

BUKU PANDUAN APLIKASI SMART GUDANG BERBASIS SCANNER DENGAN ANDROID STUDIO

BUKU PANDUAN APLIKASI SMART GUDANG BERBASIS SCANNER DENGAN ANDROID STUDIO

Dalam 24 Jam

Etika Khusnul Laeli, Jenly Ramdan
D4 Informatics Engineering



Kreatif Industri Nusantara

Penulis:

Etika Khusnul Laeli
Jenly Ramdan

ISBN : 978-602-53897-0-2

Editor:

M. Nurkamal Fauzan
Etika Khusnul Laeli
Jenly Ramdan

Penyunting:

M.Nurkamal Fauzan

Desain sampul dan Tata letak:

Informatics Research Center

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2
Bandung 40191
Tel. 022 2045-8529
Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center
Jl. Sariasisih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2020

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

*'Everybody in this
country should learn to
program a computer;
because it teaches you
how to think.'*

-Steve Jobs-

CONTRIBUTORS

ETIKA KHUSNUL LAELI, JENLY RAMDAN D4 Informatics Engineering ., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Pengenalan Android Studio	1
2 Pengenalan Web Hosting	99
3 Pengenalan XAMPP	111
4 Pengenalan Database	121
5 Contoh Studi Kasus	129
6 Metodologi Penelitian	137
7 Bagian-bagian Android Studio	157
8 Cara Mendebug Aplikasi	167
9 Interfaces Aplikasi	181
10 Kesimpulan dan Saran	205

DAFTAR ISI

Daftar Gambar	xv
Foreword	xxix
Kata Pengantar	xxxii
Acknowledgments	xxxiii
Glossary	xxxv
List of Symbols	xxxvii
Introduction	xxxix
<i>Etika Khusnul Laeli, Jenly Ramdan</i>	
1 Pengenalan Android Studio	1
1.1 Sejarah Android Studio	1
1.2 Definisi Android Studio	2
1.3 Macam-Macam Android Studio	3
1.3.1 Update 3.1	9
1.3.2 Update 3.2	9
1.4 Mengkonfigurasi Android Studio	16
1.4.1 Menemukan file konfiguasi Anda	16

1.4.2	Menyesuaikan opsi VM anda	17
1.4.3	Ukuran heap maksimum	17
1.4.4	Mengoptimalkan performa Android Studio di Windows	18
1.5	Karakteristik Android Studio	18
1.6	Varian Build	19
1.7	Fitur Profil dan Debug	19
1.8	Pesan Log	19
1.9	Manfaat Mempelajari Android Studio	20
1.10	Antarmuka Pengguna	20
1.11	Customize UI dengan memanfaatkan fitur Layout Editor	21
1.12	Pengantar Layout Editor	21
1.13	Mengubah penampilan pratinjau	22
1.14	Membuat tata letak baru	23
1.15	Membuat varian tata letak	24
1.16	Mengonversi tampilan atau tata letak	24
1.17	Mengonversi tata letak ke Constraint Layout	25
1.18	Menemukan item di Palette	25
1.19	Membuka dokumentasi dari Palette	25
1.20	Menambahkan tampilan ke tata letak Anda	26
1.21	Bahasa Java	26
1.21.1	Memahami Struktur dan Aturan Penulisan Sintaks Java	27
1.21.2	Variabel dan Tipe Data	30
1.21.3	Input dan Output di Java	30
1.21.4	6 Jenis Operator yang Harus dipahami	31
1.21.5	3 Bentuk Percabangan dalam Java	31
1.21.6	Memahami 2 Jenis Perulangan dalam Java	32
1.21.7	Menggunakan Array	34
1.21.8	Menggunakan Prosedur dan Fungsi untuk membuat Sub-Program	35
1.21.9	Mengenal dan Memahami Class HashMap di Java	36
1.21.10	Konsep Dasar OOP pada Java	38
1.21.11	Inheritance dan Method Overriding	40
1.21.12	Tingkat Akses Member dan Class (Modifier)	47
1.21.13	Mengenal Constructor dan Destructor dalam Java	52
1.21.14	Menggunakan Method Setter dan Getter untuk Enkapsulasi	56
1.21.15	Memahami kata kunci 'this' dan 'super' pada Java	59
1.21.16	Memahami Prinsip Polimorfisme dalam OOP	68

1.21.17 Memahami Interface di Java	68
1.21.18 Mengenal Class Abstrak dan Cara Pakainya	74
1.21.19 Mengenal Classs Anonymus di Java	75
1.21.20 Mengenal Lambda Expression untuk Membuat Fungsi Anonymus di Java	78
1.22 Bahasa Kotlin	80
1.22.1 Keunggulan Kotlin	81
1.22.2 Fitur Unggulan Kotlin	81
1.22.3 Persiapan Pengembang	82
1.23 Cara Instalasi Android Studio pada windows	83
1.24 Cara Instalasi Android Studio pada Linux	93
2 Pengenalan Web Hosting	99
2.1 Pengertian Web Hosting	99
2.2 Cara Kerja Web Hosting	100
2.3 Cara membuat akun 000webhost	100
2.4 Fungsi Utama Web Hosting	101
2.5 Jenis-jenis dari Web Hosting	102
2.6 Perbedaan Web Hosting dan Heroku	103
2.6.1 Web Hosting	103
2.6.2 Heroku	104
2.7 Web Hosting vs Heroku	105
2.7.1 Web Hosting	105
2.8 10 Kontrol Panel Hosting Populer Berbasis Open Source	105
2.9 Flowchart	108
2.9.1 Pengertian Flowchart	108
2.9.2 Fungsi Flowchart	108
2.9.3 Simbol-simbol Flowchart	109
3 Pengenalan XAMPP	111
3.1 XAMPP	111
3.1.1 Pengertian XAMPP	111
3.1.2 Fungsi XAMPP	112
3.1.3 Cara Install XAMPP di windows	112
3.1.4 Cara Installasi XAMPP pada Linux	116
4 Pengenalan Database	121
4.1 Pengertian Database	122

4.2	Pengertian Database menurut para ahli	123
4.3	Fungsi Database	124
4.4	Manfaat Database	125
4.5	Tujuan Basis Data	125
4.6	Maria DB	126
4.7	Istilah yang digunakan dalam RDBMS	126
5	Contoh Studi Kasus	129
5.1	Latar Belakang	129
5.2	Metode Pengumpulan Data	130
5.3	Contoh studi kasus penggunaan aplikasi ini	131
5.4	Jurnal Yang Terkait Dengan Penilitian	132
5.5	Siapakah pengguna aplikasi smart gudang	134
5.6	Tentang kelas Menengah-bawah	135
5.6.1	Kelas Menengah	135
5.6.2	Kelas Bawah	136
6	Metodologi Penelitian	137
6.1	Metodologi Penelitian	137
6.2	Analisis dan Perancangan	138
6.2.1	Sequence Diagram	139
6.2.2	Activity Diagram	142
6.2.3	Class Diagram	147
6.2.4	Statechart Diagram	148
6.3	Implementasi	154
6.3.1	Lingkungan Tujuan	154
6.4	Pengujian dan Hasil Pengujian	156
6.4.1	Identifikasi dan Rencana Pengujian	156
6.4.2	Deskripsi dan Hasil Uji	156
7	Bagian-bagian Android Studio	157
7.1	Bagian-bagian pada Android Studio	157
7.1.1	Activty Main.xml	157
7.1.2	Main Activity.java	159
7.1.3	Android Mannifest.xml	163
7.1.4	Membuat fitur scan barcode kamera menggunakan library ZXING	164
8	Cara Mendebug Aplikasi	167

8.1	Cara Mendebug Aplikasi	167
8.2	Contoh Cara Mendebug Aplikasi Hello Word	167
8.3	Cara Mendebug Aplikasi Smart Gudang	175
8.3.1	Cara Menambahkan Barang pada Smart Gudang	175
8.3.2	Cara Mengurangi Barang pada Smart Gudang	178
9	Interfaces Aplikasi	181
9.1	Interfaces Aplikasi	181
9.2	Tampilan Aplikasi	195
9.3	Tata Cara Penggunaan Aplikasi	200
9.4	Mengatasi Error	200
10	Kesimpulan dan Saran	205
10.1	Kesimpulan dan Saran	205
10.1.1	Kesimpulan	205
10.1.2	Saran	206
	Daftar Pustaka	207

DAFTAR GAMBAR

1.1	OS Android Studio	1
1.2	Android 1.0	3
1.3	Android 1.1	4
1.4	Android 1.5	5
1.5	Android 1.6	5
1.6	Android 2.0	6
1.7	Android 2.2	7
1.8	Android 2.3	8
1.9	Android 3.0-3.2.6	10
1.10	Android 4.0	11
1.11	Android 4.1.2	12
1.12	Android 4.4	13
1.13	Android 5.0	13

1.14	Android 6.0	14
1.15	Android 7.0	15
1.16	Android 8.0	16
1.17	Notifikasi terkait setelan memori yang direkomendasikan	17
1.18	Notifikasi untuk minimalkan dampak virus	18
1.19	Jendela Utama Android Studio	20
1.20	Layout XML	22
1.21	Tombol baris editor	23
1.22	Bahasa pemrograman Java	26
1.23	Contoh Deklarasi Package	27
1.24	Contoh Impor Library	27
1.25	Contoh Blok Class	28
1.26	Contoh Blok Class	28
1.27	Fungsi	28
1.28	Statement dan Ekspresi	29
1.29	Format penulisan variabel	30
1.30	Impor Scanner	30
1.31	Contoh menampilkan fungsi print()	30
1.32	Format penggunaan if/else/if	32
1.33	Contoh perulangan For	32
1.34	Contoh perulangan For Each	33
1.35	Contoh Perulangan While	33
1.36	Contoh Perulangan Do/While	34
1.37	Cara membuat Array	34
1.38	Untuk menggunakan Array List import java.util.ArrayList	35
1.39	Format untuk membuat Array List	35
1.40	Struktur Dasar membuat Fungsi	35
1.41	Contoh membuat Fungsi	36

1.42	Contoh pemanggilan Fungsi dalam fungsi main	36
1.43	Hasil output pemanggilan Fungsi di dalam fungsi main	36
1.44	Import HashMap	36
1.45	Contoh membuat objek HashMap	37
1.46	Contoh mengisi objek dengan method put	37
1.47	Contoh mengambil nilai dengan method get	37
1.48	Contoh OOP dalam bahasa pemrograman Java	38
1.49	Contoh class asumsi game	38
1.50	Contoh class	39
1.51	Contoh objek	39
1.52	Contoh mengakses atribut dan method	39
1.53	Contoh class-class pada game	40
1.54	File : Zombie.java	40
1.55	File : Pocong.java	41
1.56	File : Burung.java	41
1.57	Contoh class dengan Inheritance	42
1.58	File : Enemy.java	42
1.59	File : Zombie.java	43
1.60	File : Pocong.java	43
1.61	File : Burung.java	43
1.62	Contoh Objek asumsi game	44
1.63	Contoh pembuatan method overide Persegi.java	44
1.64	Contoh pembuatan method overide Lingkaran.java untuk semua class anak	45
1.65	Contoh pembuatan method overide PersegiPanjang.java untuk semua class anak	45
1.66	Contoh pembuatan method overide Segitiga.java untuk semua class anak	46
1.67	Eksekusi Clas Main	46

1.68	Contoh kode program modifier	47
1.69	Tabel Modifier	48
1.70	Contoh Modifier Public	48
1.71	Solusi untuk class Person	49
1.72	Contoh modifier private	49
1.73	Solusi untuk dapat mengakses member private	50
1.74	Class diagram modifier private	50
1.75	Contoh modifier protected	51
1.76	Contoh modifier protected	51
1.77	Class diagram protected	52
1.78	Contoh Constructor	52
1.79	Contoh penulisan Constructor	52
1.80	Objek baru dari class user	53
1.81	Kode Program Constructor	53
1.82	Hasil Eksekusi	53
1.83	Contoh constructor parameter sebagai inputan	54
1.84	Contoh constructor parameter mmembuat objek	54
1.85	Kode Program lengkap constructor Parameter	55
1.86	Hasil output constructor Parameter	55
1.87	Kode Program destructor C++	56
1.88	Method setter dan getter	56
1.89	Contoh method setter dan getter	57
1.90	Contoh kode program method setter dan getter	58
1.91	Contoh mengakses setter dan getter	58
1.92	Hasil output mengakses setter dan getter	59
1.93	Contoh mengakses setter dan getter	59
1.94	Contoh kata kunci this untuk mengisi variabel	60
1.95	Contoh method this	60

1.96	Contoh method this dihilangkan	60
1.97	Kode Program yang this dihilangkan akan error	61
1.98	Pesan peringatan <i>Assignment To Itself</i>	61
1.99	Solusi dengan variabel this	61
1.100	Contoh penggunaan this pada class Person	62
1.101	Contoh penggunaan super pada class Person menjadi class induk atau super class	62
1.102	Contoh membuat class turunan(sub class)	63
1.103	Contoh membuat class Demo untuk membuat objek dan method main	63
1.104	Contoh membuat class Demo untuk membuat objek dan method main	63
1.105	Mengganti super menjadi this	64
1.106	Hasil output menggantikan super menjadi this	64
1.107	Memanggil method tertentu dari kelas induk dengan super	64
1.108	Mengubah class Person	65
1.109	Mengubah class Employee	65
1.110	Mengubah class Demo	65
1.111	Hasil output class Demo	66
1.112	Membuat class MainWindow dengan meg-extends dari class JFrame	66
1.113	this mewakili class MainWindow	67
1.114	Contoh MainActivity pada Android	67
1.115	super untuk memanggil method onCreate dari class induk AppCompatActivity	67
1.116	Class Diagram PhoneUser	69
1.117	Contoh Interface class Phone	69
1.118	Contoh Interface class Phone	70
1.119	Kode program Xiomi	71
1.120	Lanjutan Kode Program Xiomi	71

1.121	Lanjutan Kode Program Xiomi	72
1.122	Main.java class Phone	72
1.123	Main.java class Phone	73
1.124	Method yang terhubung dengan Interface	73
1.125	Contoh class abstrak	74
1.126	Contoh membuat class abstrak	74
1.127	Buat nama interface	75
1.128	Buat Class Button	76
1.129	Buat Class Main	76
1.130	Hasil class anonymus	77
1.131	Contoh objek anonymus	77
1.132	Menggunakan class anonymus secara langsung	78
1.133	Menggunakan class anonymus secara langsung	78
1.134	Simbol pada Lambda Expression	79
1.135	Simbol pada Lambda Expression	79
1.136	Contoh program Lambda Expression	80
1.137	Bahasa pemrograman Kotlin	80
1.138	IntelliJ IDEA	82
1.139	IDEA Eclipse	82
1.140	Langkah pertama untuk install Android Studio, buka website resmi Android Studio	83
1.141	Halaman persetujuan Install	83
1.142	Halaman Instalasi Scope	84
1.143	Klik Next	84
1.144	Pilih komponen Android Studio	85
1.145	Klik I Agree	85
1.146	Klik Next	86
1.147	Klik install	86

1.148	Installasi sedang berjalan	87
1.149	Klik Next	87
1.150	Installasi siap digunakan	88
1.151	Klik Ok	88
1.152	Klik Next	89
1.153	Pilih Standard dan Next	89
1.154	Klik Next	90
1.155	Klik Next	90
1.156	Klik Finsih	91
1.157	Tunggu proses downloadng selanjutnya klik finish	91
1.158	Klik Finish	92
1.159	Andrid Studio siap digunakan	92
1.160	Halaman untuk mendownload android studio	93
1.161	Halaman untuk popup dan beri tanda centang pada bagian I have read and agree with the above terms and conditions	93
1.162	Ekstrak file zip yang telah diunduh	94
1.163	Terminal untuk Android Studio	94
1.164	Pilih Do Not import settings	95
1.165	Pilih Next pada jendela yang muncul	95
1.166	Pilih Standar untuk type install Android Studio	96
1.167	Pilih UI Item untuk install Android Studio	96
1.168	Klik Next	97
1.169	Klik Finish	97
1.170	Klik Finish	98
1.171	Jendela Welcome to Android Studio	98
2.1	Cara Kera web hosting	100
2.2	Masukan data-data untuk mendaftar webhost secara gratis	101
2.3	Web hosting yang digunakan untuk membuat aplikasi Smart Gudang	103

2.4	Imajinasi mempermudah cara kerja web hosting	104
2.5	Ini heroku yang digunakan untuk membuat aplikasi kita	104
2.6	Simbol flowchart dan penjelasannya	109
3.1	Halaman download aplikasi XAMPP	113
3.2	Langkah pertama instal XAMPP	114
3.3	Memilih komponen	114
3.4	Lokasi file penyimpanan	115
3.5	Proses Install XAMPP	115
3.6	Pilih yang XAMPP 7	116
3.7	Klik Next untuk melanjutkan	116
3.8	Tentukan komponen	117
3.9	Install XAMPP pada direktori	117
3.10	Klik Next bitnami for XAMPP	118
3.11	Klik Next untuk siap install	118
3.12	Tunggu prosesnya sampe selesai	119
3.13	Klik Finish	119
4.1	Halaman tampilan untuk Database	122
4.2	Halaman tampilan untuk Relasi antar Data	122
5.1	Usaha sembako kelas menengah-bawah	131
5.2	Contoh usaha kelas menengah-bawah	135
6.1	Alur metodologi penelitian	138
6.2	Relationship use case kelompok dan actor	139
6.3	Sequence Diagram Login Admin	139
6.4	Sequence Diagram Login Karyawan	140
6.5	Sequence Diagram Pendataan	140
6.6	Sequence Diagram Kelola Data	141
6.7	Sequence Diagram Kelola Stock	141
6.8	Sequence Diagram Laporan	142

6.9	Activity Diagram Login Admin	143
6.10	Activity Diagram Login Karyawan	144
6.11	Activity Diagram Pendataan	145
6.12	Activity Diagram Kelola Data	146
6.13	Activity Diagram Kelola Stock	146
6.14	Activity Diagram Laporan	147
6.15	Class diagram	147
6.16	Statechart Diagram Login	148
6.17	Statechart Diagram Pendataan	149
6.18	Statechart Diagram Kelola Barang Masuk	150
6.19	Statechart Diagram Kelola Barang Keluar	151
6.20	Statechart Diagram Kelola Tabel Barang	152
6.21	Statechart Diagram Kelola Tabel Stock	153
6.22	Statechart Diagram Laporan	154
6.23	Tabel Perangkat Keras	155
6.24	Tabel Perangkat Lunak	155
6.25	Identifikasi dan rencana pengujian	156
6.26	Pengujian halaman barang masuk	156
6.27	Pengujian halaman barang keluar	156
7.1	Untuk membuat posisi tampilan	158
7.2	Membuat icon dan menambahkan text view	158
7.3	Membuat Button	159
7.4	Main Activity meng extends dari ActivityMain.xml	159
7.5	Initialize object	160
7.6	Setting button dan text view	160
7.7	Untuk eksekusi barang masuk	160
7.8	Untuk eksekusi barang masuk	161
7.9	Untuk eksekusi barang keluar	161

7.10	Untuk eksekusi barang keluar	161
7.11	Untuk Mendapatkan Hasil Scan	162
7.12	Untuk button dapat di klik	162
7.13	Membuat tampilan web view	162
7.14	Meng-eksekusi tampilan web view	163
7.15	Manifest Android Studio	163
7.16	Gunakan script diatas	164
7.17	Gunakan script diatas	164
7.18	Gunakan script diatas	164
8.1	Software Android Studio	168
8.2	Klik Empty Activity	168
8.3	Setting configure your project	169
8.4	Sedang proses build gradle	170
8.5	Klik build bundle apk	170
8.6	Tunggu proses selesai	171
8.7	Pindahkan app debug.apk ke android	171
8.8	Tampilan Hello Word pada Android	172
8.9	Pilih Main Activity nya	173
8.10	Tekan Shift+F10	173
8.11	Pilih device yang akan digunakan	174
8.12	Berikut tampilan aplikasi nya	174
8.13	Buka MainActivity.java lalu klik runs 'app'	175
8.14	Ini website QR Code Generator	175
8.15	Tuliskan perintah pada Free Text kemudian pada Scanner QR Code di save	176
8.16	QR Code yang telah dibuat	176
8.17	Data Masuk ke dalam aplikasi	177
8.18	Data Masuk ke dalam tabel Informasi Stock Barang	178

8.19	Notifikasi barang keluar	179
8.20	Tampilan barang keluar pada Infromasi Stock Barang	180
9.1	Database yang digunakan	195
9.2	Tampilan yang belum terisi data	196
9.3	Tampilan yang sudah terisi data	197
9.4	Informasi Stock dengan web view	198
9.5	Tampilan Aplikasi Potret	199
9.6	Tampilan aplikasi landscape	200

Listings

9.1	Source Code untuk membuat layout atau tampilan utama.	181
9.2	Source Code untuk mengeksekusi layout atau tampilan utama.	183
9.3	Source Code untuk membuat layout tampil web view.	187
9.4	Source Code untuk mengeksekusi layout tampil atau web view nya.	187
9.5	Source Code Koneksi.	188
9.6	Source Code Konfigurasi.	189
9.7	Source Code RequestHandler.	190
9.8	Source Code JSONParser.java.	193
9.9	Source Code Manifest Android.	194

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih lagi Maha Penyayang, puji syukur kami panjatkan ke hadirat Allah SWT karena berkat rahmat dan hidayah-Nyalah kami dapat menyelesaikan Proyek 2 berupa pembuatan Buku yang berjudul “Smart Gudang” Dalam kesempatan ini kami tidak lupa menyampaikan ucapan terima kasih kepada semua pihak yang telah memberikan bantuan moral dan spiritual, langsung maupun tidak langsung dalam menyelesaikan buku ini, terutama kepada:

1. Orang tua dengan dukungan dan do'a nya.
2. DR. Ir. Agus Purnomo, M.T. selaku Direktur Politeknik Pos Indonesia.
3. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika.
4. Rolly Maulana Awangga, S.T., M.T. selaku Koordinator Proyek 2.
5. M. Nurkamal Fauzan S.T., M.T selaku Pembimbing Proyek 2 yang telah memberikan pengarahan dan membimbing kepada penulis.
6. Roni Habibi S.Kom., M.T. Selaku Dosen Pengaji Proyek 2.
7. M. Nurkamal Fauzan S.T., M.T selaku dosen wali kelas D4 TI 2B

8. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Kami menyadari bahwa Buku ini masih jauh dari kategori sempurna. Namun, kami sudah berusaha semampu kami. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun untuk kesempurnaan Buku yang akan datang.

ETIKA KHUSNUL LAELI , JENLY RAMDAN

Bandung, Jawa Barat

Januari, 2020

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para dosen pembimbing, mahasiswa/i agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk kedua orang tua yang selalu mensupport dan mendoakan kami yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Proyek 2.

