hasil pertama jika ada. Jika *cursor* kosong, atau baris terakhir sudah diproses, maka akan keluar dari

loop (perulangan). Jangan lupa untuk menutup *cursor* bila telah selesai menggunakannya. (Ini tidak boleh diulang terlalu sering)

```
// Perform a query and store the result in a Cursor
Cursor cursor = db.rawQuery(...);
try {
    while (cursor.moveToNext()) {
        // Do something with the data
    }
} finally {
    cursor.close();
}
```

Gambar 83. Penggunaan *Cursor*

Jika menggunakan *database SQL*, anda bisa mengimplementasikan kelas *SQLiteOpenHelper* untuk mengembalikan *cursor* ke aktivitas pemanggil atau adapter, atau anda bisa mengonversi data ke format yang lebih cocok untuk adapter. Manfaat dari yang terakhir itu adalah pengelolaan *cursor* (dan penutupannya) ditangani oleh *helper* terbuka, dan antarmuka pengguna anda tidak tergantung pada apa yang terjadi di *backend*.

5.2.2. ContentValues SQLite

Serupa dengan cara menyimpan data lainnya, *instance ContentValues* menyimpan data sebagai pasangan nilai kunci, dalam ini kuncinya adalah nama kolom dan nilainya untuk *cell*. Satu *instance ContentValues* menyatakan satu baris tabel. Metode *insert()* untuk *database* memerlukan nilai untuk mengisi baris yang diteruskan sebagai *instance ContentValues*.

```
ContentValues values = new ContentValues();
// Insert one row. Use a loop to insert multiple rows.
values.put(KEY_WORD, "Android");
values.put(KEY_DEFINITION, "Mobile operating system.");
db.insert(WORD_LIST_TABLE, null, values);
```

Gambar 84. Penggunaan *ContentValues*

5.2.3. Mengimplementasikan Database SQLite

Untuk mengimplementasikan SQLite diperlukan cara sebagai berikut:

1. Buat model data.

Praktik yang baik adalah dengan membuat kelas yang menyatakan data Anda dengan *getter* dan *setter*. Untuk *database* SQLite, *instance* kelas ini dapat menyatakan satu catatan, dan untuk *database* sederhana, satu baris dalam tabel.

```
public class WordItem {
    private int mId;
    private String mWord;
    private String mDefinition; // Getters and setters and more
}
```

Gambar 85. Model Data

2. Jadikan SQLiteOpenHelper sebagai *sub-class*

Open helper apa pun yang di buat harus meng-*extends* *SQLiteOpenHelper*.

```
public class WordListOpenHelper extends SQLiteOpenHelper {
    public WordListOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        Log.d(TAG, "Construct WordListOpenHelper");
    }
}
```

Gambar 86. SQLiteOpenHelper

3. Gunakan konstanta untuk nama tabel dan *query* pembuatan *database*

Mendeklarasikan nama tabel, kolom, dan baris sebagai konstanta. Hal ini akan membuat kode lebih terbaca, dan lebih mudah saat mengubah nama, dan *query* akan terlihat lebih mirip dengan SQL. Penulisan bisa di *class open helper*, atau dalam *class* publik tersendiri.

```
private static final int DATABASE_VERSION = 1;
// has to be 1 first time or app will crash
private static final String WORD_LIST_TABLE = "word_entries";
```

Bab 5 Data Persistence

```
private static final String DATABASE_NAME = "wordlist";//  
Column names...  
public static final String KEY_ID = "_id";  
public static final String KEY_WORD = "word";// ... and a  
string array of columns.  
private static final String[] COLUMNS = {KEY_ID, KEY_WORD};
```

Gambar 87. Penggunaan Konstanta

Diperlukan *query* untuk membuat tabel di *database*. Ini biasanya juga didefinisikan sebagai konstanta *string*. Contoh dasar ini membuat satu tabel dengan satu kolom untuk ID bertambah-otomatis/*auto increment* dan kolom untuk menampung kata.

```
private static final String WORD_LIST_TABLE_CREATE =  
"CREATE TABLE " + WORD_LIST_TABLE + " (" +  
KEY_ID + " INTEGER PRIMARY KEY, " +  
// will auto-increment if no value passed  
KEY_WORD + " TEXT );";
```