Etika Khusnul Laeli, Jenly Ramdan

GLOSSARY

Android Studio

adalah Lingkungan Pengembangan Terpadu(Integrated Development Environment/IDE), yang didasarkan pada IntelliJ IDEA.

Android SDK Build-Tools

Wajib. Mencakup fitur untuk membuat aplikasi Android.

Android SDK Platform-Tools

Wajib. Mencakup berbagai fitur yang diperlukan oleh platform Android, termasuk fitur adb.

Android SDK Tools

Wajib. Mencakup fitur penting seperti ProGuard.

Android Emulator

Direkomendasikan. Fitur emulasi perangkat berbasis QEMU yang dapat Anda gunakan untuk men-debug dan menguji aplikasi di lingkungan runtime Android sebenarnya.

Android SDK Platform

Wajib. Lingkungan Anda memerlukan setidaknya satu platform agar aplikasi dapat dikompilasi.

Image Sistem Inel atau ARM

Direkomendasikan. Image sistem diperlukan untuk menjalankan Android Emulator.

SYMBOLS

A Amplitude

$\&$ Propositional logic symbol

a Filter Coefficient

B Number of Beats

INTRODUCTION

ETIKA KHUSNUL LAELI, JENLY RAMDAN

D4 Informatics Engineering
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. Android Studio merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi pengantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi Android Studio pada diri dan organisasinya.

BAB 1

PENGENALAN ANDROID STUDIO

1.1 Sejarah Android Studio



Gambar 1.1 OS Android Studio

Pertama kali muncul Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Perusahaan ini dibangun oleh beberapa senior di beberapa perusahaan yang berbasis IT dan Communication, Andy Rubin, Rich Miner, Nick Sears dan Chris White. Rubin [1] menyatakan bahwa, Android Inc Didirikan untuk mewujudkan mobile device yang lebih fleksibel terhadap lokasi dan preferensi pemilik. Sehingga, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya selain karena OS nya yang open source.

Berawal dari konsep inilah Android Inc ternyata menarik minat Google untuk memilikinya. Maka, pada bulan Agustus 2005, Akhirnya Android Inc diakui sisi oleh Google Inc. dan seluruh sahamnya dibeli oleh Google. Perusahaan milik Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka pun ikut menjadi bagian dari raksasa Google dan sejarah Android. Disini mereka mulai menggunakan platform Linux untuk membuat sistem operasi bagi mobile phone.

Dari sinilah akhirnya banyak pengembang sistem maupun software yang mengembangkan maupun merancang sistem Android menggunakan software – software yang support dengan Android, Contohnya ialah : Android Studio.

1.2 Definisi Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada InelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang handal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android seperti:

1. Sistem build berbasis Gradle yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
5. Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. Framework dan fitur pengujian yang lengkap.
7. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

1.3 Macam-Macam Android Studio

1. Android 1.0 (Apple Pie)

Android versi pertama ini dirilis pada 23 September 2008 dan hanya dilengkapi fitur-fitur seperti Play Store, Web Browser, Kamera, Sinkronisasi antara Gmail, Contacts dan Google Agenda. Selain itu, diawal perilisannya, Android juga sudah dilengkapi aplikasi Google Maps serta dukungan streaming Youtube.



Gambar 1.2 Android 1.0

2. Android 1.1 (Banana Bread)

Sistem Operasi android yang rilis selanjutnya adalah Banana Bread, rilis pada bulan Februari 2009. Dan fitur ini juga tidak jauh berbeda dengan versi sebelumnya. HTC adalah salah satu ponsel Android pertama yang menggunakan versi ini.



Gambar 1.3 Android 1.1

3. Android 1.5(Cupcake)

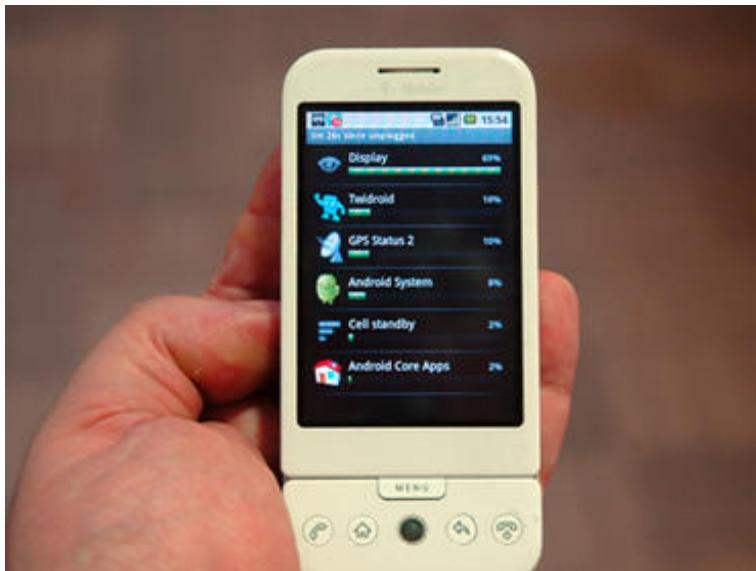
Rilis pada awal bulan April 2009 dan juga tidak jauh berbeda dengan versi Android sebelumnya. Hanya saja ada fitur tambahan seperti Support Bluetooth A2DP, AVRCP, Soft-keyboard dengan prediksi text dan record/watch videos.



Gambar 1.4 Android 1.5

4. Android 1.6(Donut)

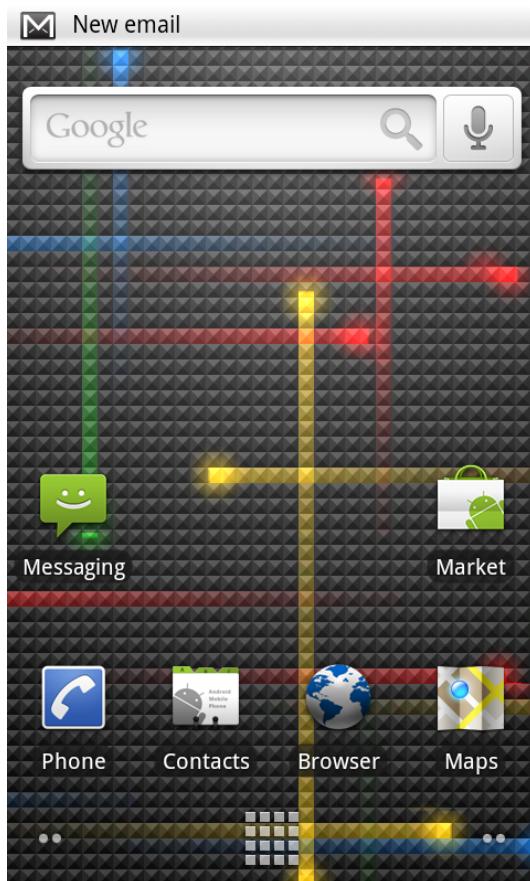
Android Donut rilis pada 15 September 2009, dan mendapat fitur tambahan seperti Gesture Framework hingga Turn-by-turn navigation. Selain itu, Android ini juga terlihat lebih sempurna pada waktu itu. Dengan minimnya bug, ditambah lebih lengkapnya fitur-fitur yang disediakan Google.



Gambar 1.5 Android 1.6

5. Android 2.0(Éclair)

Android versi 2.0 bernamakan Eclair dan rilis pada 26 Oktober 2009 silam. Yang selain bluetooth, Android versi ini juga mendapatkan fitur multi-touch, Live Wallpaper dan juga flash kamera. Selain itu, adapun beberapa fitur yang dapat anda nikmati dalam Android versi ini adalah yakni, HTML, Digital zoom, Support Microsoft Exchange, dan Updated UI.



Gambar 1.6 Android 2.0

6. Android 2.2(Froyo)

Pada bulan Mei 2010 lalu, Google telah merilis Android versi terbaru pada waktu itu. Yakni adalah Android 2.2 (Froyo). Versi ini merupakan salah satu sistem operasi Android yang juga telah disempurnakan, utamanya tentu untuk meningkatkan kecepatan kinerja suatu Android.



Gambar 1.7 Android 2.2

Dan berikut ini adalah fitur dan perbaikan yang disediakan oleh Android versi 2.2 :

- (a) Peningkatan Speed
- (b) Implementasi JIT
- (c) USB Tethering
- (d) Aplikasi instalasi untuk perluasan memori
- (e) Support file upload pada the browser
- (f) Animated GIFs

7. Android 2.3(Gingerbread)

Pada bulan Desember 2010 lalu, Google secara resmi merilis Android versi terbaru, Gingerbread. Yang secara fitur jelas sudah sangat sempurna. Ditambah lagi, Android versi 2.3 ini juga diadopsi oleh salah satu perusahaan Smartphone paling terkenal, yaitu Samsung dengan menanamkan sistem operasi ini dalam ponsel seri Nexus-nya.



Gambar 1.8 Android 2.3

8. Android 3.0 – 3.2.6(Honeycomb)

Honeycomb merupakan salah satu sistem operasi Android versi terbaru yang rilis pada bulan Februari 2011 silam. Namun, versi ini lebih ditujukan untuk Tablet yang mana pada tahun itu sangat laris dipasaran. Fitur dan perbaikan pada Android versi ini:

- (a) Support Multi core
- (b) Support Tablet lebih baik
- (c) Updated 3D UI
- (d) Layar Utama (homescreens) yang bisa diatur
- (e) Melihat aplikasi yang barusan dibuka
- (f) Menyempurnakan layout keyboard
- (g) Transport protocol untuk Media/Picture

- (h) Video chat Google Talk
- (i) Google eBooks
- (j) “Private browsing”
- (k) System-wide Clipboard
- (l) HTTP Live streaming

1.3.1 Update 3.1

- (a) Peningkatan UI
- (b) Open Accessory
- (c) USB host API
- (d) Support mouse, joysticks dan gamepad
- (e) Notifikasi MTP
- (f) RTP API untuk audio

1.3.2 Update 3.2

- (a) Optimise untuk berbagai tablets
- (b) Mode kompatibilitas display (zoom for fixed-sized apps)
- (c) Sinkronisasi Media dari SD card

1.3.2.1 Update 3.2.1

- (a) Update Android Market termasuk automatic updates yang lebih mudah
- (b) Update Google Books
- (c) Peningkatan kinerja Wi-Fi
- (d) Perbaikan prediksi tulisan tangan huruf Chinese

1.3.2.2 Update 3.2.2

- (a) Perbaikan kecil

1.3.2.3 3.2.4

- (a) Update tambahan ‘Pay as you go’ untuk tablet

1.3.2.4 Update 3.2.6

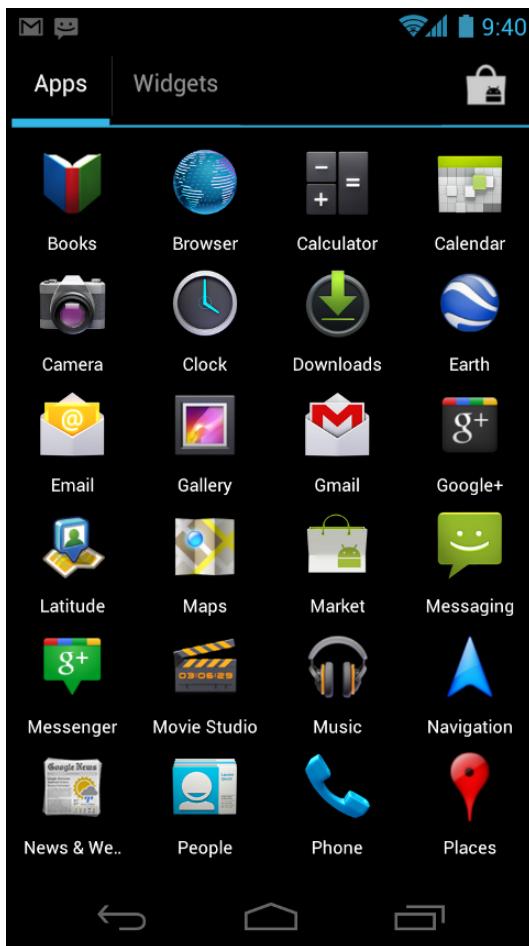
(a) Perbaikan kecil



Gambar 1.9 Android 3.0-3.2.6

9. Android 4.0(Ice Cream Sandwich)

Puncak kematangan Android yakni ketika pada versi ini, yang mana Ice Cream Sandwich rilis pada bulan Oktober 2011 silam. Dan operasi sistem ini mulai bekerja di semua jenis smartphone apapun. Selain bertambahnya fitur-fitur menarik, Ice Cream Sandwich juga merupakan versi Android paling banyak disukai pada waktu itu. Bahkan, Android Ice Cream Sandwich juga dilengkapi dengan fitur ekstra multitasking dan notifikasi yang lebih banyak.



Gambar 1.10 Android 4.0

10. Android 4.1.2(Jelly Bean)

Jelly Bean rilis pada 9 Juli 2012 lewat konferensi I/O Google. Versi ini merupakan salah satu versi Android yang kerap mendapatkan update fitur-fitur yang berguna dan menarik, beberapa halnya adalah seperti memperbaiki rotasi layar, seperti Support resolusi video 4K, Support penulisan huruf Hebrew and Arabic dari kanan ke kiri, dan peningkatan kinerja, sistem keamanan dan masih banyak lainnya.



Gambar 1.11 Android 4.1.2

11. Android 4.4(Kitkat)

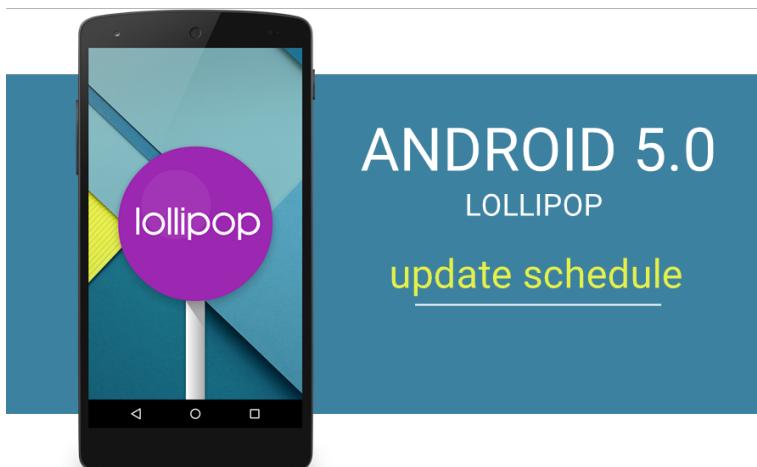
Android versi inilah yang saat ini banyak digunakan oleh mayoritas masyarakat Indonesia. Kitkat adalah versi Android yang rilis pada 2013 lalu. pada versi ini, Android banyak mendapatkan pembaharuan fitur. Seperti, terdapat fitur Screen recording, untuk merekam kegiatan yang terjadi pada layar smartphone anda, New Translucent system UI, Peningkatan akses notifikasi, System-wide settings untuk closed captioning, Peningkatan kinerja dan masih banyak yang lainnya.



Gambar 1.12 Android 4.4

12. Android 5.0(Lollipop)

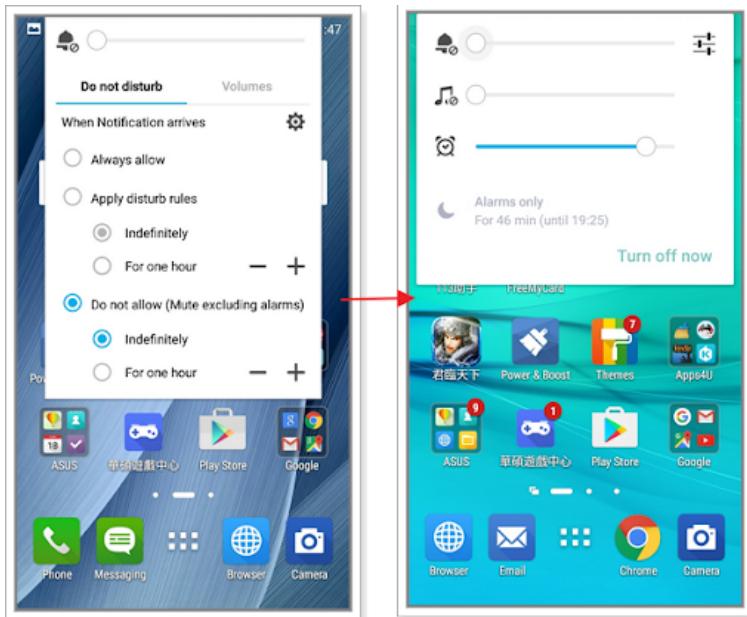
Rilis pada tahun 2014, Android yang satu ini lebih banyak menawarkan fitur tambahan untuk menyempurnakan fitur-fitur yang sudah ada. Dan Nexus 6 adalah salah satu ponsel yang paling pertama mencicipi Android versi ini. Selain itu, Google juga lebih menyempurnakan kinerja dari Android Lollipop sendiri



Gambar 1.13 Android 5.0

13. Android 6.0(Marshmallow)

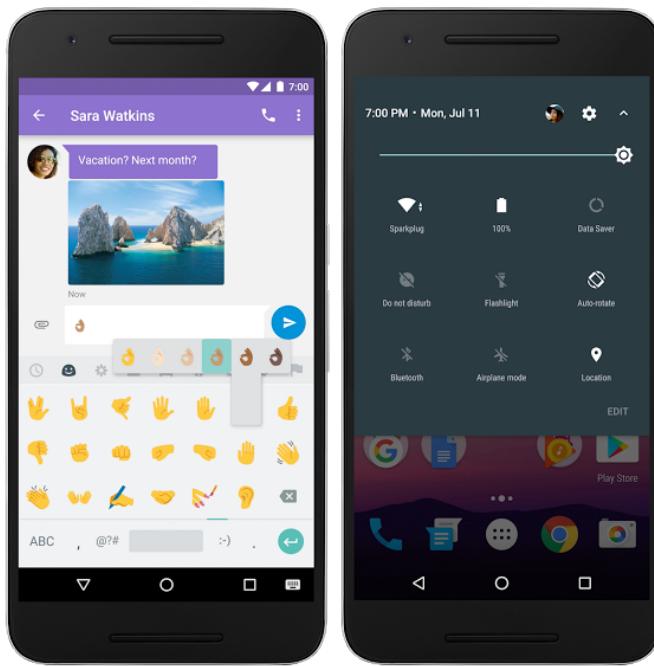
Android versi 6.0 merupakan salah satu sistem operasi Android yang rilis pada tahun 2015 silam, yang mana banyak membawa pembaharuan. Salah satunya adalah support USB Type-C. Tidak hanya itu saja, Android versi 6 ini serta memberikan fasilitas autentikasi sidik jari dan daya baterai yang lebih meningkat



Gambar 1.14 Android 6.0

14. Android 7.0(Nougat)

Android Nougat versi 7.0 rilis pada bulan Agustus 2016 silam yang lebih meningkatkan kinerja versi Android sebelumnya. Selain itu, Android Nougat juga mendapatkan banyak fitur-fitur baru yang diantaranya seperti dapat multitasking, meningkatkan fitur Doze yang dulu telah rilis di Android versi sebelumnya.



Gambar 1.15 Android 7.0

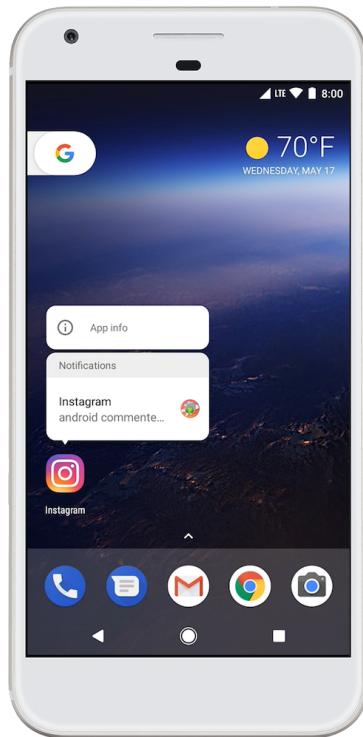
Dan inilah beberapa fitur terbaru yang terdapat pada Nougat.

- (a) Support Multi window
 - (b) Dapat langsung membalas pesan dari jendela atau menu notifikasi.
 - (c) Tampilan panel notifikasi dan quick settings yang baru.
 - (d) Mode Doze yang ditingkatkan, (Doze Mode 2.0)
 - (e) Menu di antara system settings.
15. Android 8.0(Oreo)

Android versi Oreo rilis pada bulan Agustus 2017 lalu. Tentu saja Android versi ini adalah versi final untuk sekarang ini. Beberapa fitur juga turut diluncurkan Google selaku pihak pengelola. Adapun fitur-fitur tersebut antara lain adalah

- (a) Android O lebih fokus pada kecepatan dan efisiensi
- (b) Kecepatan Boot up 2X lebih cepat
- (c) Mode Picture in picture lebih flexibel dari Android N
- (d) Aplikasi yang berjalan di latarbelakang lebih diperketat untuk menghemat battery
- (e) Battery lebih tahan lama

- (f) Emoji yang diperbaharui dan lebih banyak



Gambar 1.16 Android 8.0

1.4 Mengkonfigurasi Android Studio

Android Studio menyediakan wizard dan template yang memverifikasi persyaratan sistem Anda, seperti Java Development Kit (JDK) dan RAM yang tersedia, serta mengonfigurasi setelan default, seperti emulasi Android Virtual Device (AVD) default yang dioptimalkan dan image sistem yang telah diperbarui. Dokumen ini menjelaskan setelan konfigurasi tambahan yang mungkin dapat Anda gunakan untuk menyesuaikan penggunaan Android Studio.

1.4.1 Menemukan file konfiguasi Anda

Kedua file konfigurasi disimpan di folder konfigurasi untuk Android Studio. Nama folder bergantung pada versi Studio Anda. Misalnya, Android Studio 2.2 memiliki nama folder AndroidStudio3.3. Lokasi folder ini bergantung pada sistem operasi Anda.

1.4.2 Menyesuaikan opsi VM anda

File studio.vmoptions memungkinkan Anda menyesuaikan opsi untuk JVM di Android Studio. Untuk meningkatkan performa Studio, opsi paling umum yang biasa disesuaikan adalah ukuran heap maksimum, namun Anda juga dapat menggunakan file studio.vmoptions untuk menggantikan setelan default lainnya seperti ukuran heap awal, ukuran cache, dan switch pengumpulan sampah Java.