Gambar 88. Penggunaan *Query*

4. Implementasikan `onCreate()` untuk membuat `SQLiteDatabase` bersama tabel untuk data anda

Metode `onCreate()` hanya dipanggil jika tidak ada *database*. Buat tabel Anda dalam metode, dan boleh menambahkan data awal.

```
@Override  
public void onCreate(SQLiteDatabase db) { // Creates new  
database  
db.execSQL(WORD_LIST_TABLE_CREATE); // Create the tables  
fillDatabaseWithData(db); // Add initial data  
// Cannot initialize mWritableDatabase and mReadableDB here,  
because  
// this creates an infinite loop of on Create()  
// being repeatedly called.  
}
```

Gambar 89. Penggunaan *onCreate*

5. Implementasikan onUpgrade()

Ini adalah metode yang diperlukan. Jika *database* hanya berfungsi sebagai cache untuk data yang juga disimpan *online*, juga dapat menghapus tabel dan membuat ulang setelah peningkatan versi selesai.

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    // SAVE USER DATA FIRST!!!
    Log.w(WordListOpenHelper.class.getName(),
    "Upgrading database from version " + oldVersion + " to "
+ newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS " + WORD_LIST_TABLE);
    onCreate(db);
}
```

Gambar 90. Penggunaan onUpgrade

6. Implementasikan metode opsional

Class open helper menyediakan metode tambahan yang bisa diganti sesuai keperluan.

- onDowngrade(), Implementasi *default* menolak penurunan versi.
- onConfigure(), dipanggil sebelum onCreate. Gunakan ini hanya untuk memanggil metode yang mengonfigurasi parameter koneksi database.
- onOpen(), pekerjaan apa pun selain konfigurasi yang harus dilakukan sebelum *database* dibuka.

7. Implementasikan metode query(), insert(), delete(), update(), count() dalam SQLiteOpenHelper

- Query()

```
public WordItem query(int position) {
    String query = "SELECT * FROM " + WORD_LIST_TABLE +
    " ORDER BY " + KEY_WORD + " ASC " +
    "LIMIT " + position + ",1"; Cursor cursor = null;
    WordItem entry = new WordItem(); try {
    if (mReadableDB == null) {mReadableDB =
```

Bab 5 Data Persistence

```
getReadableDatabase();}
    cursor = mReadableDB.rawQuery(query, null);
    cursor.moveToFirst();
    entry.setId(cursor.getInt(cursor.getColumnIndex(KEY_ID)));
    entry.setWord(cursor.getString(cursor
        .getColumnIndex(KEY_WORD)));
    } catch (Exception e) {
    Log.d(TAG, "EXCEPTION! " + e);
    } finally {
    // Must close cursor and db now that we are done with it.
    cursor.close();
    return entry;
    }
}
```

Gambar 91. Penggunaan Metode Query()

- Insert()

```
newId = mWritableDatabase.insert(WORD_LIST_TABLE, null, values);
```

Gambar 92. Penggunaan Metode Insert()

- Delete()

```
deleted = mWritableDatabase.delete(WORD_LIST_TABLE,
    KEY_ID + " = ?", new String[]{String.valueOf(id)});
```

Gambar 93. Penggunaan Metode Delete()

- Update()

```
ContentValues values = new ContentValues();
values.put(KEY_WORD, word);
mNumberOfRowsUpdated = mWritableDatabase.update(WORD_LIST_TABLE,
    values, // new values to insert
    KEY_ID + " = ?",
    new String[]{String.valueOf(id)});
```

Gambar 94. Penggunaan Metode Update()

- Count()

```
@Override
public int getItemCount() {
    return (int) mDB.count();
}
Di open helper:
public long count(){
if (mReadableDB == null) {mReadableDB =
getReadableDatabase();}
return DatabaseUtils.queryNumEntries(mReadableDB,
WORD_LIST_TABLE);
}
```

Gambar 95. Penggunaan Metode Count()

8. Dalam MainActivity, buat *instance* SQLiteOpenHelper Untuk menangani *database*, dalam MainActivity, di onCreate, bisa memanggil