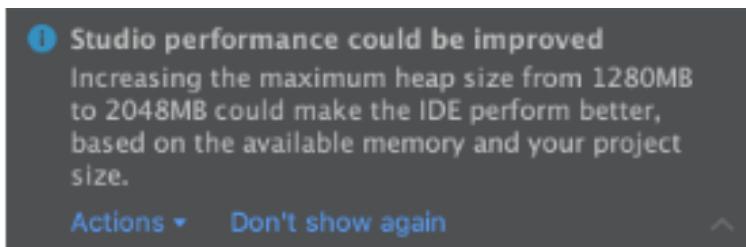
Untuk membuat file studio.vmoptions baru atau membuka file yang sudah ada, gunakan langkah berikut:

- (a) Klik Help - Edit Custom VM Options. Jika Anda belum pernah mengedit opsi VM untuk Android Studio sebelumnya, IDE akan meminta Anda membuat file studio.vmoptions baru. Untuk membuat file, klik Yes.
- (b) File studio.vmoptions akan terbuka di jendela editor Android Studio. Edit file tersebut untuk menambahkan opsi VM yang Anda sesuaikan sendiri. Untuk daftar lengkap opsi JVM yang dapat disesuaikan, lihat halaman Java HotSpot VM Options dari Oracle. File studio.vmoptions yang Anda buat akan ditambahkan ke file studio.vmoptions default, yang berada di direktori bin/ dalam folder penginstalan Android Studio.

1.4.3 Ukuran heap maksimum

Secara default, Android Studio memiliki ukuran heap maksimum 1.280 MB. Jika menangani project besar, atau sistem Anda memiliki RAM yang tinggi, Anda dapat meningkatkan performa dengan meningkatkan ukuran heap maksimum untuk proses Android Studio, seperti IDE inti, daemon Gradle, dan daemon Kotlin.

Android Studio otomatis memeriksa kemungkinan pengoptimalan ukuran heap dan memberi tahu Anda jika mendeteksi bahwa performa dapat ditingkatkan.



Gambar 1.17 Notifikasi terkait setelan memori yang direkomendasikan

Jika menggunakan sistem 64 bit yang memiliki RAM minimal 5 GB, Anda juga dapat menyesuaikan ukuran heap untuk project secara manual. Untuk melakukannya, ikuti langkah berikut:

- (a) Klik File - Settings dari panel menu (atau Android Studio - Preferences di macOS).
- (b) Klik Appearance and Behavior - System Settings - Memory Settings.
- (c) Sesuaikan ukuran heap agar cocok dengan jumlah yang Anda inginkan.
- (d) Klik Apply.

1.4.4 Mengoptimalkan performa Android Studio di Windows

Performa Android Studio di Windows dapat dipengaruhi oleh berbagai faktor. Bagian ini menjelaskan cara mengoptimalkan setelan Android Studio untuk mendapatkan performa terbaik di Windows.

1.4.4.1 Minimalkan dampak software antivirus pada kecepatan build Beberapa software antivirus dapat mengganggu proses build Android Studio, yang menyebabkan build berjalan sangat lambat. Saat Anda menjalankan build di Android Studio, Gradle mengompilasi resource dan kode sumber aplikasi Anda, lalu memaketkan resource yang telah dikompilasi bersama-sama ke dalam sebuah APK. Selama proses ini, banyak file dibuat di komputer Anda. Jika software antivirus Anda mengaktifkan pemindaian real-time, antivirus dapat memaksa proses build untuk berhenti setiap kali file dibuat saat antivirus memindai file tersebut.

Untuk menghindari masalah ini, Anda dapat mengecualikan direktori tertentu dari pemindaian real-time dalam software antivirus.



Perhatian: Untuk memastikan keamanan komputer Anda dari software berbahaya, sebaiknya jangan menonaktifkan pemindaian real-time atau software antivirus sepenuhnya.

Gambar 1.18 Notifikasi untuk minimalkan dampak virus

1.5 Karakteristik Android Studio

- (a) Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul.

(b) Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga(third-party application). Semua aplikasi dapat dibangun untuk memilih akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

(c) Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna,kalender atau lokasi geografis.

(d) Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

1.6 Varian Build

Sistem build dapat membantu teman-teman untuk membuat beberapa versi berbeda untuk aplikasi yang sama dari satu project. Hal ini berguna saat Anda menyediakan aplikasi dalam versi gratis dan berbayar, atau jika teman-teman ingin mendistribusikan beberapa APK untuk berbagai konfigurasi perangkat di Google Play.

1.7 Fitur Profil dan Debug

Gunakan proses debug inline untuk menyempurnakan panduan kode Anda dalam tampilan debugger verifikasi inline untuk referensi, ekspresi, dan nilai variabel. Informasi debug inline meliputi :

(a) Nilai variabel inline

(b) Objek perujuk yang merujuk ke objek terpilih

(c) Nilai yang dihasilkan metode

(d) Ekspresi operator dan lambda

(e) Nilai tooltip

1.8 Pesan Log

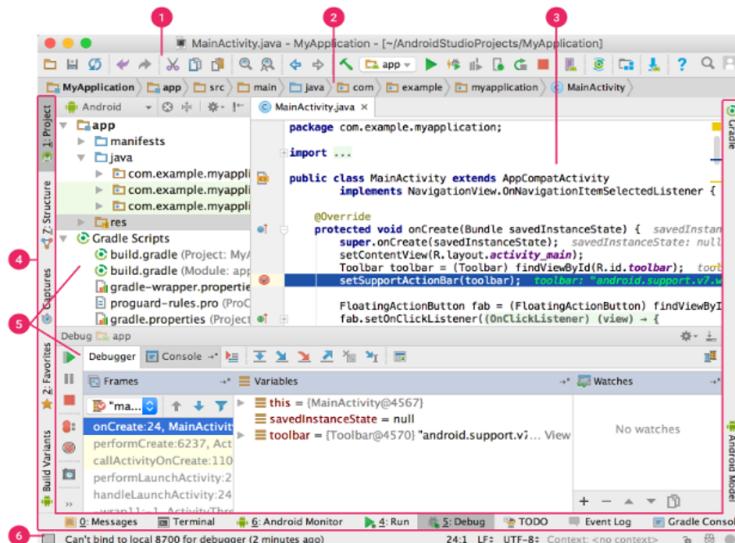
Saat membuat dan menjalankan aplikasi dengan Android Studio, Anda bisa melihat output pada adb dan pesan log perangkat di jendela Logcat.

1.9 Manfaat Mempelajari Android Studio

- (a) Dengan mempelajari Android Studio dapat membantu Anda untuk mempercepat pembuatan Aplikasi yang Anda Inginkan
- (b) Android Studio merupakan sebuah tools yang mudah dipahami dan digunakan
- (c) Dalam satu tools ini Anda bisa mendapatkan berbagai manfaat mulai dari pembuatan aplikasi hingga testing aplikasi
- (d) Bahkan, dengan belajar Android Studio maka Anda bisa menghemat waktu kerja untuk dapat lebih produktif
- (e) Dapat memperdalam ilmu codingan dengan baik. Karena dalam android studio diberikan beberapa refensi ketika mengetik sintaks. Dengam begitu tentunya Anda akan mencari tahu apa saja kegunaan dari sintaks yang terdapat.
- (f) Sarana pembelajaran coding dan pembuatan aplikasi yang baik dan praktis hanya dengan Android Studio.

1.10 Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang diidentifikasi dalam gambar seperti di bawah ini :



Gambar 1.19 Jendela Utama Android Studio

1. Toolbar

Toolbar memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan fitur Android

2. Menu Navigasi

Menu navigasi membantu Anda menjelajah project dan membuka file untuk di edit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela Project.

3. Jendela Editor

Jendela Editor adalah tempat Anda membuat dan memodifikasi kode. Tergantung jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.

4. Panel Jendela Fitur

Panel Jendela Fitur berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan Anda memperluas atau mencuitkan setiap jendela fitur.

5. Jendela Fitur

Jendela Fitur memberi Anda akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi dan banyak lagi. Anda dapat memperluas dan mencuitkan jendela ini.

6. Status Bar

Status Bar menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesa.

1.11 Customize UI dengan memanfaatkan fitur Layout Editor

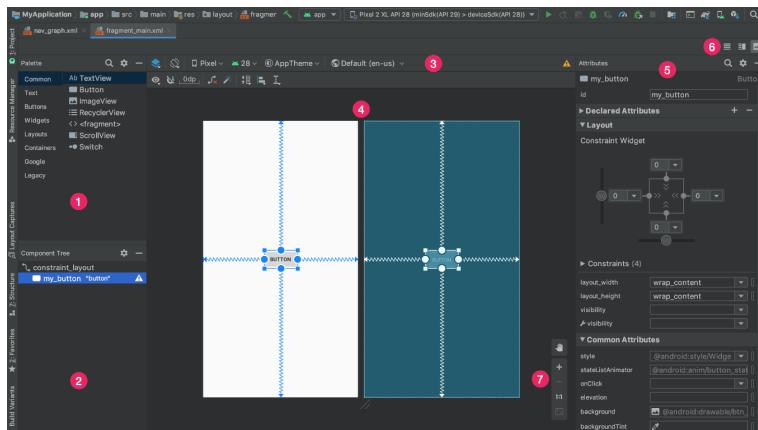
Layout Editor adalah fitur yang disediakan oleh android studio untuk customize ui pada file xml dengan cara drag. Layout Editor dapat dengan cepat mem-build tata letak dengan menarik elemen UI ke dalam editor desain visual sebagai ganti menulis XML tata letak secara manual. Editor desain ini bisa menampilkan pratinjau tata letak Anda pada beragam versi dan perangkat Android, dan Anda bisa secara dinamis mengubah ukuran tata letak untuk memastikannya berfungsi dengan baik pada berbagai ukuran layar.

Layout Editor sangat berguna saat membangun tata letak baru dengan ConstraintLayout—pengelola tata letak yang disediakan dalam library dukungan yang kompatibel dengan Android 2.3 (API level 9) dan versi yang lebih tinggi.

1.12 Pengantar Layout Editor

Layout Editor akan muncul bila Anda membuka file tata letak XML.

1. Palette: Daftar tampilan dan grup tampilan yang bisa Anda tarik ke dalam tata letak.
2. Component Tree: Menampilkan hierarki tata letak Anda
3. Toolbar: Tombol untuk mengonfigurasi penampilan tata letak Anda di editor dan untuk mengubah beberapa atribut tata letak.
4. Design Editor: Tata letak dalam tampilan Desain atau Cetak Biru, atau keduanya.
5. Attributes: Kontrol untuk atribut tampilan yang dipilih



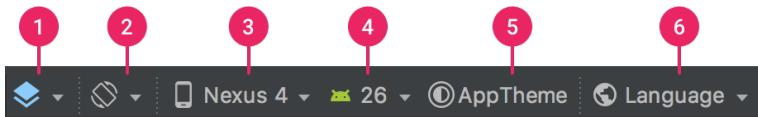
Gambar 1.20 Layout XML

Jika Anda membuka file tata letak XML, editor desain akan dibuka secara default (seperti yang ditampilkan dalam gambar 1).

Untuk mengedit XML tata letak di editor teks, klik tab Text di bagian bawah jendela. Sedangkan di editor teks, Anda juga bisa menampilkan Palette, Component Tree, dan editor desain dengan mengklik Preview di sisi kanan jendela. Jendela Attributes tidak tersedia dari editor teks.

1.13 Mengubah penampilan pratinjau

Tombol-tombol di baris atas editor desain memungkinkan Anda mengonfigurasi penampilan tata letak dalam editor. Toolbar ini juga tersedia di jendela Preview editor teks.



Gambar 1.21 Tombol baris editor

Keterangan :

1. Design and blueprint : Pilih cara Anda ingin menampilkan tata letak di editor; pilih tampilan Design (pratinjau sesungguhnya dari tata letak Anda), tampilan Blueprint (hanya outline untuk setiap tampilan), atau Design + Blueprint untuk tampilan berdampingan.
2. Screen orientation and layout variants: Pilih antara orientasi layar lanskap dan potret, atau mode layar lainnya yang aplikasi Anda sediakan dengan tata letak alternatif, seperti mode malam. Menu ini juga berisi perintah-perintah untuk membuat varian tata letak baru.
3. Device type and size: Pilih jenis perangkat (ponsel/tablet, Android TV, atau Wear OS) dan konfigurasi layar (ukuran dan kepadatan). Anda bisa memilih dari sejumlah tipe perangkat yang telah dikonfigurasi dan definisi AVD Anda sendiri, atau memulai AVD baru dengan memilih Add Device Definition dari daftar
4. API Version: Pilih versi Android untuk melihat pratinjau tata letak Anda
5. App theme: Pilih tema UI mana yang akan diterapkan ke pratinjau. (Ini hanya berfungsi untuk gaya tata letak yang didukung; sehingga banyak tema dalam daftar ini yang menyebabkan error.)
6. Language: Pilih bahasa untuk menampilkan string UI Anda. Daftar ini hanya menampilkan bahasa yang tersedia dalam resource string Anda. Jika Anda ingin mengedit terjemahan, klik Edit Translations dari menu drop-down.

1.14 Membuat tata letak baru

Saat menambahkan tata letak baru untuk aplikasi Anda, mulailah dengan membuat file tata letak dalam direktori layout/ default project Anda agar diterapkan pada semua konfigurasi perangkat. Setelah memiliki tata letak default, Anda bisa membuat variasi tata letak untuk konfigurasi perangkat tertentu (misalnya untuk layar ekstra besar).

Ada beberapa macam cara untuk membuat tata letak baru, bergantung pada tampilan jendela Project Anda, tetapi prosedur berikut dapat diakses dari tampilan apa saja:

1. Di jendela Project, klik modul (misalnya app) yang ingin Anda tambahkan tata letak.

2. Di menu utama, pilih File - New - XML - Layout XML File
3. Dalam dialog yang muncul, masukkan nama file, tag tata letak akar, dan kumpulan sumber yang memiliki tata letak tersebut. Lalu, klik Finish

Cara lain untuk memulai file tata letak baru (walaupun dialog yang muncul berbeda) adalah sebagai berikut: Jika Anda memilih tampilan Project di jendela Project: buka direktori res untuk modul aplikasi Anda, klik kanan direktori tata letak tempat Anda ingin menambahkan tata letak kemudian klik New - Layout resource file.

Jika Anda telah memilih tampilan Android di jendela Project: klik kanan folder layout kemudian pilih New - Layout resource file.

1.15 Membuat varian tata letak

Jika Anda sudah memiliki tata letak dan ingin membuat versi alternatifnya untuk mengoptimalkan tata letak bagi orientasi atau ukuran layar yang berbeda, ikuti langkah-langkah ini: Buka file tata letak asli dan pastikan Anda menampilkan editor desain (klik tab Design di bagian bawah jendela).

Klik Orientation for Preview di toolbar. Dalam daftar drop-down, klik varian yang diusulkan misalnya Create Landscape Variant dan selesai, atau klik Create Other lalu lanjutkan ke langkah berikutnya.

Dalam dialog yang muncul, Anda hanya perlu mendefinisikan qualifier resource untuk nama direktori tersebut. Anda bisa mengetikkannya dalam Directory name atau memilih dari daftar Available qualifiers, satu per satu, dan mengklik Add

Setelah menambahkan semua qualifier, klik OK. Jika Anda memiliki beberapa variasi tata letak yang sama, Anda bisa dengan mudah beralih tata letak dari daftar yang muncul saat mengklik Layout Variants. Untuk mengetahui informasi selengkapnya tentang cara membuat tata letak pada layar yang berbeda

1.16 Mengonversi tampilan atau tata letak

Anda bisa mengonversi suatu tampilan ke jenis tampilan lain, dan Anda bisa mengonversi tata letak (grup tampilan) ke jenis tata letak lain. Dengan cara sebagai berikut :

1. Klik tab Design di bagian bawah jendela editor.
2. Di Component Tree, klik kanan tampilan atau tata letak, kemudian klik Convert view.
3. Dalam dialog yang muncul, pilih tipe baru tampilan atau tata letak, kemudian klik Apply.

1.17 Mengonversi tata letak ke Constraint Layout

ConstraintLayout adalah grup tampilan yang tersedia di library Constraint Layout, yang telah disertakan bersama Android Studio 2.2 dan versi yang lebih tinggi. Tata letak ini dibuat dari nol dengan Layout Editor, jadi apa pun dapat diakses dari editor desain dan Anda tidak perlu mengedit XML secara manual. Yang paling penting, sistem tata letak berbasis batasan ini memungkinkan Anda membuat hampir semua tata letak tanpa grup tampilan tersarang.

Untuk meningkatkan performa tata letak, Anda harus mengonversi tata letak lama ke ConstraintLayout.

Untuk mengonversi tata letak yang ada ke ConstraintLayout, lakukan langkah-langkah berikut ini:

1. Buka tata letak yang ada di Android Studio dan klik tab Design di bagian bawah jendela editor.
2. Di jendela Component Tree, klik kanan tata letak kemudian klik Convert layout to ConstraintLayout
3. Perintah untuk mengonversi tata letak secara spesifik ke ConstraintLayout lebih cerdas dalam hal perkiraan batasan dan pemeliharaan tata letak daripada perintah Convert view sederhana yang dijelaskan di bagian sebelumnya.

1.18 Menemukan item di Palette

1. Untuk menelusuri tampilan atau grup tampilan berdasarkan nama di Palette, klik tombol Search di sisi atas palet, atau mulai ketik nama item ketika jendela Palette aktif.
2. Anda bisa menemukan item yang sering digunakan dalam kategori Common di Palette. Untuk menambahkan item ke kategori ini, klik kanan tampilan atau grup tampilan di Palette lalu klik Favorite di menu konteks.

1.19 Membuka dokumentasi dari Palette

1. Untuk membuka dokumentasi referensi Android Developers untuk tampilan atau grup tampilan, pilih elemen UI di Palette dan tekan Shift+F1.
2. Untuk membuka dokumentasi Panduan Material untuk tampilan atau grup tampilan, klik kanan elemen UI di Palette lalu pilih Material Guidelines dari menu konteks. Jika tidak ada entri spesifik untuk item tersebut, perintah ini akan membuka halaman beranda Dokumentasi Panduan Material.

1.20 Menambahkan tampilan ke tata letak Anda

- Untuk mulai membuat tata letak Anda, tarik saja tampilan dan grup tampilan dari Palette ke dalam editor desain. Saat Anda menempatkan tampilan dalam tata letak, editor akan menampilkan informasi tentang hubungan tampilan dengan tata letak lainnya.
- Jika Anda menggunakan ConstraintLayout, Anda bisa secara otomatis membuat batasan dengan menggunakan fitur Infer Constraints dan Autoconnect.

1.21 Bahasa Java

Bahasa Pemrograman Java



Gambar 1.22 Bahasa pemrograman Java

Bahasa pemrograman java merupakan bahasa yang berada pada urutan 10 besar bahasa pemrograman yang terpopuler di dunia saat ini.[2] Bahasa pemrograman Java pada tahun 2017 merupakan bahasa paling populer, namun sekarang sudah disalip oleh bahasa pemrograman JavaScript dan Python.

Salah satu penyebabnya yaitu karena jutaan aplikasi android dibuat menggunakan bahasa pemrograman java. Untuk membuat aplikasi android menggunakan bahasa java kita bisa menggunakan tools atau IDE:

- Android Studio (IDE resmi didukung penuh oleh google)
- Eclipse (IDE lain yang sebelumnya didukung penuh oleh google sebelum adanya android studio)

Untuk pemula yang baru ingin membuat aplikasi android disarankan menggunakan

bahasa pemrograman java. Java merupakan bahasa pemrograman yang menggunakan paradigma OOP (Object Oriented Programming). Oleh sebab itu, teman-teman harus belajar konsep OOP pada Java. OOP ini merupakan hal yang harus dipelajari untuk memahami Java lebih dalam. Karena Java sendiri merupakan bahasa yang didesain untuk OOP.

1.21.1 Memahami Struktur dan Aturan Penulisan Sintaks Java

Struktur program Java secara umum dibagi menjadi 4 bagian:

1. Deklarasi Package

Package merupakan sebuah folder yang berisi sekumpulan program Java. Deklarasi package biasanya dilakukan saat membuat program atau aplikasi besar. Contoh deklarasi package;

```
package com.smartgudang.program;
```

Gambar 1.23 Contoh Deklarasi Package

Aturannya : nama domain dibalik, lalu diikuti nama programnya.

2. Impor Library

Pada bagian ini, kita melakukan impor library yang dibutuhkan pada program. Library merupakan sekumpulan class dan fungsi yang bisa kita gunakan dalam membuat program.

Contoh impor library;

```
import java.util.Scanner;
```

Gambar 1.24 Contoh Impor Library

Pada contoh tersebut, kita mengimpor class Scanner dari package java.util.

3. Bagian Class

Java merupakan bahasa pemrograman yang menggunakan paradigma OOP (*Object Oriented Programming*). Setiap program harus dibungkus di dalam class agar nanti bisa dibuat menjadi objek.

Contoh:

```
class NamaProgram {
    public static void main(String args[]){
        System.out.println("Hello World");
    }
}
```

Gambar 1.25 Contoh Blok Class

Blok class dibuka dengan tanda kurung kurawal kemudian ditutup dengan tutip kurawal. Di dalam blok class juga bisa diisi dengan method atau fungsi-fungsi dan juga variabel.

4. Method Main

Method **main** atau fungsi **main()** merupakan blok program yang akan dieksekusi pertama kali. Method **main()** wajib teman-teman buat, karena kalau tidak programnya tidak bisa dieksekusi.

Contoh method **main()**:

```
public static void main(String args[]){
    System.out.println("Hello World");
}
```

Gambar 1.26 Contoh Blok Class

Method **main()** memiliki parameter **args[]** yang digunakan untuk menyimpan sebuah nilai dari argumen di *command line*.

(a) Lalu di dalam method **main()** terdapat statement seperti berikut :

```
System.out.println("Hello World");
```

Gambar 1.27 Fungsi
System.Out.Println

Fungsi ini digunakan untuk menampilkan teks ke layar monitor.

(b) Statement dan Ekspresi pada Java

Statement dan eksrepesi adalah bagian terkecil dalam program. Setiap statement dan ekspresi di Java, harus diakhiri dengan titik koma (;).

```
System.out.println("Hello World");
System.out.println("Apa kabar?");
var x = 3;
var y = 8;
var z = x + y;
```

Gambar 1.28 Statement dan Ekspresi
System.Out.Println

Statement dan ekspresi akan menjadi instruksi yang akan dikerjakan oleh komputer.

(c) Blok Program Java

Blok program merupakan kumpulan dari statement dan ekspresi yang dibungkus menjadi satu. Blok program selalu dibuka dengan kurung kurawal dan ditutup dengan .

Intinya: jika teman-teman menemukan kurung kurawal dan , maka itu adalah sebuah blok program.

(d) Penulisan Komentar pada Java

Komentar merupakan bagian program yang tidak akan dieksekusi oleh komputer. Komentar biasanya digunakan untuk:

- i. Memberi keterangan pada kode program.
- ii. Menonaktifkan fungsi tertentu.
- iii. Membuat dokumentasi, dan lain-lain.