```
mDB = new WordListOpenHelper(this);
```

Gambar 96. Penggunaan Instance SQLiteOpenHelper

9. Panggil metode SQLiteOpenHelper untuk digunakan bersama *database*

Adapun hal yang perlu diperhatikan saat implementasi SQLite sebagai berikut:

- Bila mengimplementasikan metode, selalu masukan operasi *database* ke dalam blok *try/catch*.
- Aplikasi prototipe pasti tidak memvalidasi data pengguna. Bila membuat aplikasi untuk dipublikasikan ke Android *market*, selalu pastikan data pengguna untuk menghindari SQL *injection* yang berbahaya ke dalam *database* Anda.
- Kirimkan data bersama APK sebagai sumber daya, dan bangun *database* saat pengguna membuka aplikasi untuk pertama kali.

Bab 5 Data Persistence

Misalnya, dalam format CSV. Selanjutnya bisa membaca data dengan aliran masukan dan menambahkannya ke *database*.

- Bangun dan isi dahulu *database* SQLite lalu sertakan dalam APK. Dengan metode ini, tulis aplikasi yang akan membuat dan mengisi *database*. Hal tersebut bisa dilakukan pada emulator. Selanjutnya bisa menyalin *file* tersebut di tempat penyimpanan *database* (direktori “/data/data/YOUR_PACKAGE/databases/”) dan menyertakannya sebagai aset bersama aplikasi. Saat aplikasi dimulai untuk pertama kali, salin kembali *file database* ke dalam direktori “/data/data/YOUR_PACKAGE/databases/”.

BAB 6

ANDROID SENSOR

Capaian Pembelajaran

1. Mampu mengetahui Android sensor pada Android.
2. Mampu membuat aplikasi menggunakan Android sensor.
3. Mampu menjalankan aplikasi menggunakan Android sensor.

6.1. Teori Android Sensor

Sebagian besar perangkat Android memiliki sensor *built-in* yang mengukur gerakan, orientasi, dan berbagai kondisi lingkungan. Sensor ini mampu menyediakan data mentah dengan presisi dan akurasi tinggi, dan berguna jika Anda ingin memantau *positioning* atau pergerakan tiga dimensi perangkat, atau memantau perubahan di lingkungan sekitar di dekat perangkat. Misalnya, suatu game mungkin akan melacak pembacaan dari sensor gravitasi perangkat untuk menyimpulkan gerakan dan gestur pengguna yang kompleks, seperti kemiringan, goyangan, rotasi, atau ayunan. Demikian juga, aplikasi cuaca mungkin menggunakan sensor suhu dan sensor kelembapan perangkat untuk menghitung dan melaporkan titik embun, atau suatu aplikasi perjalanan mungkin menggunakan sensor medan geomagnetik dan akselerometer untuk melaporkan arah kompas.

Framework sensor Android memungkinkan Anda mengakses berbagai jenis sensor. Beberapa sensor ini berbasis *hardware* dan beberapa berbasis *software*. Sensor berbasis *hardware* adalah komponen fisik yang di-*build* pada perangkat *handset* atau tablet. Sensor tersebut mengambil data dengan mengukur langsung properti lingkungan spesifik, seperti akselerasi, kekuatan medan geomagnetik, atau perubahan sudut. Sensor berbasis *software* bukanlah perangkat fisik, meskipun perangkat tersebut meniru sensor berbasis *hardware*. Sensor berbasis *software* mengambil data dari satu atau beberapa sensor berbasis *hardware* dan terkadang

Bab 6 Android Sensor

disebut sensor virtual atau sensor sintetis. Sensor akselerasi linier dan sensor gravitasi adalah contoh sensor berbasis *software*.

Hanya sedikit perangkat Android yang memiliki semua jenis sensor. Contohnya, sebagian besar perangkat *handset* dan tablet memiliki akselerometer dan magnetometer, namun sedikit perangkat yang memiliki barometer atau termometer. Selain itu, perangkat dapat memiliki lebih dari satu sensor dari jenis tertentu. Contohnya, suatu perangkat dapat memiliki dua sensor gravitasi, masing-masing dengan rentang yang berbeda.

Tabel 2. Jenis Sensor yang Didukung

Sensor	Jenis	Deskripsi
TYPE_ACCELEROMETER	<i>Hardware</i>	Mengukur gaya akselerasi dalam m/s^2 yang diterapkan di perangkat pada ketiga sumbu fisik (x, y, dan z), termasuk gaya gravitasi.
TYPE_AMBIENT_TEMPERATURE	<i>Hardware</i>	Mengukur suhu ruangan sekitar dalam derajat Celcius ($^{\circ}C$). Lihat catatan di bawah.
TYPE_GRAVITY	<i>software</i> atau <i>Hardware</i>	Mengukur gaya gravitasi dalam m/s^2 yang diterapkan di perangkat pada ketiga sumbu fisik (x, y, z).
TYPE_GYROSCOPE	<i>Hardware</i>	Mengukur tingkat rotasi perangkat dalam rad/s di sekitar setiap tiga sumbu fisik (x, y, dan z).
TYPE_LIGHT	<i>Hardware</i>	Mengukur level cahaya sekitar (pencahayaannya) dalam lx.
TYPE_LINEAR_ACCELERATION	<i>software</i> atau <i>Hardware</i>	Mengukur gaya akselerasi dalam m/s^2 yang diterapkan di perangkat pada ketiga sumbu fisik (x, y, dan z), tidak termasuk gaya gravitasi.
TYPE_MAGNETIC_FIELD	<i>Hardware</i>	Mengukur medan geomagnetik sekitar untuk ketiga sumbu fisik (x, y, z) dalam μT .
TYPE_ORIENTATION	<i>software</i>	Mengukur derajat rotasi yang dibuat perangkat di sekitar ketiga sumbu fisik (x, y, z). Pada API level 3, Anda dapat memperoleh matriks kemiringan dan matriks rotasi untuk perangkat menggunakan sensor gravitasi dan sensor medan geomagnetik bersama dengan metode <code>getRotationMatrix()</code> .

Bab 6 Android Sensor

Sensor	Jenis	Deskripsi
TYPE_PRESSURE	Hardware	Mengukur tekanan udara sekitar dalam hPa atau mbar.
TYPE_PROXIMITY	Hardware	Mengukur kedekatan objek dalam cm sesuai dengan layar tampilan perangkat. Sensor ini biasanya digunakan untuk menentukan apakah suatu <i>handset</i> ditahan di telinga seseorang.
TYPE_RELATIVE_HUMIDITY	Hardware	Mengukur kelembapan udara relatif dalam persen (%).
TYPE_ROTATION_VECTOR	software atau Hardware	Mengukur orientasi perangkat dengan memberikan tiga elemen vektor rotasi perangkat.
TYPE_TEMPERATURE	Hardware	Mengukur suhu perangkat dalam derajat Celcius ($^{\circ}\text{C}$). Penerapan sensor ini bervariasi antarperangkat dan sensor ini diganti dengan sensor TYPE_AMBIENT_TEMPERATURE di API Level 14

6.2. Latihan Soal Android Sensor

Buatlah aplikasi kompas pada Android seperti gambar di bawah. Untuk antarmuka aplikasi, gunakan TextView dan widget ImageView.



Gambar 97. Latihan Soal Android Sensor

6.3. Jawaban Soal Android Sensor

Untuk TextView yang menunjukkan berapa nilai *heading* dari kompas sementara ImageView berperan sebagai arah navigasi kompas dengan menggunakan sebuah gambar navigasi.

1. Langkah pertama membuat *file* Activity_Main.xml dalam membangun antarmuka aplikasi kompas

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#141414"
android:foregroundGravity="center|center_horizontal"
android:focusableInTouchMode="false">

<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/background_light"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignTop="@+id/tvHeading"
android:layout_marginTop="-62dp" />

<ImageView
android:layout_width="wrap_content"
android:layout_height="120dp"
android:src="@drawable/prompt"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignTop="@+id/tvHeading"
android:layout_marginTop="-62dp"
android:id="@+id/imageView" />

<TextView
android:id="@+id/tvHeading"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_marginBottom="40dp"
android:layout_marginTop="40dp"
android:text="Heading 0"
android:textColor="#ffffff"
android:textSize="28dp" />
```

Bab 6 Android Sensor

```
<ImageView
Android:layout_width="wrap_content"
Android:layout_height="120dp"
Android:src="@drawable/background_compass"
Android:layout_alignParentLeft="true"
Android:layout_alignParentStart="true"
Android:layout_alignTop="@+id/tvHeading"
Android:layout_alignParentBottom="true"
Android:layout_marginTop="-39dp"
Android:layout_marginRight="15dp"
Android:layout_marginLeft="15dp" />

<ImageView
Android:id="@+id/imageViewCompass"
Android:layout_width="wrap_content"
Android:layout_height="wrap_content"
Android:src="@drawable/compass"
Android:layout_centerVertical="true"
Android:layout_alignParentLeft="true"
Android:layout_alignParentStart="true"
Android:layout_marginRight="17dp"
Android:layout_marginLeft="17dp" />

<ImageView
Android:layout_width="wrap_content"
Android:layout_height="120dp"
Android:src="@drawable/miui_cover"
Android:layout_alignParentLeft="true"
Android:layout_alignParentStart="true"
Android:layout_alignTop="@+id/tvHeading"
Android:layout_alignParentBottom="true"
Android:layout_marginTop="-40dp"
Android:layout_marginRight="20dp"
Android:layout_marginLeft="20dp" />

<TextView
Android:layout_width="wrap_content"
Android:layout_height="wrap_content"
Android:textAppearance="?Android:attr/textAppearanceMedium"
Android:text="- App by - \nRifqi Fauzi Rahmadzani"
Android:id="@+id/textView"
Android:textAlignment="center"
Android:layout_alignParentBottom="true"
Android:layout_centerHorizontal="true"
Android:layout_marginBottom="20dp"
Android:textColor="#acacac"
```