Penulisan komentar pada Java menggunakan :

- i. Garis miring ganda (//) untuk komentar satu baris.
- ii. Garis miring bintang (*...*) untuk komentar yang lebih dari satu baris.

(e) Penulisan String dan Karakter

String merupakan kumpulan dari karakter atau sering mengenalnya dengan teks dan Aturan penulisan string pada Java, harus diapit dengan tanda petik ganda. Apabila diapit dengan tanda petik tunggal, maka akan menjadi sebuah karakter.

(f) Case Sensitive

Java bersifat **Case Sensitive** artinya huruf besar atau kapital dan huruf kecil dibedakan.

(g) Gaya Penulisan Case

Gaya penulisan case (case style) yang digunakan oleh Java adalah: **camel-Case, PascalCase, dan ALL UPPER**.

1.21.2 Variabel dan Tipe Data

Variabel adalah tempat menyimpan nilai sementara sedangkan Tipe Data adalah jenis data yang tersimpan dalam variabel. Macam-macam Tipe Data :

1. **char** Tipe data karakter, contoh A,B,C,D,E dll.
2. **int** angka atau bilangan bulat, contoh 2,4,5,20 dll.
3. **float** bilangan desimal, contoh 2.1,3.4, dll.
4. **string** kumpulan dari karakter yang membentuk teks, contoh **Hello World!**.
5. **boolean** tipe data yang hanya bernilai true dan false.

Membuat Variabel

Hal yang perlu diketahui dalam pembuatan variabel di java adalah cara penulisannya. Format :

```
<tipe data> namaVariabel;
```

Gambar 1.29 Format penulisan variabel

1.21.3 Input dan Output di Java

Program komputer terdiri dari tiga komponen utama, yaitu :

1. Input: nilai yang kita masukan ke program.
2. Proses: langkah demi langkah yang dilakukan untuk mengelola input menjadi sesuatu yang berguna.
3. Output: hasil pengolahan.

Contoh mengambil Input dengan Class Scanner ;

```
import java.util.Scanner;
```

Gambar 1.30 Impor Scanner

Contoh menampilkan output;

```
System.out.print(namaDepan);
System.out.print(namaBelakang);
```

Gambar 1.31 Contoh menampilkan fungsi **print()**

1.21.4 6 Jenis Operator yang Harus dipahami

1. Operator Aritmatika.
2. Operator Penugasan.
3. Operator Pembanding.
4. Operator Logika.
5. Operator Bitwise.
6. Operator Ternary.

1.21.5 3 Bentuk Percabangan dalam Java

Percabangan adalah sebuah istilah yang digunakan untuk menyebut alur program yang bercabang. Cara menuliskan kode percabangan dalam Java dengan menggunakan kata kunci **if**, **else**, **switch**, dan **case**. 3 bentuk percangan yaitu :

1. Percabangan IF Percabangan ini hanya memiliki satu pilihan. Artinya, pilihan di dalam IF hanya akan dikerjakan kalau kondisinya benar. Tapi kalau salah, tidak akan melakukan apa-apa. Alias lanjut eksekusi ke perintah berikutnya.
2. Percabangan IF/ELSE Sedangkan percabangan IF/ELSE memiliki pilihan alternatif kalau kondisinya salah.
IF : "Jika kondisi benar maka kerjakan, kalau tidak salah lanjut ".
IF/ELSE : "Jika kondisi benar maka kerjakan kalau salah maka yang diajalankan kondisi else".
3. Percabangan IF/ELSE/IF atau SWITCH/CASE Jika percabangan IF/ESLE hanya memiliki dua pilihan saja. Maka percahangan IF/ELSE/IF memiliki lebih dari dua pilihan.

```

if (suatu kondisi) {
    // maka kerjakan ini
    // kerjakan perintah ini juga
    // ...
} else if (kondisi lain) {
    // kerjakan ini
    // kerjakan ini juga
    // ...
} else if (kondisi yang lain lagi) {
    // kerjakan perintah ini
    // kerjakan ini juga
    // ...
} else {
    // kerjakan ini kalau
    // semua kondisi di atas
    // tidak ada yang benar
    // ...
}

```

Gambar 1.32 Format penggunaan if/else/if

1.21.6 Memahami 2 Jenis Perulangan dalam Java

Perulangan dalam pemrograman java dibagi menjadi dua jenis :

- Counted loop:** Perulangan yang jumlah pengulangannya terhitung atau tentu. Perulangan ini terdiri dari perulangan For dan For each.
- (a) Contoh Perulangan For

```

class Bintang{
    public static void main(String[] args){

        for(int i=0; i <= 5; i++){
            System.out.println("*****");
        }
    }
}

```

Gambar 1.33 Contoh perulangan For

- (b) Contoh perulangan For Each

```
public class PerulanganForeach {  
    public static void main(String[] args) {  
  
        // membuat array  
        int angka[] = {3,1,42,24,12};  
  
        // menggunakan perulangan For each untuk  
        for( int x : angka ){  
            System.out.print(x + " ");  
        }  
  
    }  
}
```

Gambar 1.34 Contoh perulangan For Each

2. **Uncounted loop:** Perulangan yang jumlah pengulangannya tidak terhitung atau tidak tentu. Perulangan *uncounted loop* terdiri dari perulangan *While* dan *Do/While*.

- (a) Contoh Perulangan While

```
int i = 0;  
  
while ( i <= 10 ){  
    // blok kode yang akan diulang  
    System.out.println('Perulangan ke-' + i);  
  
    // increment nilai i  
    i++;  
}
```

Gambar 1.35 Contoh Perulangan While

- (b) Contoh Perulangan Do/While

```

public class PerulanganDoWhile {
    public static void main(String[] args) {

        // membuat variabel
        int i = 0;

        do {
            System.out.println("perulangan ke-" + i);
            i++;
        } while ( i <= 10);

    }
}

```

Gambar 1.36 Contoh Perulangan Do/While

1.21.7 Menggunakan Array

Array adalah sebuah variabel yang bisa menyimpan banyak data dalam satu variabel. Array menggunakan indeks untuk memudahkan akses terhadap data yang disimpannya.

Cara membuat Array :

```

// cara pertama
String[] nama;

// cara kedua
String nama[];

// cara ketiga dengan kata kunci new
String[] nama = new String[5];

```

Gambar 1.37 Cara membuat Array

Keterangan :

1. Menggunakan kurung siku [] untuk membuat array.
2. Kurung siku bisa diletakkan setelah tipe data atau nama array.
3. Angka 5 dalam kurung artinya batas atau ukuran array-nya.

ArrayList

Array yang dibahas diatas memiliki beberapa kekurangan yaitu :

1. Tidak mampu menyimpan data dengan tipe yang berbeda.
2. Ukurannya tidak dinamis.

Oleh karena itu, ada Array List yang menutupi. Array list merupakan sebuah class yang memungkinkan kita membuat sebuah objek untuk menampung apapun. Untuk menggunakan Array List, kita harus mengimportnya terlebih dahulu.

```
import java.util.ArrayList;
```

Gambar 1.38 Untuk menggunakan Array List import java.util.ArrayList

Setelah itu baru bisa membuat objek Array List:

```
ArrayList al = new ArrayList();
```

Gambar 1.39 Format untuk membuat Array List

1.21.8 Menggunakan Prosedur dan Fungsi untuk membuat Sub-Program

1.21.8.1 Pengertian Prosedur, Fungsi dan Method

1. **Prosedur** adalah **sebutan** untuk fungsi yang tidak mengembalikan nilai. Fungsi ini biasanya ditandai dengan kata kunci **void**.
2. **Fungsi** adalah **sebutan** untuk fungsi yang mengembalikan nilai.
3. **Method** adalah **fungsi** yang berada di dalam Class. Sebutan ini, biasanya digunakan pada OOP.

Untuk memudahkan sebut saja semuanya **Fungsi**. Struktur dasar dalam membuat fungsi:

```
static TypeDataKembalian namaFungsi(){  
    // statemen atau kode fungsi  
}
```

Gambar 1.40 Struktur Dasar membuat Fungsi

Keterangan:

1. Kata kunci **static**, artinya teman-teman membuat fungsi yang dapat dipanggil tanpa harus membuat instansi objek.
2. **TypeDataKembalian** adalah tipe data dari nilai yang dikembalikan setelah fungsi di eksekusi.
3. **namaFungsi()** adalah nama fungsinya. Biasanya ditulis huruf kecil di awalnya. Llau jika terdapat lebih dari satu susku kata, huruf awal di kata kedua ditulis Kapital.

Contoh :

```
static void ucapSalam(){
    System.out.println("Selamat Pagi");
}
```

Gambar 1.41 Contoh membuat Fungsi

Tipe data **void** artinya kosong, sehingga fungsi tersebut tidak mengembalikan nilai.

1.21.8.2 Cara Memanggil/Eksekusi Fungsi Fungsi dapat dipanggil dari fungsi **main** atau dari fungsi yang lainnya.

Contoh pemanggilan fungsi dalam fungsi **main**:

```
public static void main(String[] args){
    ucapSalam();
}
```

Gambar 1.42 Contoh pemanggilan Fungsi dalam fungsi **main**

Maka hasil outputnya seperti di bawah:

The image shows the text "Selamat Pagi" in a white sans-serif font, centered within a dark rectangular background.

Gambar 1.43 Hasil output pemanggilan Fungsi di dalam fungsi **main**

1.21.9 Mengenal dan Memahami Class **HashMap** di Java

Class **HashMap** merupakan class turunan dari class **AbstractMap** implementasi dari interface **Map**. **HashMap** adalah sebuah class yang berisi sekumpulan pasangan nilai *value* dan kunci *key*.

1.21.9.1 Cara Membuat **HashMap** Sebelum menggunakan **HashMap**, teman-teman harus mengimportnya terlebih dahulu :

```
import java.util.HashMap;
```

Gambar 1.44 Import **HashMap**

Contoh membuat HashMap:

```
HashMap<Integer, String> days = new HashMap<Integer, String>
```

Gambar 1.45 Contoh membuat objek HashMap

Tipe Data yang digunakan untuk *key* adalah **Integer** dan *value* adalah **String**. Artinya; key harus bertipe **Integer** dan nilai yang tersimpan harus dalam bertipe **String**.

1.21.9.2 Mengisi Nilai ke HashMap Diatas teman-teman sudah membuat objek hashmap bernama **days** dengan tipe data:

1. K(key):Integer
2. V(value):String

Untuk mengisi nilai ke objek **days**, teman-teman bisa menggunakan method **put** seperti di bawah:

```
days.put(1, "Minggu");
days.put(2, "Senin");
days.put(3, "Selasa");
days.put(4, "Rabu");
days.put(5, "Kamis");
days.put(6, "Jum'at");
days.put(7, "Sabtu");
```

Gambar 1.46 Contoh mengisi objek dengan method **put**

1.21.9.3 Mengambil Nilai dari HashMap Untuk mengambil nilai dari Hashmap, teman-teman bisa menggunakan method **get()** dengan parameter kuncinya. Contoh :

```
// mengambil hari senin
days.get(2)
```

Gambar 1.47 Contoh mengambil nilai dengan method **get**

1.21.9.4 Menghapus Nilai dari HashMap Ada dua method yang dapat digunakan untuk menghapus nilai dari HashMap:

1. **remove()** menghapus salah satu nilai.
2. **clear()** menghapus semua nilai.

1.21.10 Konsep Dasar OOP pada Java

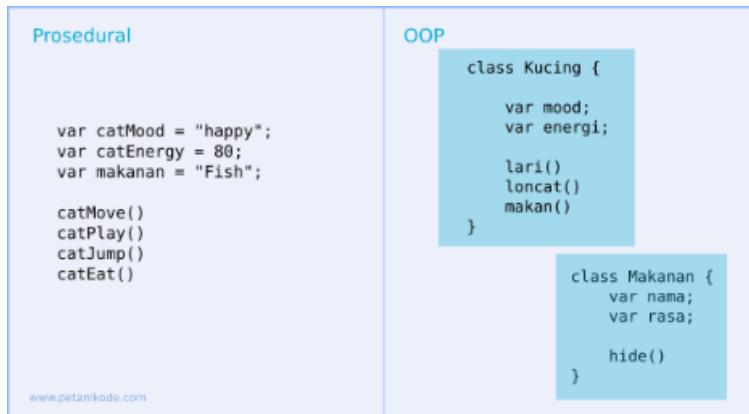
Contoh OOP dalam java :

```
class Hello {
    public static void main(String[] args){
        System.out.println("Hello World!");
    }
}
```

Gambar 1.48 Contoh OOP dalam bahasa pemrograman Java

Jadi, setelah kita membahas OOP apa teman-teman sudah memahami apa itu OOP ? Ya benar, jadi OOP adalah bahasa pemrograman berorientasikan objek(PBO). Fungsi dan variabel tersebut dibungkus dalam sebuah objek atau class yang dapat saling berinteraksi, sehingga terbentuklah sebuah program.

Nah, tapi sebelum teman-teman membuat object teman-teman harus membuat class terlebih dahulu. Apa itu fungsi class dan object ? Class berfungsi untuk mengumpulkan prosedur/fungsi dan variable dalam satu tempat. Kita tuangkan pikiran kita ke dalam sebuah game, coba lihat gambar di bawah ;



Gambar 1.49 Contoh class asumsi game

Apa yang ada di pikiran teman-teman ? NAH, jadi lewat game ini teman-teman bisa dengan mudah memahami apa sih atau mana sih yang di sebut variabel dan mana

yang disebut prosedur. Nah, class ini yang nanti nya kita bakal pakai untuk objek. Jadi, Class adalah rancangan atau blue print dari sebuah objek. Sedangkan objek adalah sebuah variabel yang merupakan instance atau perwujudan dari class. Instance juga bisa diartikan sebagai wujud dari class. Class itu berisi definisi variabel dan fungsi yang menggambarkan objek.

Dalam OOP :

1. Variabel disebut atribut atau properti.
2. Fungsi disebut method.

Contoh Class :

```
class NamaClass {  
    String atribut1;  
    String atribut2;  
  
    void namaMethod(){ ... }  
    void namaMethodLain(){ ... }  
}
```

Gambar 1.50 Contoh class

Setelahnya teman-teman bisa membuat objek (instance) seperti di bawah ini :

```
NamaClass namaObj = new NamaClass();
```

Gambar 1.51 Contoh objek

Pembahasan untuk objek: Kata kunci **new** berfungsi untuk membuat objek baru dari **class** tertentu.

Setelah membuat objek, teman-teman lanjut mengakses atribut dan method dari objek tersebut. Contoh :

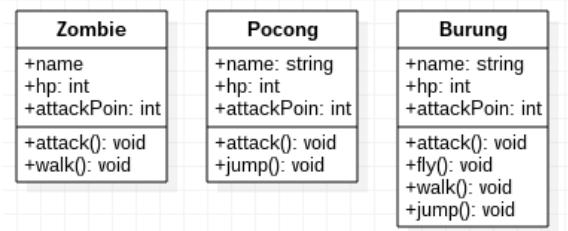
```
namaObj.namaMethod();  
namaObj.atribut1;
```

Gambar 1.52 Contoh mengakses atribut dan method

Tanda titik (.) berfungsi untuk mengakses atribut dan method.

1.21.11 Inheritance dan Method Overriding

1.21.11.1 Inheritance Sebelumnya sudah kita bahasa pada konsep OOP bahwa sebuah class atau ojek bisa saling berhubungan dengan class yang lain. Nah, salah satu bentuk hubungannya adalah **inheritance** (pewarisan). Bisa ddiabratkan hubungan ini seperti hubungan keluarga antara orang tua dan anak. Class pada Java bisa memiliki satu atau lebih keturuna atau class anak. Kenapa teman-teman harus menggunakan Inheritance ? Baik, kita bisa asumsikan terlebih dahulu pada game di bawah ini



Gambar 1.53 Contoh class-class pada game

Selanjutnya teman-teman bisa buat kode untuk masing-masing kelas seperti berikut:

1. File:Zombie.java

```

class Zombie {
    String name;
    int hp;
    int attackPoin;

    void attack(){
        // ...
    }

    void walk(){
        //...
    }
}

```

Gambar 1.54 File : Zombie.java

2. File : Pocong.java

```
class Pocong {  
    String name;  
    int hp;  
    int attackPoin;  
  
    void attack(){  
        // ...  
    }  
  
    void jump(){  
        // ...  
    }  
}
```

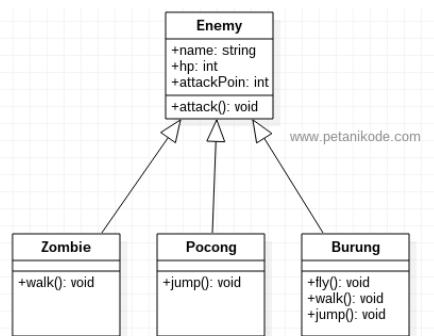
Gambar 1.55 File : Pocong.java

3. File : Burung.java

```
class Burung {  
    String name;  
    int hp;  
    int attackPoin;  
  
    void attack(){  
        // ...  
    }  
  
    void walk(){  
        // ...  
    }  
  
    void jump(){  
        // ...  
    }  
  
    void fly(){  
        // ...  
    }  
}
```

Gambar 1.56 File : Burung.java

Setelah kita membuat kode-kode seperti diatas, apakah efektif jika menggunakan fungsi diatas? Sebenarnya boleh saja menggunakan fungsi diatas namun tidak efektif karena begitu sama saja menulis berulang-ulang properti dan method yang sama. Mari membuat solusi untuk permasalahan diatas. Caranya dengan menggunakan **inheritance** seperti di bawah ini :



Gambar 1.57 Contoh class dengan Inheritance

Setelah teman-teman buat solusi dengan class Inheritance lalu teman-teman buatkan kode dalam Java nya seperti di bawah :

1. File : Enemy.java

Class Enemy adalah class induk yang akan memiliki Zombie,Pocong dan Burung. Apapun properti yang ada dalam class induk akan dimiliki oleh class anak.Begini bentuk kode nya

```

class Enemy {
    String name;
    int hp;
    int attackPoin;

    void attack(){
        System.out.println("Serang!");
    }
}
  
```

Gambar 1.58 File : Enemy.java

2. Pada class Zombie.java , teman-teman akan menggunakan kata kunci **extends** untuk menyatakan kalau dia adalah class turunan **Enemy**

```
class Zombie extends Enemy {
    void walk(){
        System.out.println("Zombie jalan-jalan");
    }
}
```

Gambar 1.59 File : Zombie.java

3. Pada class Pocong.java , teman-teman akan menggunakan kata kunci **extends** untuk menyatakan kalau dia adalah class turunan **Enemy**

```
class Pocong extends Enemy {
    void jump(){
        System.out.println("loncat-loncat!");
    }
}
```

Gambar 1.60 File : Pocong.java

4. Pada class Burung.java , teman-teman akan menggunakan kata kunci **extends** untuk menyatakan kalau dia adalah class turunan **Enemy**

```
class Burung extends Enemy {
    void walk(){
        System.out.println("Burung berjalan");
    }
    void jump(){
        System.out.println("Burung loncat-loncat");
    }
    void fly(){
        System.out.println("Burung Terbang...");
    }
}
```

Gambar 1.61 File : Burung.java

Selanjutnya utntuk membuat objek dari class-clas yang elah dibuat bisa seperti ini:

```
Enemy monster = new Enemy();
Zombie zumbi = new Zombie();
Pocong hantuPocong = new Pocong();
Burung garuda = new Burung();
```

Gambar 1.62 Contoh Objek asumsi game

1.21.11.2 Method Overriding Method Overriding dilakukan saat teman-teman ingin membuat ulang sebuah method pada sub class atau class anak. Method Overriding dibuat dengan menambahkan anotasi **@override** di atas nama method atau sebelum pembuatan method. Contoh :

1. Contoh : Persegi.java

```
class Persegi extends BangunDatar {
    float sisi;

    @Override
    float luas(){
        float luas = sisi * sisi;
        System.out.println("Luas Persegi: " + luas);
        return luas;
    }

    @Override
    float keliling(){
        float keliling = 4 * sisi;
        System.out.println("Keliling Persegi: " + keliling);
        return keliling;
    }
}
```

Gambar 1.63 Contoh pembuatan method overide Persegi.java

Artinya disitu berarti menulis ulang method **luas()** dan **keliling()** pada clas anak. Sekarang teman-teman bisa buat **method overriding** untuk semua class anak.

2. File : Lingkaran.java

```

package inheritance;

public class Lingkaran extends BangunDatar{

    // jari-jari lingkaran
    float r;

    @Override
    float luas(){
        float luas = (float) (Math.PI * r * r);
        System.out.println("Luas lingkaran: " + luas);
        return luas;
    }

    @Override
    float keliling(){
        float keliling = (float) (2 * Math.PI * r);
        System.out.println("Keliling Lingkaran: " + keliling);
        return keliling;
    }

}

```

Gambar 1.64 Contoh pembuatan method overide Lingkaran.java untuk semua class anak

3. File : PersegiPanjang.java

```

package inheritance;

public class PersegiPanjang extends BangunDatar {
    float panjang;
    float lebar;

    @Override
    float luas(){
        float luas = panjang * lebar;
        System.out.println("Luas Persegi Panjang: " + luas);
        return luas;
    }

    @Override
    float keliling(){
        float kll = 2*panjang + 2*lebar;
        System.out.println("Keliling Persegi Panjang: " + kll);
        return kll;
    }
}

```

Gambar 1.65 Contoh pembuatan method overide PersegiPanjang.java untuk semua class anak

4. File: Segitiga.java

```
package inheritance;

public class Segitiga extends BangunDatar {

    float alas;
    float tinggi;

    @Override
    float luas() {
        float luas = 1/2 * (alas * tinggi);
        System.out.println("Luas Segitiga: " + luas);
        return luas;
    }

}
```

Gambar 1.66 Contoh pembuatan method overide Segitiga.java untuk semua class anak

Untuk class **Segitiga** teman-teman hanya melakukan **override** terhadap method **Luas()** saja. Karena method **keliling()**, segitiga memiliki rumus yang berbeda-beda.