Bab 6 Android Sensor

```
Android:textSize="14dp" />
</RelativeLayout>
```

Gambar 98. Jawaban Soal Android Sensor (1)

2. Langkah berikutnya membuat file MainActivity.java sesuaikan nama package dengan nama aplikasi yang anda buat

```
package com.wanvysaputra.compass;

import Android.app.Activity;
import Android.Hardware.Sensor;
import Android.Hardware.SensorEvent;
import Android.Hardware.SensorEventListener;
import Android.Hardware.SensorManager;
import Android.os.Bundle;
import Android.view.animation.Animation;
import Android.view.animation.RotateAnimation;
import Android.widget.ImageView;
import Android.widget.TextView;

public class MainActivity extends Activity implements
SensorEventListener {

// define the display assembly compass picture
private ImageView image;

// record the compass picture angle turned
private float currentDegree = 0f;

// device sensor manager
private SensorManager mSensorManager;

TextView tvHeading;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.Activity_Main);

// our compass image
image = (ImageView) findViewById(R.id.imageViewCompass);

// TextView that will tell the user what degree is he heading
tvHeading = (TextView) findViewById(R.id.tvHeading);
```

Bab 6 Android Sensor

```
// initialize your Android device sensor capabilities
mSensorManager = (SensorManager)
getSystemService (SENSOR_SERVICE);
}

@Override
protected void onResume() {
super.onResume();

// for the system's orientation sensor registered listeners
mSensorManager.registerListener(this,
mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),
SensorManager.SENSOR_DELAY_GAME);
}

@Override
protected void onPause() {
super.onPause();

// to stop the listener and save battery
mSensorManager.unregisterListener(this);
}

@Override
public void onSensorChanged(SensorEvent event) {

// get the angle around the z-axis rotated
float degree = Math.round(event.values[0]);

tvHeading.setText("Heading " + Float.toString(degree) + "°");

// create a rotation animation (reverse turn degree degrees)
RotateAnimation ra = new RotateAnimation(
currentDegree,
-degree,
Animation.RELATIVE_TO_SELF, 0.5f,
Animation.RELATIVE_TO_SELF, 0.5f);

// how long the animation will take place
ra.setDuration(60);

// set the animation after the end of the reservation status
ra.setFillAfter(true);

// Start the animation
image.startAnimation(ra);
currentDegree = -degree;
```


Bab 6 Android Sensor

```
}  
  
@Override  
public void onAccuracyChanged(Sensor sensor, int accuracy) {  
    // not in use  
}  
}
```

Gambar 99. Jawaban Soal Android Sensor (2)

3. Terakhir agar tampilan *screen* aplikasi tetap dalam kondisi *portrait* aktifkan fungsi orientasi *screen* pada *file* AndroidManifest.xml edit seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.wanvysaputra.compass">  
  
    <application  
        android:allowBackup="true"  
        android:icon="@drawable/app_icon"  
        android:label="@string/app_name"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme">  
        <activity android:name=".MainActivity"  
            android:screenOrientation="portrait">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
  