Sekarang mari coba eksekusi class **Main**

```
compile-single:
run-single:
Menghitung laus bangun datar
Menghitung keliling bangun datar
Luas Persegi: 4.0
Keliling Persegi: 8.0
Luas lingkaran: 1520.5309
Keliling Lingkaran: 138.23007
Luas Persegi Panjang:32.0
Keliling Persegi Panjang: 24.0
Luas Segitiga: 96.0
Menghitung keliling bangun datar
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 1.67 Eksekusi Clas Main

Penjelasan:

Di awal teman-teman menulis atribut dari **Enemy** yang sama berulang-ulang, berbeda jika teman-teman menulis method yang sama tapi isi atau rumus dalam method

berbeda. **Inheritance** adalah salah satu pikar penting dalam OOOP yang harus dipahami. Karena teman-teman akan sering menggunakannya dalam pembuatan program.

1.21.12 Tingkat Akses Member dan Class (Modifier)

Class dalam pemrograman Java dapat saling berhubungan dengan cara memberikan akses terhadap akses member mereka. Salah satu hubungan yang sudah teman-teman pelajari dalam buku ini adalah inheritance (pewarisan). Semua yang ada dalam *class* atribut dan *method* disebut member. Setelahnya akan ada tingkatan akses yang biasa disebut modifier. Pada hubungan *inheritance*, semua member di dalam class induk akan bisa diakses oleh class anak(subclass), kecuali member tersebut diberikan modifier *private*.

Perlu diingat bahwa Modifier tidak hanya diberikan pada member saja, tapi juga bisa diberikan kepada interface, enum dan class itu sendiri. Contoh:

```
1 package modifier;
2
3 public class User extends Person {
4     private String username;
5     private String password;
6
7     @Override
8     public String getName(){
9         return this.name;
10    }
11 }
12 }
```

Gambar 1.68 Contoh kode program modifier

Modifier ini nanti nya akan menentukan batasan akses member dan class. Secara umum ada 3 macam modifier yangdigunakan dalam bahasa Java yaitu **public** **private** dan **protected**. Apabila teman-teman tidak menggunakan tiga kunci tersebut tanda nya member atau class yang teman-teman gunakan tidak menggunakan modifier. Berikut tabel jangkauan untuk masing-masing modifier :

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Gambar 1.69 Tabel Modifier

Keterangan :

1. Y artinya bisa diakses.
2. N artinya tidak bisa diakses.
3. **Subclass** artinya class anak.
4. **World** artinya seluruh package di aplikasi.

Perlu diingat

Apabila teman-teman tidak menggunakan modifier, maka class dan member hanya akan bisa di akses dari Class itu sendiri dan package(class yang berada satu package).

1.21.12.1 Contoh Modifier public Maka agar bisa diakses oleh siapapun harus diberikan modifier.

```
package modifier;

class Person {
    public String name;

    public changeName(String newName){
        this.name = newName;
    }
}
```

Gambar 1.70 Contoh Modifier Public

Keterangan : Pada class **person** terdapat dua member, yaitu :

1. atribut **name**
2. method **changeName()**

Nah jadi kedua member tersebut merupakan contoh yang telah di berikan modifier **public**. Artinya member tersebut bisa diakses dari mana saja. Namun pada class **Person** tidak diberikan modifier, maka yang terjadi class tersebut tidak bisa diimpor (diakses) dari luar package. Mari kita buat solusinya.

Solusinya teman-teman haus menambahkan **public** ke dalam class **Person**.

```

1  package modifier;
2
3  @
4  public class Person {
5      String name;
6
7      public void setName(String name){
8          this.name = name;
9      }
10
11     public String getName(){
12         return this.name;
13     }
14 }
15

```

Gambar 1.71 Solusi untuk class Person

Nah, setelah kita menggunakan kode program tersebut dijamin tidak akan error lagi dan class **Person** akan bisa diimpor dari package manapun.

1.21.12.2 Contoh Modifier private Modifier **private** akan membuat member hanya bisa diakses oleh class itu sendiri.

Modifier **private** tidak bisa diberikan class,enum dan interface. Modifier **private** hanya bisa diberikan kepada member class.

Contoh modifier private :

```

class Person {
    private String name;

    public void setName(name){
        this.name = name;
    }

    public String getName(){
        return this.name;
    }
}

```

Gambar 1.72 Contoh modifier private

Nah, pada modifier **private** diatas, kita memberikan modifier **private** pada atribut **name** dan modifier **public** pada method **setName()** dan **getName()**

Lalu bagaimana cara mengakses member **private** dari luar class? Teman-teman bisa memanfaatkan **method setter dan getter**, karena method ini akan selalu memberikan modifier **public**.

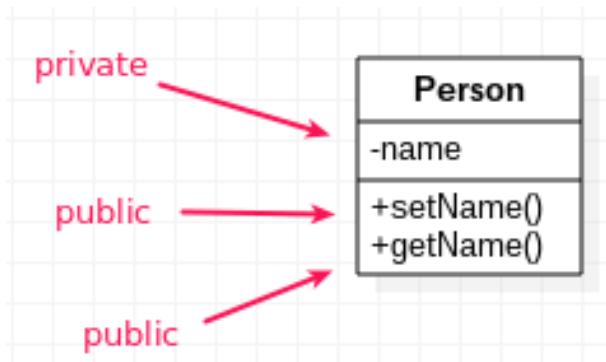
Contoh :

```
Person mPerson = new Person();
mPerson.setName("Petani Kode");

System.out.println("Person Name: " + mPerson.getName());
```

Gambar 1.73 Solusi untuk dapat mengakses member private

Bisa diasumsikan pada class diagram di bawah ini, untuk modifier **private** digambarkan dengan simbol (-).



Gambar 1.74 Class diagram modifier private

1.21.12.3 Contoh Modifier Protected Modifier **protected** akan membuat member dan class yang hanya bisa di akses dari:

1. Class itu sendiri
2. Sub class atau class anak
3. Package (class yang berada satu packag)

Modifier **protected** hanya bisa digunakan pada member saja. Contoh penggunaan modifier protected :

```
package modifier;

public class Person {
    protected String name;

    public void setName(String name){
        this.name = name;
    }

    public String getName(){
        return this.name;
    }
}
```

Gambar 1.75 Contoh modifier protected

Pada modifier seperti gambar diatas, kita hanya memberikan modifier **protected** pada atribut **name**. Dan apabila teman-teman mencoba untuk mengakses dari class yang satu package maka tidak akan terjadi error. Namun, apabila teman-teman mencoba untuk mengakses dari luar package akan terjadi error.

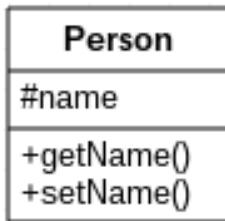
```
import modifier.Person;

public class Author {
    Person p = new Person();

    public Author() {
        // akan terjadi error di sini karena atribut name
        // telah diberikan modifier protected
        p.name = "Petani Kode";
    }
}
```

Gambar 1.76 Contoh modifier protected

Pada clas diagram, modifier **protected** bisa digambarkan dengan tanda pagar (#).



Gambar 1.77 Class diagram protected

1.21.13 Mengenal Constructor dan Destructor dalam Java

1.21.13.1 Constructor dalam Java Constructor adalah method khusus yang akan dieksesusi pada saat pembuatan objek *instance*. Method ini biasanya digunakan untuk inisialisasi atau mempersiapkan data untuk objek.

1. Contoh Constructor dan Cara membuatnya

(a) Contoh untuk Constructor;

```

package konstruktor;

public class User {
    private String username;
    private String password;
    // ini adalah constructor
    public User() {
        System.out.println("eksekusi method constructor...");
    }
}
  
```

Gambar 1.78 Contoh Constructor

(b) Cara membuat constructor sangat sederhana dengan cara menuliskan nama method constructor sesuai dengan nama class.

Pada constructor diatas dituliskan contoh nya seperti di bawah ini :

```

public User() {
    System.out.println("eksekusi method constructor...");
}
  
```

Gambar 1.79 Contoh penulisan Constructor

Catatan :

Pastikan memberikan modifier **public** kepada Constructor, karena akan di eksesusi pada saat pembuatan objek *instance*.

Mari teman-teman membuat objek baru dari class **user** :

- (c) Membuat Objek pada Constructor Parameter

```
User petani = new User();
```

Gambar 1.80 Objek baru dari class user

Sehingga akan menjadi kode program yang lengkap seperti berikut ;

- (d) Kode Program Constructor Parameter

```
package konstruktor;

public class User {
    private String username;
    private String password;

    public User() {
        System.out.println("eksekusi method constructor...");
    }
}

class DemoConstructor{
    public static void main(String[] args) {
        User petani = new User();
    }
}
```

Gambar 1.81 Kode Program Constructor

- (e) Hasilnya saat dieksekusi :

```
Running javac...
compile-single:
run-single:
eksekusi method constructor...
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 1.82 Hasil Eksekusi

2. Constructor dengan Parameter

Constructor ini biasanya digunakan untuk *initialize* (menyiapkan) data untuk class. Untuk melakukan ini, teman-teman harus membuat parameter sebagai inputan untuk constructor.

3. Contoh Constructor parameter inputan

```
public class User {  
    public String username;  
    public String password;  
  
    public User(String username, String password){  
        this.username = username;  
        this.password = password;  
    }  
}
```

Gambar 1.83 Contoh constructor parameter sebagai inputan

Mengacu pada gambar di atas, teman-teman menambahkan parameter **username** dan **password** ke dalam constructor.

Sehingga ketika teman-teman ingin membuat objek, teman-teman harus menambahkan nilai parameter seperti di bawah ini :

4. Contoh constructor parameter membuat objek

```
User petani = new User("petanikode", "kopi");
```

Gambar 1.84 Contoh constructor parameter mmembuat objek

Sehingga berdasarkan kode program diatas akan menghasilkan kode program lengkap untuk constructor parameter seperti gambar di bawah ini ;

5. Contoh Constructor Parameter Kode Program

```

package konstruktor;

public class User {
    public String username;
    public String password;

    public User(String username, String password){
        this.username = username;
        this.password = password;
    }
}

class DemoConstructor{
    public static void main(String[] args) {
        User petani = new User("petanikode", "kopi");
        System.out.println("Username: " + petani.username);
        System.out.println("Password: " + petani.password);
    }
}

```

Gambar 1.85 Kode Program lengkap constructor Parameter

- Hasil outputnya seperti gambar di bawah ini :

```

Running javac...
compile-single:
run-single:           ↗
Username: petanikode
Password: kopi
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 1.86 Hasil output constructor Parameter

1.21.13.2 Desructor dalam Java Destructor adalah method khusus yang akan dieksekusi saat objek dihapus dari memori.

Java tidak memiliki method destructor, karena java menggunakan *garbage collector* untuk menajamen memorinya.

Jadi, *garbage collector* akan otomatis mehapus objek yang tidak terpakai. Tetapi untuk bahasa pemrograman lain, seperti C++ teman-teman bisa membuat destructor sendiri seperti ini :

```

class User {
public:
    User( String *username ); // <-- ini constructor
    ~User(); // <-- ini destructor.
private:
    String username;
    String password;
};

```

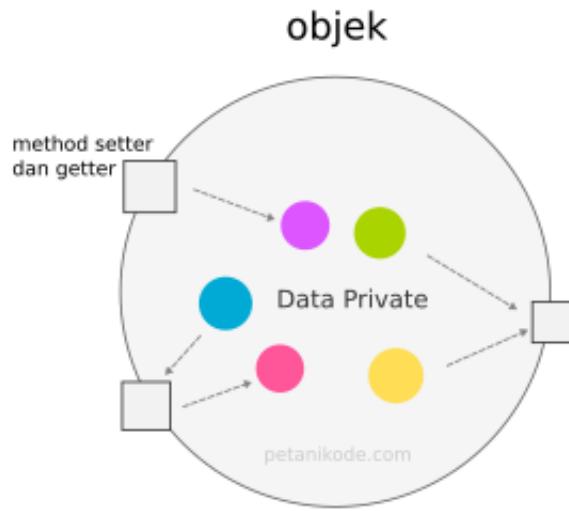
Gambar 1.87 Kode Program destructor C++

Jadi, kesimpulan yang bisa di dapat yaitu Cosntructor adalah sebuah method. Mthod yang akan dieksekusi saat pembuatan objek. Sementara untuk destructor adalah kebalikan dari constructor dan Java tidak memiliki destructor karena menggunakan metode *garbage collector*

1.21.14 Menggunakan Method Setter dan Getter untuk Enkapsulasi

Method setter dan getter adalah dua method yang tugasnya untuk mengambil dan mengisi data ke dalam objek.

Dalam OOP pasti teman-teman tidak asing dengan istilah *encapsulation* (Pembungkus), dimana data dibungkus dengan modifier **private** agar tidak bisa diakses secara langsung dari luar class.



Gambar 1.88 Method setter dan getter

Pada gambar method setter dan getter inilah yang akan membantu kita mengakses data tersebut. Beberapa alasan dibuat seperti di gambar yaitu :

1. Untuk meningkatkan keamanan data
2. Agar lebih mudah dalam mengontrol atribut dan method
3. Class bisa kita buat menjadi *read-only* dan *write-only*
4. dan fleksibel, programmer dapat mengganti sebagian dari kode tanpa harus takut berdampak pada kode yang lain

1.21.14.1 Cara Membuat Method Setter dan Getter Cara membuat method setter dan getter sama seperti membuat method biasa. Contoh ;

```
class User {  
    private String username;  
    private String password;  
  
    // ini method setter  
    public void setUsername(String username){  
        this.username = username;  
    }  
  
    public void setPassword(String password){  
        this.password = password;  
    }  
  
    // ini method getter  
    public String getUsername(){  
        return this.username;  
    }  
  
    public String getPassword(){  
        return this.password;  
    }  
}
```

Gambar 1.89 Contoh method setter dan getter

Method setter dan getter harus diberikan modifier **public**, karena method ini akan diakses dari luar class.

Perbedaan method setter dan getter terletak nilai kembalian, parameter dan isi method nya.

Method setter tidak memiliki nilai kembalian **void**(kosong). Karena tugas method setter hanya mngisi data ke dalam atribut.

Sedangkan method getter memiliki nilai kembalian sesuai dengan tipe data yang akan diambil.

```
// ini method setter
public void setUsername(String username) {
    this.username = username;
}

// ini method getter
public String getUsername(){
    return this.username;
}
```

Gambar 1.90 Contoh kode program method setter dan getter

Nama method setter dan getter harus diawali dengan **set** untuk setter dan **get** untuk getter.

Apa boleh nama method nya diganti dengan bahasa indonesia ? misalnya **isi** untuk setter dan **ambil** untuk getter. Boleh-boleh saja, namun tidak dianjurkan karena nantinya teman-teman akan bekerja tim, nah nanti tim yang lain akan kesulitan apalagi jika tim tersebut menggunakan bahasa Inggris.

1.21.14.2 Menggunakan Method Setter dan Getter Setelah teman-teman membuat method setter dan getter, teman-teman bisa mengakses atau menggunakannya seperti method biasa.

Contoh :

```
// Kode ini ditulis di dalam method main

// membuat objek dari class User
User dian = new User();

// menggunakan method setter
dian.setUsername("dian");
dian.setPassword("kopiJava");

// menggunakan method getter
System.out.println("Username: " + dian.getUsername());
System.out.println("Password: " + dian.getPassword());
```

Gambar 1.91 Contoh mengakses setter dan getter

Hasil outputnya :

```
Running javac...
compile-single:
run-single:
Username: dian
Password: kopijava
BUILD SUCCESSFUL (total time: 2
|
```

Gambar 1.92 Hasil output mengakses setter dan getter

Jadi, kesimpulannya yaitu method setter dan getter bertugas untuk mengakses dan mengisi data ke dalam objek. Contoh-contoh kasus tertentu bisa ditemukan seperti dalam pembuatan model data, pembuatan aplikasi android dll.

```
// Kode ini ditulis di dalam method main

// membuat objek dari class User
User dian = new User();

// menggunakan method setter
dian.setUsername("dian");
dian.setPassword("kopijava");

// menggunakan method getter
System.out.println("Username: " + dian.getUsername());
System.out.println("Password: " + dian.getPassword());
```

Gambar 1.93 Contoh mengakses setter dan getter

1.21.15 Memahami kata kunci 'this' dan 'super' pada Java

1.21.15.1 Fungsi this pada Java Kata kunci **this** seringkali muncul dalam program Java.

Jadi, apa fungsi this ini? Mari kita mencoba menggunakan kode-programnya.

1. Contoh kode program **this** untuk mengisi variabel.

```

class User {
    private String username;
    private String password;

    // ini method setter
    public void setUsername(String username){
        this.username = username;
    }

    public void setPassword(String password){
        this.password = password;
    }

    // ini method getter
    public String getUsername(){
        return this.username;
    }

    public String getPassword(){
        return this.password;
    }
}

```

Gambar 1.94 Contoh kata kunci **this** untuk mengisi variabel

- Perhatikan salah satu method pada kode program

```

public void setUsername(String username){
    this.username = username;
}

```

Gambar 1.95 Contoh method **this**

- Bagaimana kalo hapus **this** pada method ini?

```

public void setUsername(String username){
    username = username;
}

```

Gambar 1.96 Contoh method **this** dihilangkan

- Kode program yang dijalankan akan error seperti di bawah ini

```

3 class User {
4     private String username; ↗
5     private String password; ↗
6
7     public void setUsername(String username){ ↗
8         username = username; ? ↗
9     }
10
11    public void setPassword(String password){ ↗
12        this.password = password;
13    }

```

Gambar 1.97 Kode Program yang **this** dihilangkan akan error

Kode program ketika dijalankan akan error, karena program akan bingung untuk menentukan dimana keberadaan variabel **username** yang dimaksud. Apakah variabel yang di class (**private String Username** ataukah variabel **username** yang ada di dalam parameter. Dan jika teman-teman menyentuh *balon peringatan* di samping nomor maka akan mendapatkan pesan: "Assignment To Itself".



The screenshot shows a Java code editor with the following code:

```

2
3
4
5
6
7 class User {
8     private String username; ↗
9     private String password; ↗
10
11    public void setUsername(String username){ ↗
12        username = username; ? ↗
13    }
14
15    public void setPassword(String password){ ↗
16        this.password = password;
17    }
18
19    // ini method getter
20    public String getUsername(){
21    }

```

A red box highlights the line `username = username;`. A tooltip window titled "Assignment To Itself" appears, containing the message "The assigned value is never used". An arrow points from the tooltip to the line of code.

Gambar 1.98 Pesan peringatan *Assignment To Itself*

Yang artinya variabel tersebut mengisi dirinya sendiri.

Nah, solusinya yaitu dengan menggunakan **this** untuk menyatakan variabel yang dimaksud adalah variabel yang ada di dalam class. Seperti gambar di bawah ini :

```

public void setUsername(String username){
    this.username = username;
}

```

Gambar 1.99 Solusi dengan variabel **this**

Jadi, kata kunci **this** itu digunakan sebagai referensi dari class itu sendiri. Jika masih bingung, bisa lihat contoh di bawah :

```
class Person {
    private String name;

    public void setName(String name){
        this.name = name;
    }
}
```

Gambar 1.100 Contoh penggunaan **this** pada class **Person**

Artinya, **this** yang dimaksud pada class tersebut adalah class **Person** dan **ingat this** hanya bisa digunakan di dalam class saja.

1.21.15.2 Fungsi super pada Java Setelah mempelajari **this** teman-teman akan mempelajari lebih lanjut yaitu **super**. Jika **this** merepresentasikan objek dari class itu sendiri, maka **super** akan mempresentasikan objek dari class induk.

Contoh :

```
public class Person {
    String name = "Etika";
    int age = 22;
}
```

Gambar 1.101 Contoh penggunaan **super** pada class **Person** menjadi class induk atau super class

Kemudian, coba teman-teman buat class turunannya (sub class) yaitu : class **Employee**

```

public class Employee extends Person {

    float salary = 4000f;
    String name = "Dian";
    int age = 23;

    public void showInfo(){
        System.out.println("Name: " + super.name);
        System.out.println("Age: " + super.age);
        System.out.println("Salary: $" + salary);
    }
}

```

Gambar 1.102 Contoh membuat class turunan(sub class)

Berikutnya teman-teman bisa membuat class **Demo** untuk membuat objek dan method **main()**

```

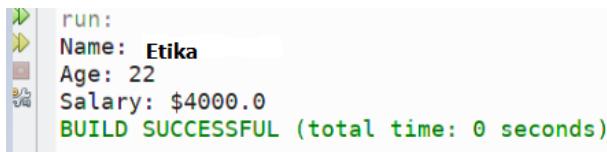
public class Demo {
    public static void main(String[] args) {
        Employee dian = new Employee("Dian", 23, 4000f);

        dian.showInfo();
    }
}

```

Gambar 1.103 Contoh membuat class Demo untuk membuat objek dan metho **main**

Setelah dibuat semuanya, mari teman-teman lakukan eksekusi :



```

run:
Name: Etika
Age: 22
Salary: $4000.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 1.104 Contoh membuat class Demo untuk membuat objek dan method **main**

Mengapa hasil outputnya begini ? Karena pada method **showInfo()** kita menggunakan kata kunci **super** untuk mengambil nilai dari variabel yang ada di dalam class induk *super class*.

Coba ganti **super** menjadi **this**

```
public void showInfo(){
    System.out.println("Name: " + this.name);
    System.out.println("Age: " + this.age);
    System.out.println("Salary: $" + salary);
}
```

Gambar 1.105 Mengganti **super** menjadi **this**

Maka hasil outputnya akan seperti di bawah ini :

```
run:
Name: Dian
Age: 23
Salary: $4000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 1.106 Hasil output menggantikan **super** menjadi **this**

Keterangan :

Nilai yang diambil pada class ini adalah nilai dari class **Employee**, karena **this** mewakili class **Employee** sedangkan **super** mewakili kelas induknya, yaitu **Person**. Pada kasus ini, kita menggunakan **super** sebagai objek untuk mngambil nilai. Kata kunci **super()** dapat juga digunakan untuk memanggil method tertentu dari kelas induk. Contoh :

```
// mengeksekusi konstruktor induk
super();

// mengeksekusi sebuah method dari class induk
super.namaMethod();
```

Gambar 1.107 Memanggil method tertentu dari kelas induk dengan **super**

Mari teman-teaman coba mengubah class Person menjadi seperti di bawah ini :

```
public class Person {  
    public Person() {  
        System.out.println("Eksekusi konstruktor Person()");  
    }  
  
    public void showMessage(){  
        System.out.println("Hello, ini pesan dari class Person...");  
    }  
}
```

Gambar 1.108 Mengubah class Person

Kemudian class **Employee** diubah menjadi :

```
public class Employee extends Person {  
  
    public Employee() {  
        // eksekusi konstruktor induk  
        super();  
  
        System.out.println("Eksekusi konstruktor Employee()");  
    }  
  
    public void info(){  
        super.showMessage();  
    }  
}
```

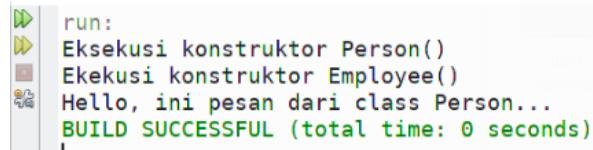
Gambar 1.109 Mengubah class Employee

Lalu pada class **Demo** diubah menjadi :

```
public class Demo {  
    public static void main(String[] args) {  
        Employee dian = new Employee();  
  
        dian.info();  
    }  
}
```

Gambar 1.110 Mengubah class Demo

Kemudian coba eksekusi class **Demo** , maka hasil outputnya akan seperti ini :



```

run:
Eksekusi konstruktor Person()
Eksekusi konstruktor Employee()
Hello, ini pesan dari class Person...
BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 1.111 Hasil output class Demo

Perhatikan class **Employee** pada konstruktornya kita mengeksekusi konstruktor induk dengan **super()**

Lalu pada method **info()** kita mengeksekusi method **showMessage()** dari induk dengan **super ShowMessage()**. Jika berhasil, maka akan mendapatkan hasil seperti diatas.