</manifest>
```

Gambar 100. Jawaban Soal Android Sensor (3)

Setelah selesai semuanya coba jalankan aplikasinya melalui *emulator* ataupun *USB debugging*.

Glosarium

Activity (activities): suatu modul fungsionalitas aplikasi yang biasanya berhubungan langsung dengan suatu *layer user interface* beserta fungsionalitas di dalamnya.

Broadcast Intents: suatu *Intent* sistem yang dikirim ke seluruh aplikasi yang telah mendaftar pada suatu *Broadcast Receiver* yang diinginkan.

Broadcast receiver: mekanisme di mana aplikasi dapat memberikan tanggapan atas *Broadcast Intents* yang diterimanya.

Contact providers: Suatu mekanisme untuk berbagi (*sharing*) data di antara aplikasi Android

File manifest (AndroidManifest.xml): perekat yang mengumpulkan beragam elemen yang membentuk suatu aplikasi Android

Filtering: proses penyaringan data yang membatasi hasil *query* sesuai dengan kriteria yang dicari

Fragments: *mini-activities* yang dapat di tambahkan atau dihilangkan dari sebuah *Activity*.

GridView: *view* yang menampilkan *items dalam bentuk grid* dua dimensi yang dapat di *scroll*.

Glosarium

Intents: mekanisme di mana satu *Activity* dapat menjalankan *Activity* lainnya dan menerapkan alur proses yang dibutuhkan dalam aplikasi.

List views: *views* yang memungkinkan *programmer* untuk menampilkan suatu daftar panjang *items*.

Projecting: proses penentuan berapa banyak dan apa saja kolom yang dikembalikan dalam sebuah *query*.

Services: proses-proses yang berjalan secara *background* dan tidak memiliki *user interface*

Sorting: proses pengurutan data untuk hasil *query* yang akan diakses dan ditampilkan nantinya.

SpinnerView: *view* yang menampilkan satu *item* pada satu dari suatu daftar dan memungkinkan *users* untuk memilih *items* tersebut.

SQLite: suatu *embedded RDBMS (Relational Database Management System)* karena disediakan dalam bentuk *library* yang dihubungkan ke suatu aplikasi secara langsung dan digunakan sebagai *database* internal aplikasi Android.

View: sebuah *widget* yang memiliki tampilan dalam layar aplikasi

ViewGroup: kerangka *layout* di mana *programmer* dapat mengatur tampilan dan urutan dari *views*

Daftar Pustaka

- Cinar, Onur. 2015 *Android: Quick APIs Reference*. California: Apress.
- Craig, Clifton and Adam Gerber. 2015. *Learn Android Studio: Build Android Apps Quickly and Effectively*. California: Apress
- Guntoro. 2018. *Menguasai 20 Android Project Simple Example dalam 10 Hari*. Badoy Studio. Diakses 29 Juni 2020, dari BadoyStudio
- Hansun, Seng dan Mascel Bonar Kristanda dan Michael Wijaya Saputa. 2018. *Pemrograman Android dengan Android Studio ide*. Yogyakarta: Penerbit Andi
- Smith, Dave and Erik Hellman. 2015. *Android Recipes: A Problem=Solution Approach*. California: Apress.
- Smith, Dave and Jeff Friesen. 2014. *Android Recipes: A Problem=Solution Approach*. California: Apress.
- Smyth, Neil. 2015. *Android Studio Development Essentials*. eBookFrenzy.
- Wei-Meng, Lee. 2012. *Beginning Android 4 Application Development*. USA: John Willey & Sons.

Pemrograman Berbasis Objek

Pemrograman Mobile dengan Android Studio

Wanvy Arifha Saputra

Puji Syukur kehadiran Allah Subhanahu wa Ta'ala, karena atas limpahan rahmat-Nya, sehingga kami dapat menyelesaikan bahan ajar Pemrograman Berbasis Objek – Pemrograman Mobile dengan Android Studio. Bahan ajar ini disusun berdasarkan topik tingkatan pemahaman. Bahan ajar ini juga dilengkapi dengan latihan soal untuk menguji pemahaman mahasiswa terkait dengan materi.

Capaian Pembelajaran:

1. Mampu melakukan instalasi Android Studio
2. Mampu menjalankan Android Studio
3. Mampu mengetahui konsep dasar sistem operasi Android



Politeknik Elektronika Negeri
Semarang
Politeknik Elektronika Negeri Semarang, Jl. Sepren 61, Kecamatan Gunung
Sempu, Kota Semarang 50232, Jawa Tengah, Indonesia
Telp : 0291-7663000
Email : pens@pens.ac.id

ISBN 978-602-70448-3-0



9 706237 684201