1.21.15.3 Penerapan this dan super pada Dunia Nyata Pertama, teman-teman perhatikan program untuk membuat GUI pada tutorial JFrame yang ada pada google. Contoh :

```

import javax.swing.JFrame;

public class MainWindow extends JFrame {

    public MainWindow(){
        this.setTitle("Jendela Utama");
        this.setSize(600,320);
    }

    public static void main(String[] args) {
        MainWindow mWindow = new MainWindow();
        mWindow.setVisible(true);
    }
}

```

Gambar 1.112 Membuat class **MainWindow** dengan meg-extends dari class **JFrame**

Penjelasan program diatas : Pada program ini, teman-teman membuat class **Main-Window** dengan meng-extends dari class **JFrame**.

```
public MainWindow(){  
    this.setTitle("Jendela Utama");  
    this.setSize(600,320);  
}
```

Gambar 1.113 this mewakili class MainWindow

Pada program tersebut, **this** digunakan untuk mewakili class **MainWindow** di dalam konstruktornya. Method **setTitle()** dan **setSize()** didapatkan dari class Induk (**JFrame**) karena tadi kita meng-extends dari class induk. Mengapa tidak menggunakan **super**? Karena, yang ingin diubah terletak pada **title** dan **size**-nya dari class **MainWindow**, bukan class **JFrame**. Contoh :

Coba perhatikan class **MainActivity** pada **Android**

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

Gambar 1.114 Contoh MainActivity pada Android

Nah, di dalam class **MainActivity** tersebut, kita menggunakan **super** untuk memanggil method **OnCreate()** dari class induk (**AppCompatActivity**). Lebih jelasnya seperti gambar di bawah ini :

```
super.onCreate(savedInstanceState);
```

Gambar 1.115 **super** untuk memanggil method **onCreate** dari class induk **AppCompatActivity**

Selanjutnya, teman-teman gunakan **this** pada **Toast.makeText().this** pada program ini untuk mewakili class **MainActivity**

Kesimpulan :

Yang perlu dipahami dalam **this** dan **super** adalah :

1. **this** : Objek yang mewakili class itu sendiri
2. **super** : Objek yang mewakili class induk *super class*

1.21.16 Memahami Prinsip Polimorfisme dalam OOP

Pemrograman berorientasikan objek atau OOP, memiliki beberapa konsep penting yang harus dipahami. Diantaranya ada Class, Objek, Enkapsulasi, Inheritance dan lainnya. Polimorfisme adalah sebuah prinsip dalam biologi di mana organisme atau spesies dapat memiliki banyak bentuk atau tahapan *stages*.

Prinsip ini bisa diadopsi pada pemrograman berorientasikan objek. Sehingga menjadi seperti berikut :

Polimorfisme dalam OOP adalah sebuah prinsip di mana class dapat memiliki banyak **"bentuk"** method yang berbeda-beda meskipun namanya sama. **"Bentuk"** disini dapat diartikan yaitu isinya berbeda, parameternya berbeda, dan tipe datanya berbeda. Polimorfisme pada Java ada dua macam, yaitu :

1. Static Polymorphism (Polimorfisme statis)
2. Dynamic Polymorphism (Polimorfisme dinamis)

Perbedaan dua macam polimorfisme terdapat pada cara membuat polimorfismenya, yaitu :

1. Polimorfisme statis menggunakan **method overloading** sedangkan polimorfisme dinamis menggunakan **method overriding**.
2. Polimorfisme dinamis biasanya terjadi saat menggunakan **pewarisan(inheritance)** dan implementasi *interface* sedangkan polimorfisme statis hanya terjadi di dalam satu class saja.

Perbedaan Method Overloading dengan Method Overriding

Method Overloading terjadi pada **sebuah class** yang memiliki **nama method yang sama** tapi memiliki **parameter dan tipe data yang berbeda** sedangkan method Overriding akan memiliki nama method yang sama , tapi isi dan parameter berbeda dari class induk.

1.21.17 Memahami Interface di Java

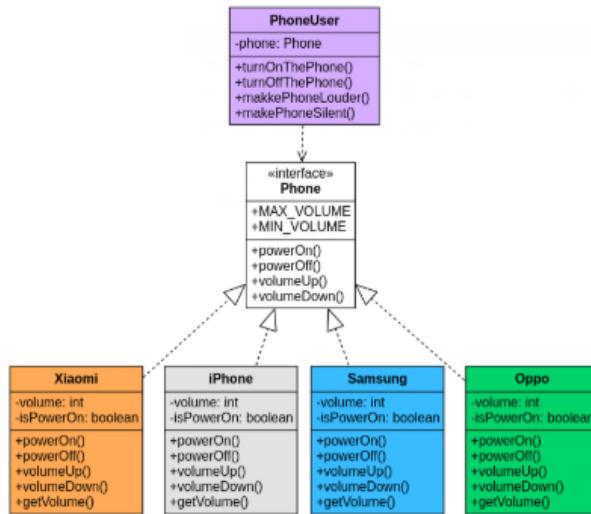
Interface adalah *blueprint* dari class. Isi method-nya kosong dan nanti akan diimplementasikan pada class lain. Singkatnya, Interface berasal dari kata inter = antar dan face = muka. Jadi, Interface adalah antarmuka.

Secara umum fungsi *interface* sebagai penghubung antara sesuatu yang abstrak dengan sesuatu yang nyata.

Contoh Program Interface:

Contoh hubungan antar dua objek yaitu manusia dengan handphone. Manusia bisa kita sebut sebagai PhoneUser karena dia akan dijadikan objek HP.

Maka teman-teman bisa buat class diagramnya sebagai berikut :



Gambar 1.116 Class Diagram PhoneUser

Setelah selesai membuat class diagram nya, teman-teman bisa membuat class baru di dalam **default package** dengan nama phone dan isi kodennya seperti di bawah:

```

public interface Phone {
    int MAX_VOLUME = 100;
    int MIN_VOLUME = 0;

    void powerOn();
    void powerOff();
    void volumeUp();
    void volumeDown();
}
  
```

Gambar 1.117 Contoh Interface class Phone

Selanjutnya teman-teman bisa membuat class baru dengan nama **PhoneUser** dengan isi kode nya seperti di bawah :

```
public class PhoneUser {  
  
    private Phone phone;  
  
    public PhoneUser(Phone phone)  
        this.phone = phone;  
    }  
  
    void turnOnThePhone(){  
        this.phone.powerOn();  
    }  
  
    void turnOffThePhone(){  
        this.phone.powerOff();  
    }  
  
    void makePhoneLouder(){  
        this.phone.volumeUp();  
    }  
  
    void makePhoneSilent(){  
        this.phone.volumeDown();  
    }  
}
```

Gambar 1.118 Contoh Interface class Phone

Berikutnya setelah membuat class PhoneUser, selanjutnya teman-teman bisa buat class implementasi dari Interface Phone:

1. Kode program Xiomi

```

public class Xiaomi implements Phone {

    private int volume;
    private boolean isPowerOn;

    public Xiaomi() {
        // set volume awal
        this.volume = 50;
    }

    @Override
    public void powerOn() {
        isPowerOn = true;
        System.out.println("Handphone menyala...");
        System.out.println("Selamat datang di XIAOMI");
        System.out.println("Android version 29");
    }

    @Override
    public void powerOff() {
        isPowerOn = false;
        System.out.println("Handphone dimatikan");
    }

    @Override
    public void volumeUp() {
        if (isPowerOn) {
            if (this.volume == MAX_VOLUME) {
                System.out.println("Volume FULL!!!");
                System.out.println("sudah " + this.getVolume() + "%");
            } else {

```

Gambar 1.119 Kode program Xiomi

2. lanjutan kode program Xiomi

```

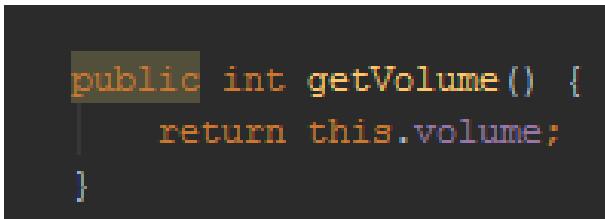
    }
            if (this.volume == MAX_VOLUME) {
                System.out.println("Volume FULL!!!");
                System.out.println("sudah " + this.getVolume() + "%");
            } else {
                this.volume += 10;
                System.out.println("Volume sekarang: " + this.getVolume());
            }
        } else {
            System.out.println("Myalakan dulu donk HP-nya!!!");
        }
    }

    @Override
    public void volumeDown() {
        if (isPowerOn) {
            if (this.volume == MIN_VOLUME) {
                System.out.println("Volume = 0%");
            } else {
                this.volume -= 10;
                System.out.println("Volume sekarang: " + this.getVolume());
            }
        } else {
            System.out.println("Myalakan dulu donk HP-nya!!!");
        }
    }
}

```

Gambar 1.120 Lanjutan Kode Program Xiomi

3. lanjutan kode program Xiomi



```

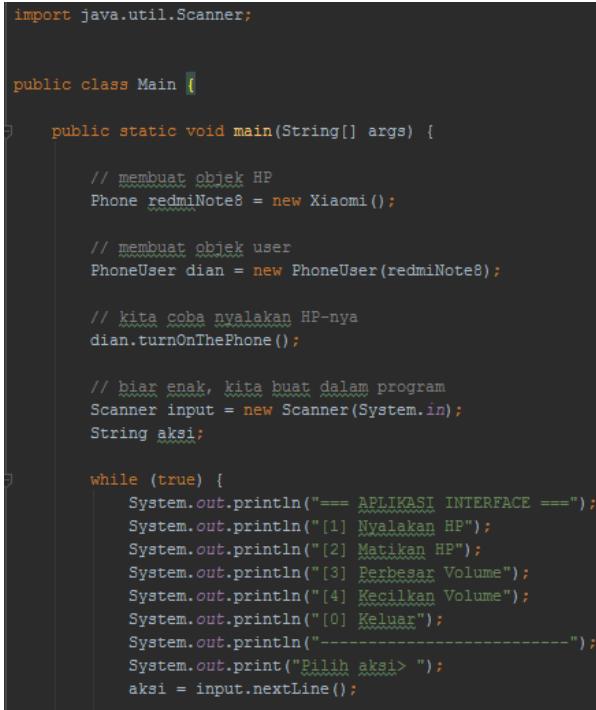
public int getVolume() {
    return this.volume;
}

```

Gambar 1.121 Lanjutan Kode Program Xiomi

Setelah selesai membuat kode program diatas, kemudian lanjut buat class Main agar program yang dibuat bisa di eksekusi

1. Main.java class Phone



```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // membuat objek HP
        Phone redmiNote8 = new Xiaomi();

        // membuat objek user
        PhoneUser dian = new PhoneUser(redmiNote8);

        // kita coba nyalakan HP-nya
        dian.turnOnThePhone();

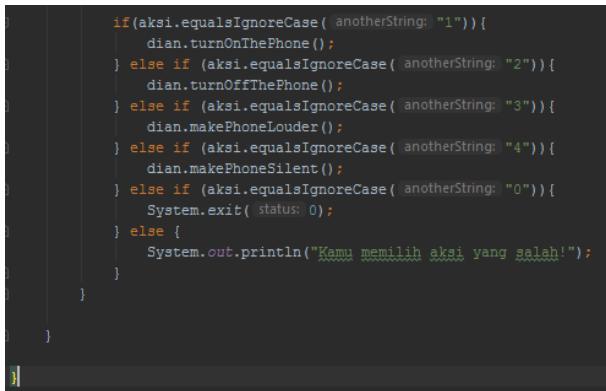
        // biar enak, kita buat dalam program
        Scanner input = new Scanner(System.in);
        String aksi;

        while (true) {
            System.out.println("== APLIKASI INTERFACE ==");
            System.out.println("[1] Nyalakan HP");
            System.out.println("[2] Matikan HP");
            System.out.println("[3] Perbesar Volume");
            System.out.println("[4] Kecilkan Volume");
            System.out.println("[0] Keluar");
            System.out.println("-----");
            System.out.print("Pilih aksi> ");
            aksi = input.nextLine();
        }
    }
}

```

Gambar 1.122 Main.java class Phone

2. lanjutan Main.java pada class Phone



```

        if(aksi.equalsIgnoreCase( anotherString: "1")){
            dian.turnOnThePhone();
        } else if (aksi.equalsIgnoreCase( anotherString: "2")){
            dian.turnOffThePhone();
        } else if (aksi.equalsIgnoreCase( anotherString: "3")){
            dian.makePhoneLouder();
        } else if (aksi.equalsIgnoreCase( anotherString: "4")){
            dian.makePhoneSilent();
        } else if (aksi.equalsIgnoreCase( anotherString: "0")){
            System.exit( status: 0);
        } else {
            System.out.println("Kamu memilih aksi yang salah!");
        }
    }
}

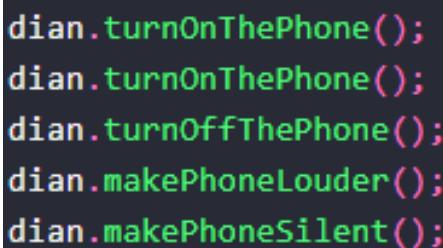
```

Gambar 1.123 Main.java class Phone

Setelah itu teman-teman bisa menjalankan kode program tersebut dengan perintah **Run File** atau bisa juga menekan **Shift + F6**.

Nah, di dalam method **main()**, tadi hanya menggunakan objek dari class **PhoneUser** saja.

Coba lihat kode-kode di bawah ini :



```

dian.turnOnThePhone();
dian.turnOnThePhone();
dian.turnOffThePhone();
dian.makePhoneLouder();
dian.makePhoneSilent();

```

Gambar 1.124 Method yang terhubung dengan Interface

Nah, disini saat teman-teman memanggil method ini maka method yang ada pada objek **redmiNote8** yang akan dipanggil. Ini karena mereka terhubung *interface*.

1.21.17.1 Hal-hal yang tidak boleh dilakukan di Interface

Ada beberapa hal yang tidak boleh dilakukan saat membuat interface :

1. Jangan buat variabel di dalam interface, tapi membuat konstanta boleh.
2. Jangan mengisi method-nya, cukup tuliskan nama method, tipe data, dan parameter saja. Tapi untuk default metod boleh punya isi.
3. Jangan berikan modifier **private** atau apun **protected** pada method dan konstanta yang ada di dalam interface.

4. Interface tidak bisa dibuat objek instance-nya dengan kata kunci **new**.

Jadi, jika hal-hal tadi dilanggar maka program akan error. **Kesimpulan:**

Interface merupakan penghubung antar objek. Interface bersifat abstrak, sehingga objek yang menggunakananya tidak akan peduli bagaimana ia diimplementasikan. Interface sebenarnya mengamalkan prinsip Abstraksi dan Enkapsulasi(pembungkusan).

1.21.18 Mengenal Class Abstrak dan Cara Pakainya

Class Abstrak adalah class yang masih dalam bentuk abstrak. Karena bentuknya masih abstrak dan dia tidak bisa dibuat langsung menjadi objek. Sebuah class agar dapat disebut class abstrak setidaknya **memiliki** satu atau lebih **method abstrak**. Method abstrak adalah method yang tidak memiliki implementasi atau tidak ada bentuk konkritnya. Agar class abstrak dapat digunakan, maka harus diubah ke bentuk konkret. Cara membuat class abstrak menjadi konkret adalah dengan membuat implementasi dari method-metho yang masih abstak dan bisa dilakukan dengan **perwarisan(inheritance)**. Mengapa class harus dibuat abstrak? Karena, pada kondisi tertentu class induk tidak ingin dibuat sebagai objek, karena methodnya belum jelas akan diimplementasikan seperti apa, oleh karena itu class sebaiknya dijasikan abstrak. Contoh class abstrak :

```
// ini abstrak method
void sayHello();

// ini bukan abstrak method karena
// punya implementasi di body method
void greeting(){
    System.out.println("Hello Java");
}
```

Gambar 1.125 Contoh class abstrak

```
abstract class Database {
    abstract String getDatabaseName();
    abstract String getTableName();
}
```

Gambar 1.126 Contoh membuat class abstrak

1.21.18.1 Cara membuat class abstrak di Java

1.21.18.2 Perbedaan class abstrak dengan Interface Class Abstrak dan Interface sama-sama digunakan untuk membuat abstraksi. Keduanya hampir memiliki sifat yang sama. Bedanya yaitu :

1. Di class abstrak kita bisa buat properti atau variabel sedangkan di interface kita cuma bisa buat konstanta saja.
2. Di class abstrak kita bisa implementasikan kode method seperti class biasa, sedangkan di interface harus menggunakan **default**.
3. Di class abstrak dapat memiliki member **private** dan **protected** sedangkan interface harus publik semua.
4. Class abstrak diimplementasikan dengan pewarisan (**extends**) sedangkan interface menggunakan **implements**.

1.21.19 Mengenal Classs Anonymus di Java

Class Anonymus adalah class yang **tidak memiliki nama**. Biasanya dinuati hanya untuk **sekali pakai** saja. Pada dsarnya class anonymous sama seperti class biasa. Biasanya class anonymous dibuat untuk mengimplementasikan **interface** dan **class abstrak** secara langsung.

Mengapa harus menggunakan class anonymous? Anonymous class biasanya dibutuhkan saat tidak ingin membuat implementasi dari interface dan class abstrak dalam file Java yang berbeda.

Contoh Program dengan Class Anonymus

1. Pertama buat nama proyek terlebih dahulu.
2. Kemudian buat interface baru pada package dengan nama **Clicable** dan isi kode nya seperti ini :

```
interface Clickable {  
    void onClick();  
}
```

Gambar 1.127 Buat nama interface

3. Setelah itu buat class **Button** seperti ini :

```
public class Button {
    private Clickable action;

    void setClickAction(Clickable action){
        this.action = action;
    }

    void doClick(){
        action.onClick();
    }
}
```

Gambar 1.128 Buat Class Button

4. Setalah itu buat clas **Main** seperti di bawah :

```
public class Main {

    public static void main(String[] args) {

        Button btn = new Button();

        // membuat class anonymous untuk implementasi interface
        btn.setOnClickListener(new Clickable() {
            @Override
            public void onClick() {
                System.out.println("Tombol sudah diklik!");
                System.out.println("Yay!");
            }
        });

        // mencoba klik tombol
        btn.doClick();
    }
}
```

Gambar 1.129 Buat Class Main

5. Setelah semua sudah selesai, lanjut dijalankan kode programnya

```
run:  
Tombol sudah diklik!  
Yay!  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Gambar 1.130 Hasil class anonymus

1.21.19.1 Menggunakan Anonymus Objek

Jika ada class anonymus, pasti ada objek anonymus. Objek anonymus adalah objek yang tidak memiliki nama dan biasa digunakan sekali pakai saja. Sebenarnya tadi diatas teman-teman sudah menggunakan objek anonymus sebagai parameter method **setActionClick()**.

```
// jika tidak menggunakan anonymous objek  
// maka kita harus bikin instance baru dari anon  
Clickable clickAction = new Clickable() {  
    // kode implementasi di sini  
}  
  
// baru setelah itu dipakai menjadi partameter  
btn.setOnClickListener(clickAction);  
  
// Tapi jika menggunakan anonymous objek,  
// kita tak perlu bikin objek baru.  
// langsung saja ditulis seperti ini  
  
btn.setOnClickListener(new Clickable(){  
    // kode implementasi di sini  
});
```

Gambar 1.131 Contoh objek anonymus

Tapi, teman-teman juga bisa menggunakan class anonymus secara langsung tanpa harus membuat objek baru.

```

interface Talkable{
    void sayHello();
}

// implementasi dan langsung digunakan
new Talkable(){

    @Override
    void sayHello(){
        System.out.println("Hello World!");
    }

}.sayHello();
// ^ karena ini anonymous objek,
//   jadi kita bisa langsung panggil methodnya

```

Gambar 1.132 Menggunakan class anonymous secara langsung

Jadi, dengan kata kunci **new**, teman-teman bisa membuat anonymous objek. Contoh class anonymous dipakai class biasa:

```

// membuat class biasa
class Hello {
    void saySomething(){
        System.out.println("Something..");
    }
}

// menggunakan class Hello untuk membuat anonymous objek
new Hello().saySomething();

```

Gambar 1.133 Menggunakan class anonymous secara langsung

Kesimpulan :

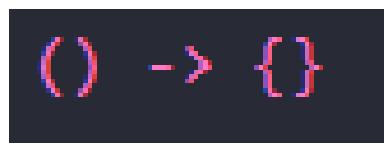
Class Anonymus sangat membantu saat teman-teman buuh implementasi langsung dari interface dan class abstrak sehingga mampu menghemat waktu karena tidak perlu membuat class baru lagi secara terpisah. Dan kode program juga terlihat lebih bersih dan mudah terbaca.

1.21.20 Mengenal Lambda Expression untuk Membuat Fungsi Anonymous di Java

Fungsi anonymous adalah fungsi yang tidak memiliki nama. Fungsi anonymous di Java dikenal juga dengan **lambda expression**. Artinya, saat teman-teman membuat

fungsi anonymous, teman-teman akan mengeksekusinya saat itu juga. Tidak bisa dipanggil lagi seperti fungsi biasa.

Simbol-simbol yang perlu diingat ;



Gambar 1.134 Simbol pada Lambda Expression

1. () tempat menaruh paramter.
2. (operator lambda) simbol yang mendandakan fungsi ini seperti simbol panah adalah lambda atau anonymous.
3. body fungsinya.

Contoh:

```
// ini fungsi biasa
int jumlahkan(int a, int b){
    return a + b;
}

// ini fungsi anonymous
(int a, int b) -> { return a + b }
```

Gambar 1.135 Simbol pada Lambda Expression

Mengapa harus pakai fungsi anonymous? Lambda expression atau fungsi anonymous sebenarnya hadir untuk menyempurnakan **class anonymous**. Saat interface hanya memiliki satu method saja untuk diimplementasikan. Teman-teman harus membuat class (anonymous) baru. Padahal teman-teman cuma butuh method-nya saja. Contoh program dengan lambda expression:

```

public class Main {

    public static void main(String[] args) {

        Button btn = new Button();

        // membuat class anonymous untuk implementasi interface
        btn.setOnClickListener(() -> {
            System.out.println("Tombol sudah diklik!");
            System.out.println("Yay!");
        });

        // mencoba klik tombol
        btn.setOnClickListener();
    }
}

```

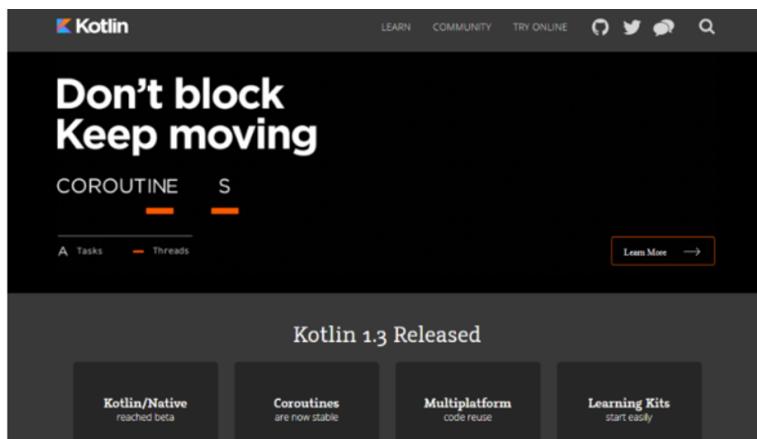
Gambar 1.136 Contoh program Lambda Expression

Setelahnya , teman-teman bisa mnegakses di laptop masing-masing.
Kesimpulan:

Fungsi anonymous hanya bisa digunakan saat ingin mengimplementasikan interface yang memiliki satu method. Jika interface itu memiliki lebih dari satu method, maka sebaiknya pakai class anonymous.

1.22 Bahasa Kotlin

Bahasa Pemrograman Kotlin



Gambar 1.137 Bahasa pemrograman Kotlin

Kotlin diciptakan oleh JetBrains yaitu perusahaan yang terkenal membuat IDE seperti : Android Studio, RubbyMine, PHPStrome, dll.

Kotlin sengaja diciptakan oleh JetBrains untuk melengkapi segala kekurangan dari bahasa pemrograman Java. Memang benar bahwa bahasa pemrograman kotlin lebih simple dibandingkan Java.

1.22.1 Keunggulan Kotlin

Keunggulan lainnya dari bahasa Kotlin yaitu bahasa ini bisa berjalan beriringan dengan bahasa pemrograman Java.[3] Dan juga bisa menggunakan library dari Java. Dan daya tarik dari Kotlin adalah hasil kompilasinya dalam bentuk *bytecode* JVM atau JavaScript. Selain itu Kotlin juga dibuat oleh praktisi industri, bukan dari pihak akademis, sehingga lebih memahami kebutuhan yang diperlukan. Mempelajari Kotlin tidak memerlukan *learning curve* yang lama, cukup beberapa jam saja untuk memahami bahasa yang digunakan. Karena bahsa Kotlin emang sangat sederhana, bisa dikatakan Kotlin merupakan Swift-nya Android.

1.22.2 Fitur Unggulan Kotlin

Bahasa pemrograman yang relatif baru mengedepankan produktifitas, oleh sebab itu Kotlin hadir untuk memudahkan dalam pembuatan kode program. Berikut merupakan fitur-fitur dari Kode Program ;

1. **Aman dari Null**, tidak perlu lagi untuk mengkhawatirkan akan timbulnya *Null Pointer Exception*. *Compiler* secara sistematis akan menandai *pointer* yang berpotensi *null*.
2. **Pemrograman Fungsional**, mendukung *lambda* dan kemampuan *mapping*, *folding*, pada *Collection* Java.
3. **Anotasi Data**, yang otomatis dibuat untuk mengurangi *boilerplate* seperti *equals*, *hashCode*, *toString*.
4. **Syntax Ramping**, yang menawarkan penulisan *syntax* secara cepat. Membuat fungsi *one-liner* membuat Java Beans hanya dengan satu baris, *method setter* dan *getter* otomatis.
5. **Ekstensi Fungsi**, yang memungkinkan untuk menambahkan *method* pada *class* tanpa harus mengubah *source code*-nya.
6. **Interpolasi String** yang memungkinkan untuk memasukkan variabel ke dalam String tanpa penyambungan seperti pada EcmaScript.

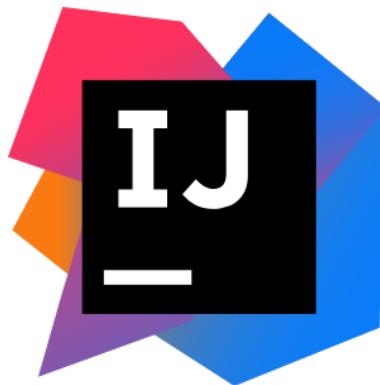
1.22.3 Persiapan Pengembang

Kebutuhan yang perlu disiapkan untuk pengembangan aplikasi android dengan Kotlin adalah :

1. Android Studio 2.2+
2. Gradle 2.2+
3. Plugin Kotlin 1.0.6+

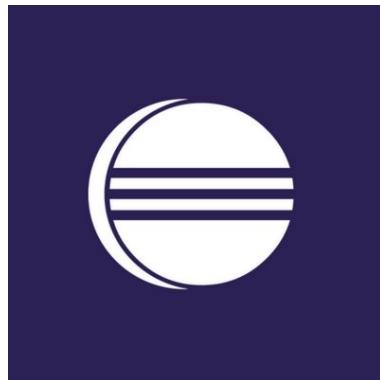
Pembuatan aplikasi android saat ini bisa menggunakan IDE :

1. IntelliJ IDEA



Gambar 1.138 IntelliJ IDEA

2. Android Studio
3. Eclipse



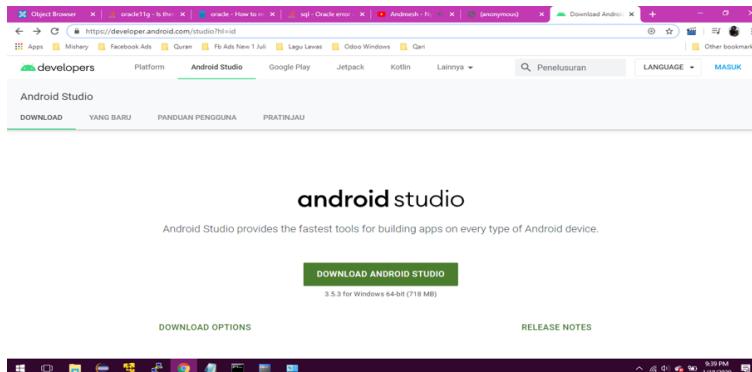
Gambar 1.139 IDEA Eclipse

1.23 Cara Instalasi Android Studio pada windows

Sebelum mengedit alangkah baiknya diinstall terlebih dahulu editornya seperti sebagai berikut :

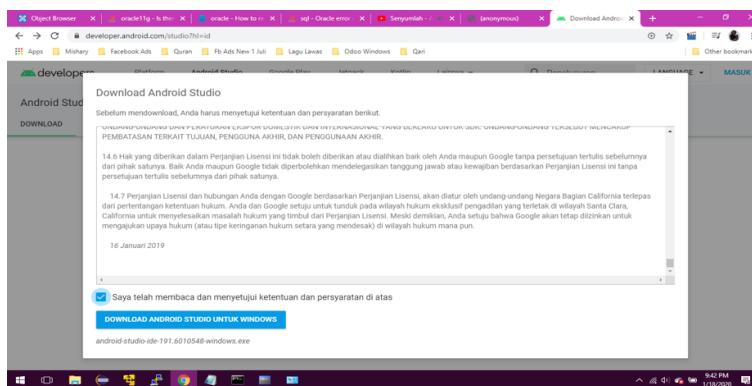
1. Download installernya terlebih dahulu berikut merupakan linknya :

<https://developer.android.com/studio?hl=id>. Kemudian lakukan Double-click pada tulisan download android studio seperti pada gambar



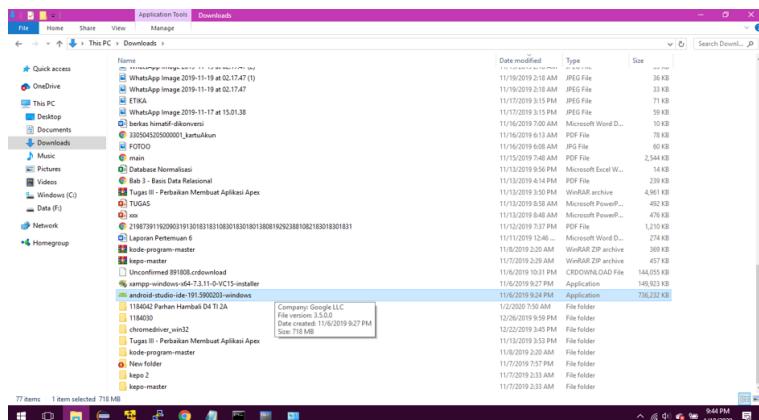
Gambar 1.140 Langkah pertama untuk install Android Studio, buka website resmi Android Studio

2. Maka akan muncul halaman awal installer



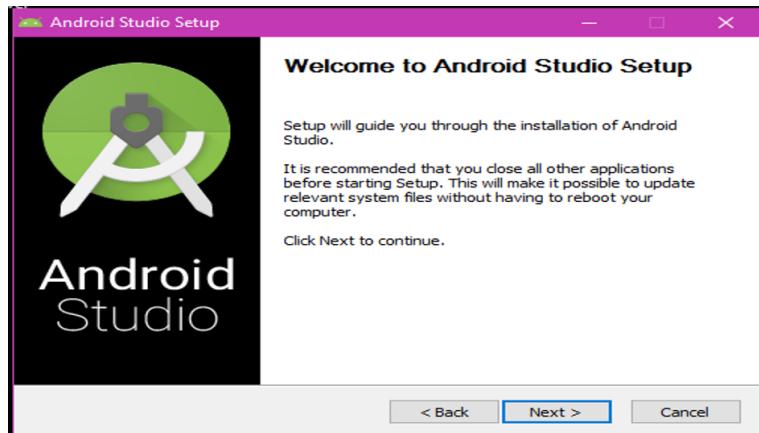
Gambar 1.141 Halaman persetujuan Install

3. Klik *Next* maka akan muncul Halaman install scope



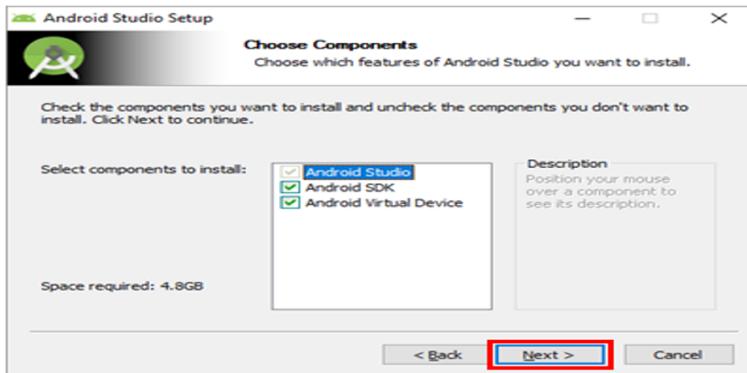
Gambar 1.142 Halaman Installasi Scope

4. Klik *Next* maka akan muncul Halaman dari tampilan Android Studio



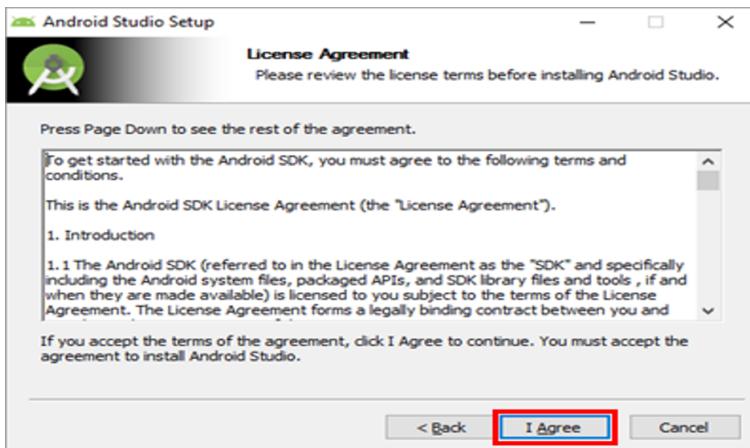
Gambar 1.143 Klik Next

5. Klik *Next* untuk melanjutkan proses instalasi.



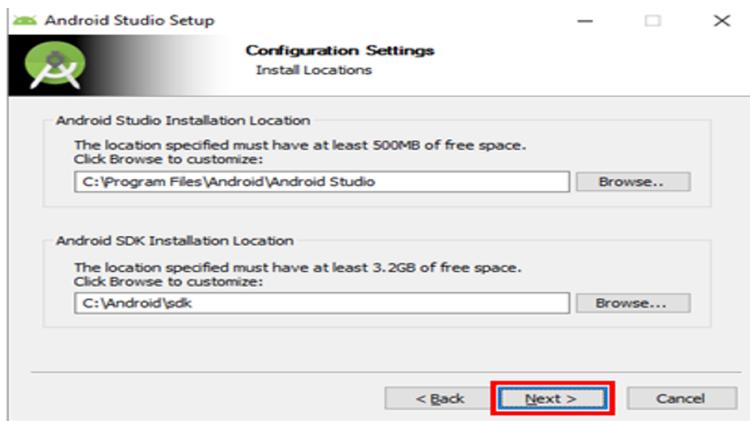
Gambar 1.144 Pilih komponen Android Studio

6. Klik *Next* maka akan muncul halaman persetujuan Andorid SDK



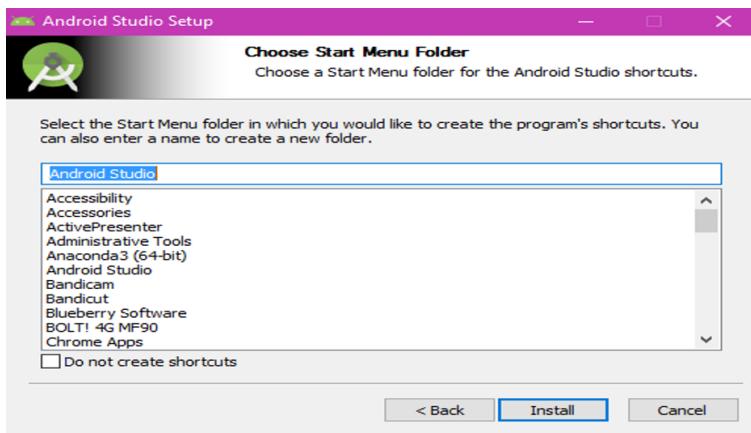
Gambar 1.145 Klik I Agree

7. Pilih lokasi untuk penyimpanan File Andorid Studio dan SDK Android Studio lalu klik *Next*



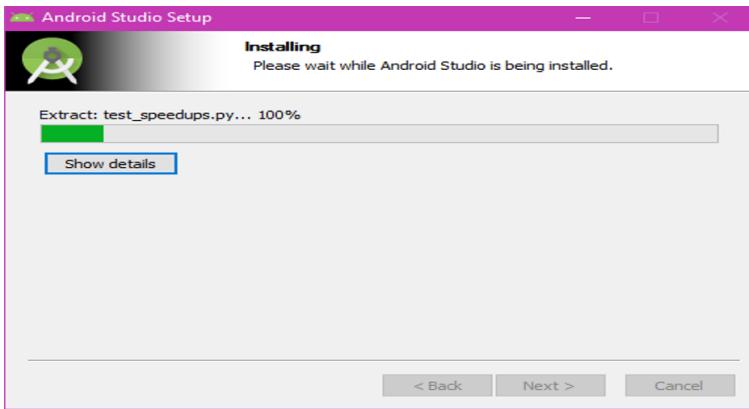
Gambar 1.146 Klik Next

8. Pilih Android Studio pada Menu Folder lalu klik install



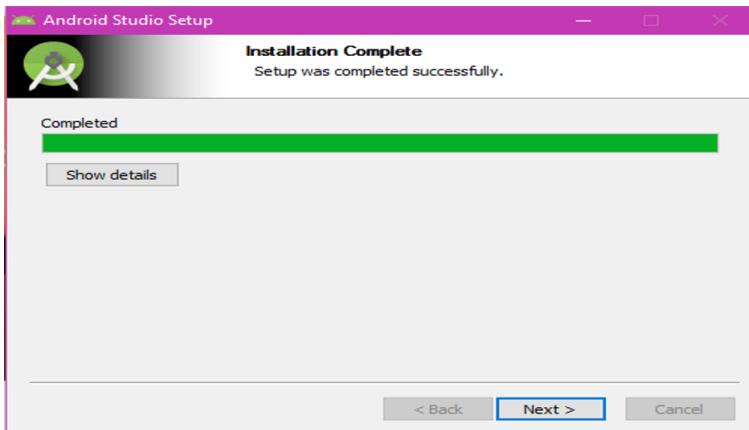
Gambar 1.147 Klik install

9. Klik *Install* maka akan muncul halaman seperti pada gambar, installasi sedang berjalan dan tunggu hingga proses loading berhenti.



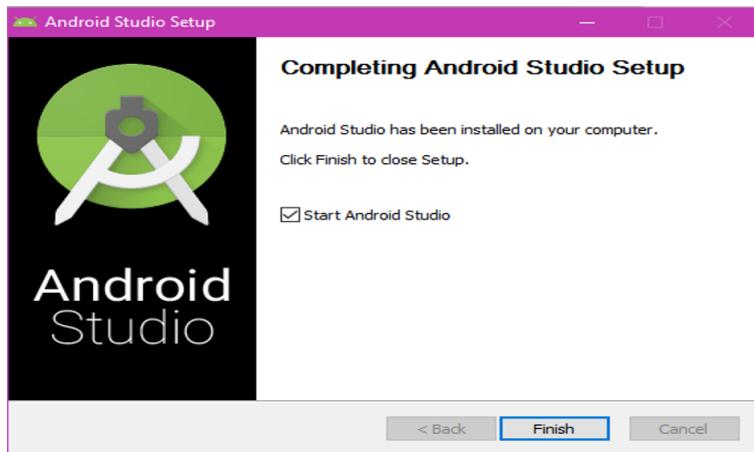
Gambar 1.148 Installasi sedang berjalan

10. Kemudian klik Next setelah proses installasi selesai



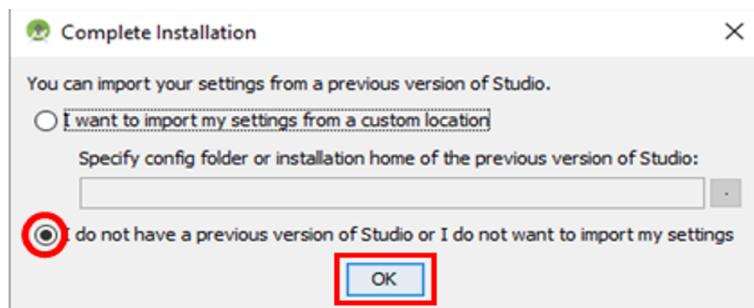
Gambar 1.149 Klik Next

11. Klik *Next* maka akan muncul halaman seperti pada gambar, maka installasi siap digunakan.



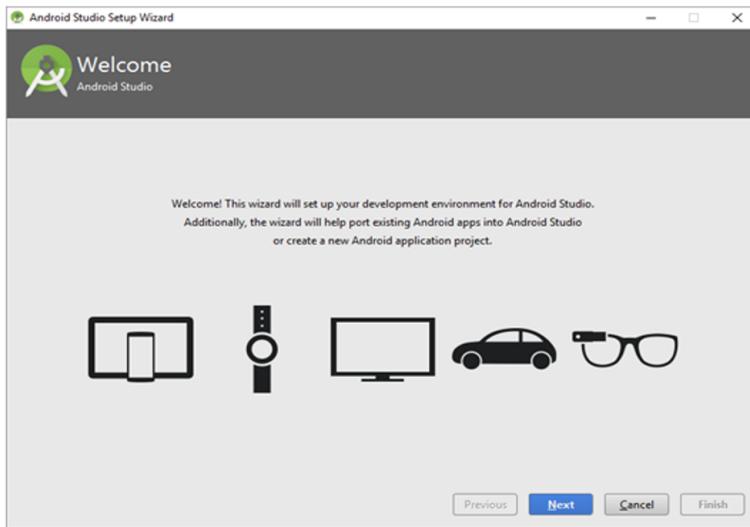
Gambar 1.150 Installasi siap digunakan

12. Setelah selesai installasi makann di lanjut untuk custom location.



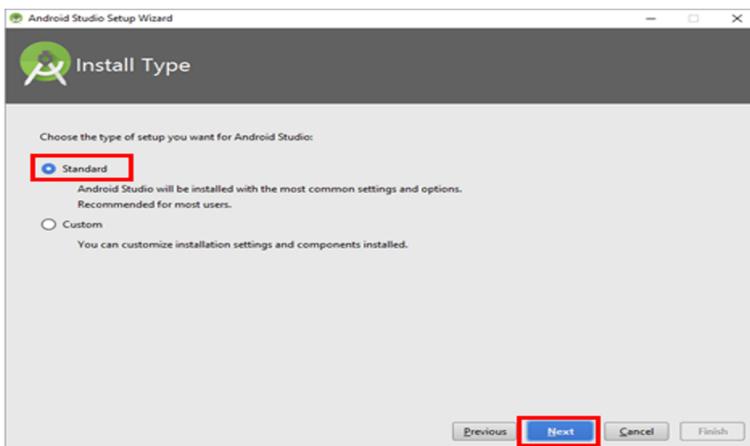
Gambar 1.151 Klik Ok

13. Setelah selesai installasi maka di lanjut pada halaman welcome Android Studio



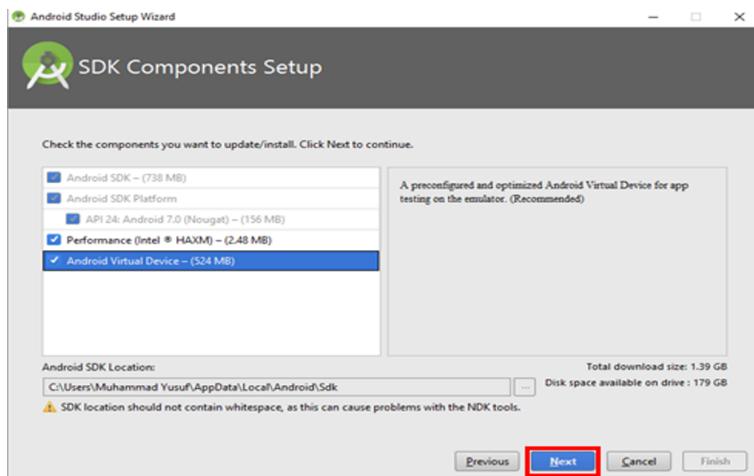
Gambar 1.152 Klik Next

14. Kemudian akan muncul halaman sebagai berikut

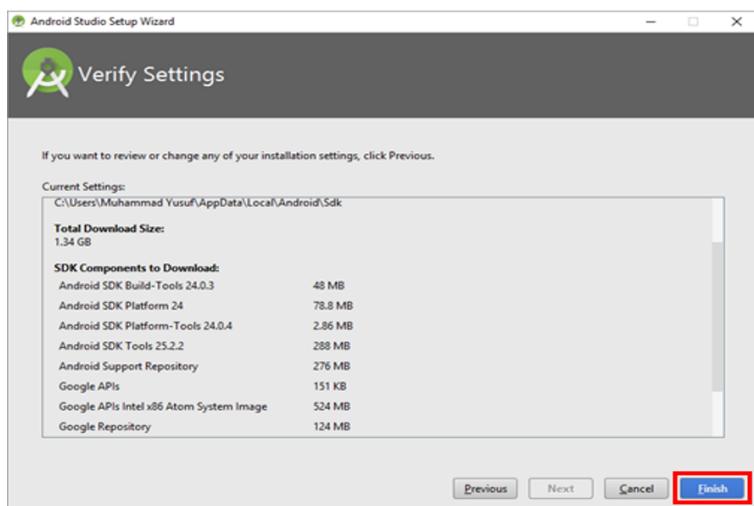


Gambar 1.153 Pilih Standard dan Next

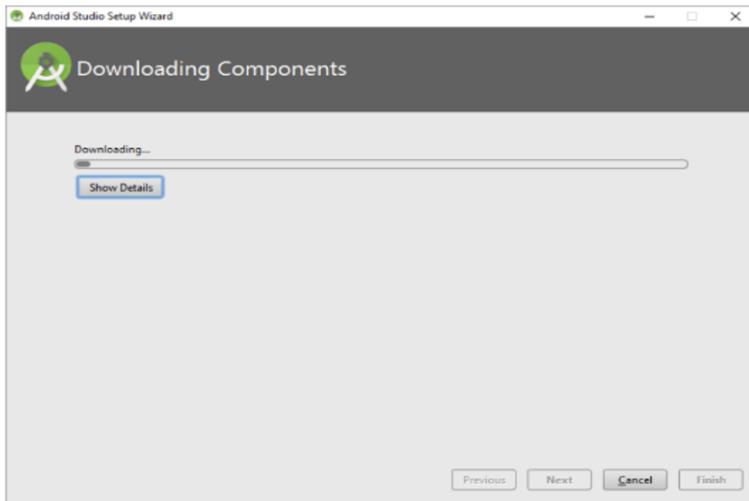
15. Klik *Next* untuk setup SDK komponen, lalu pilih Android Virtual Disk.

**Gambar 1.154** Klik Next

16. Klik Next maka akan muncul halaman seperti gambar dan klik Finish

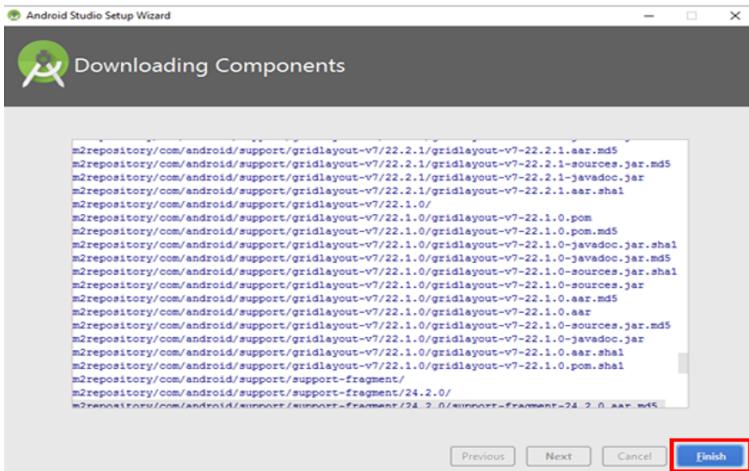
**Gambar 1.155** Klik Next

17. Klik Next maka akan muncul halaman seperti pada gambar dan tunggu proses download selesai.



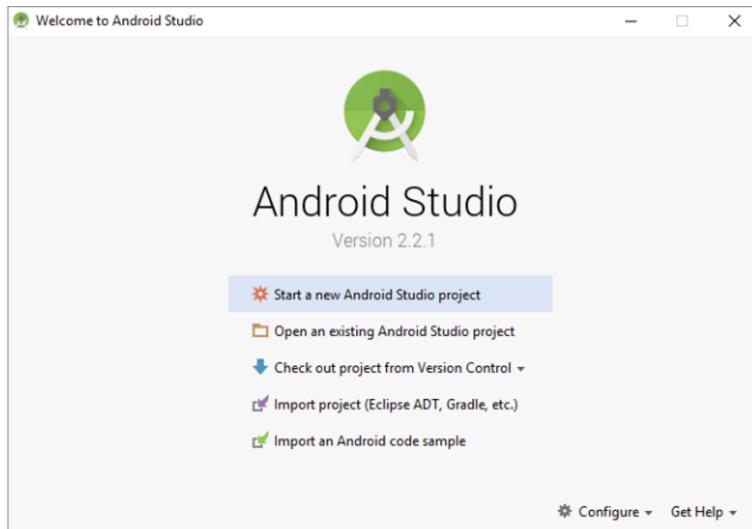
Gambar 1.156 Klik Finsih

18. Klik *Next* maka akan muncul halaman seperti pada gambar, tunggu hingga proses downloadng selesai.



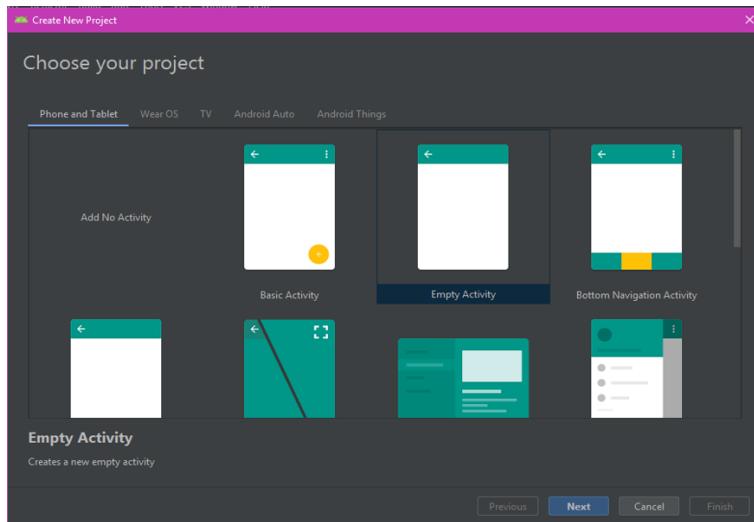
Gambar 1.157 Tunggu proses downloadng selanjutnya klik finish

19. Klik *finish* maka akan muncul halaman seperti gambar dan android siap digunakan



Gambar 1.158 Klik Finish

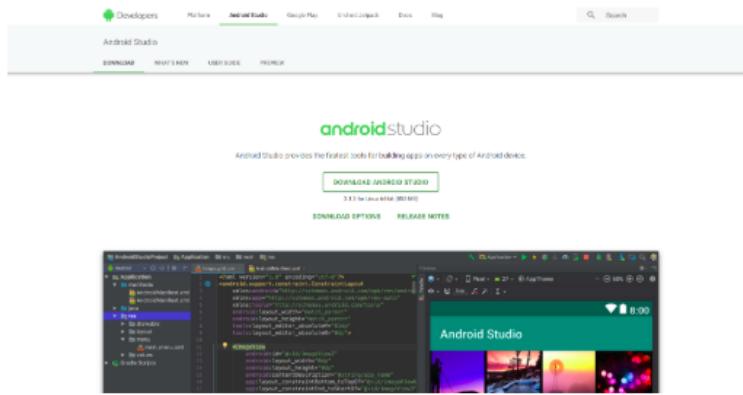
20. Klik start a new Android Studio Project lau android studio siap digunakan.



Gambar 1.159 Andrid Studio siap digunakan

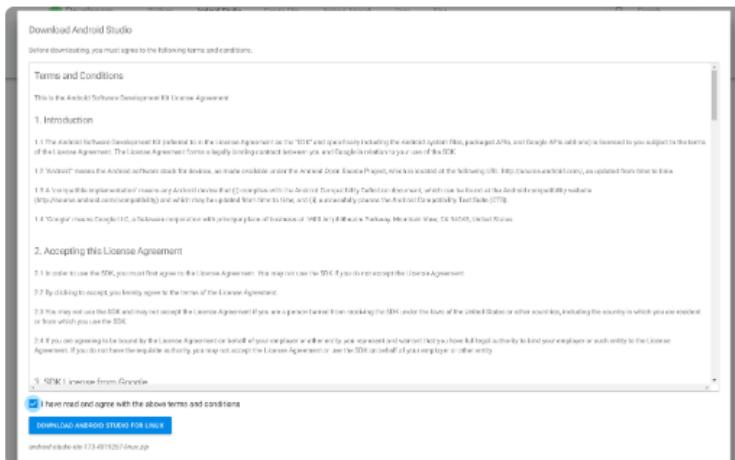
1.24 Cara Instalasi Android Studio pada Linux

1. Buka halaman <https://developer.android.com/studio/> lalu klik tombol Download Android Studio.



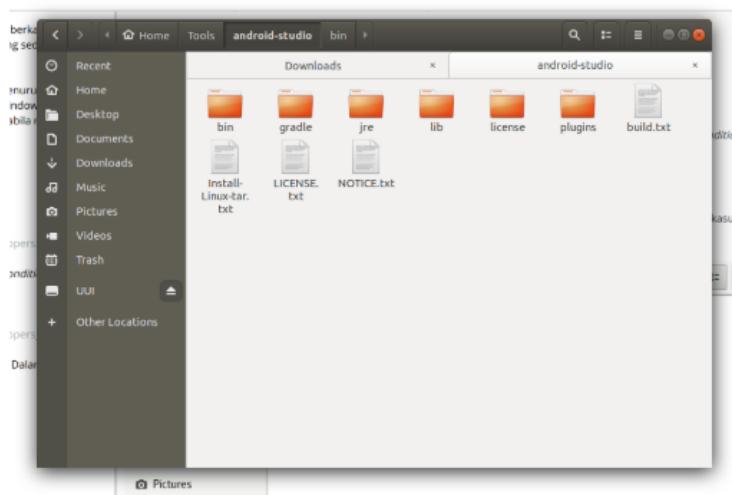
Gambar 1.160 Halaman untuk mendownload android studio

2. Kemudian, di popup yang muncul, beri tanda centang pada bagian I have read and agree with the above terms and conditions dan klik tombol Download Android Studio for Linux.



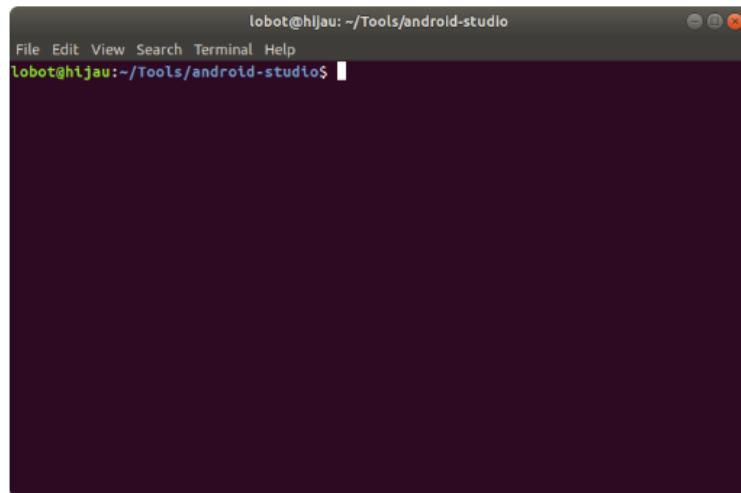
Gambar 1.161 Halaman untuk popup dan beri tanda centang pada bagian I have read and agree with the above terms and conditions

3. Tunggu hingga selesai. Setelah selesai, ekstrak file zip yang telah diunduh di folder manapun di direktori home. Dalam kasus kami, Android Studio disimpan di folder /Tools/android-studio.



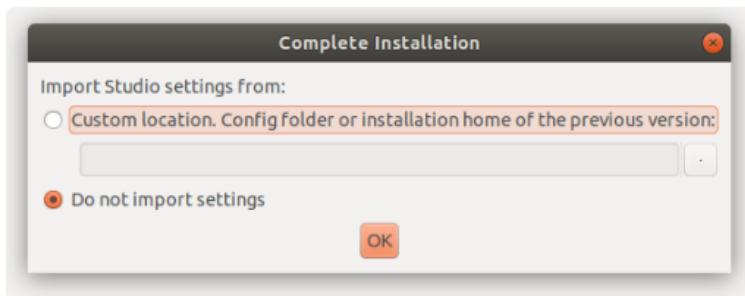
Gambar 1.162 Ekstrak file zip yang telah diunduh

4. Masih di aplikasi File Manager, klik kanan di folder android-studio yang belumnya sudah di ekstrak, lalu pilih Open in Terminal.



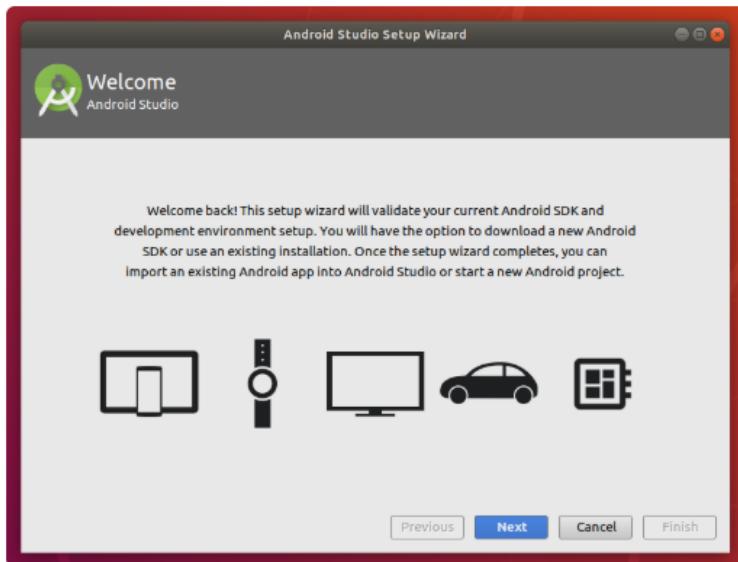
Gambar 1.163 Terminal untuk Android Studio

5. Di jendela baru yang muncul, ketikkan ./bin/studio.sh dan tekan ENTER.
6. Saat jendela Complete Installation muncul, pilih Do not import settings dan klik OK.



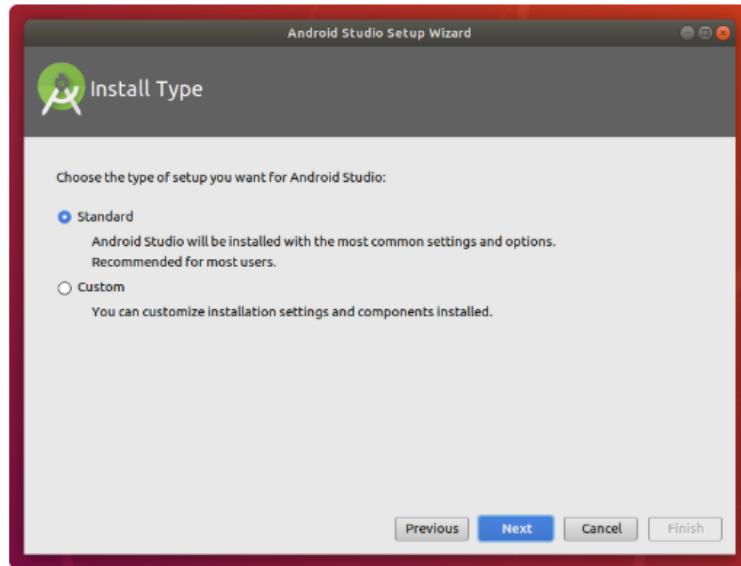
Gambar 1.164 Pilih Do Not import settings

7. Di jendela yang muncul klik tombol Next.



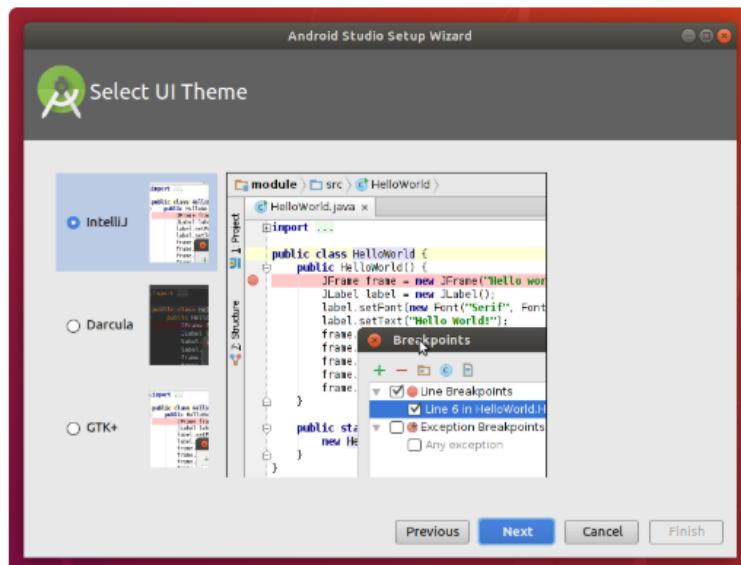
Gambar 1.165 Pilih Next pada jendela yang muncul

8. Selanjutnya pilih opsi Standard dan klik tombol Next



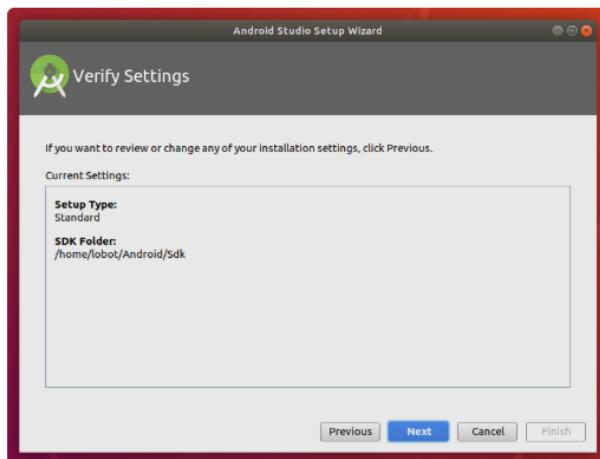
Gambar 1.166 Pilih Standar untuk type install Android Studio

9. Berikutnya kita akan diminta memilih UI Theme. Silahkan pilih yang paling disukai lalu klik tombol Next.



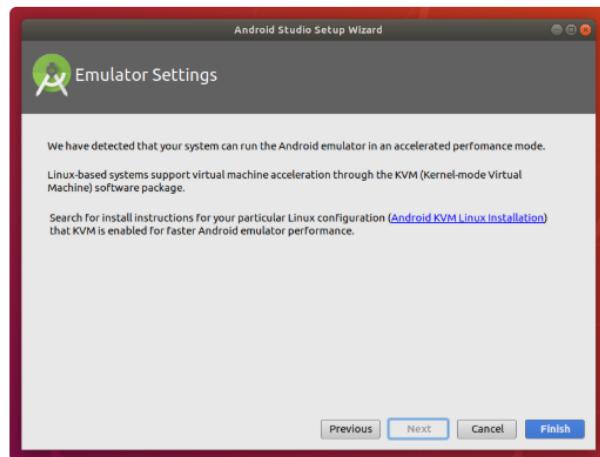
Gambar 1.167 Pilih UI Item untuk install Android Studio

10. Klik Next lagi.



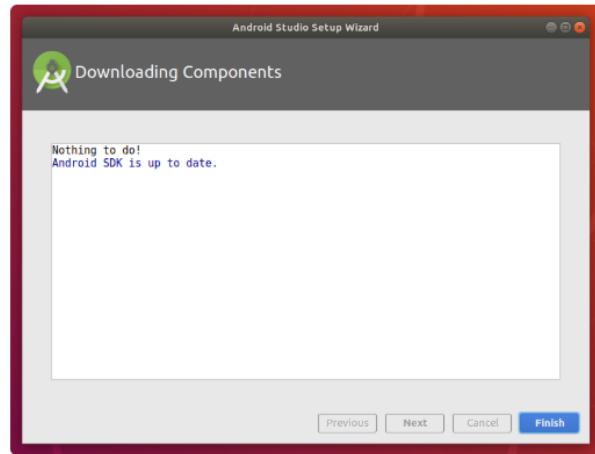
Gambar 1.168 Klik Next

11. Sampai di sini klik tombol Finish dan Android Studio akan mulai mengunduh file-file yang diperlukan.



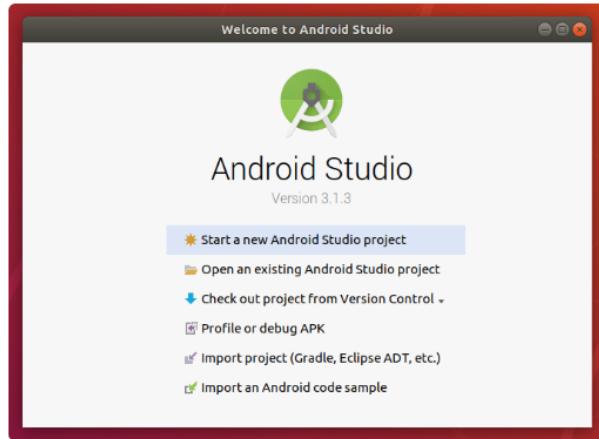
Gambar 1.169 Klik Finish

12. Setelah selesai, nanti kita bisa mengklik tombol Finish di jendela yang baru.



Gambar 1.170 Klik Finish

13. Tunggu hingga jendela Welcome to Android Studio muncul.



Gambar 1.171 Jendela Welcome to Android Studio

BAB 2

PENGENALAN WEB HOSTING

2.1 Pengertian Web Hosting

Web Hosting adalah sebuah komputer yang terhubung ke internet dan dipergunakan untuk menyimpan data website agar dapat diakses secara online.

Dengan memakai web hosting ini maka seluruh informasi yang disimpan dapat ditampilkan. Semua informasi yang disimpan di sebuah tempat disebut server web hosting. Untuk bisa tersambung ke internet dan dapat diakses oleh semua orang, server web hosting dikelola dalam ruang penyimpanan data bernama data center.

Istiahs web hosting sendiri merujuk pada set aktivitas atau layanan penyimpanan informasi suatu website hingga akhirnya bisa ditampilkan ketika Anda akses.

2.2 Cara Kerja Web Hosting



Gambar 2.1 Cara Kera web hosting

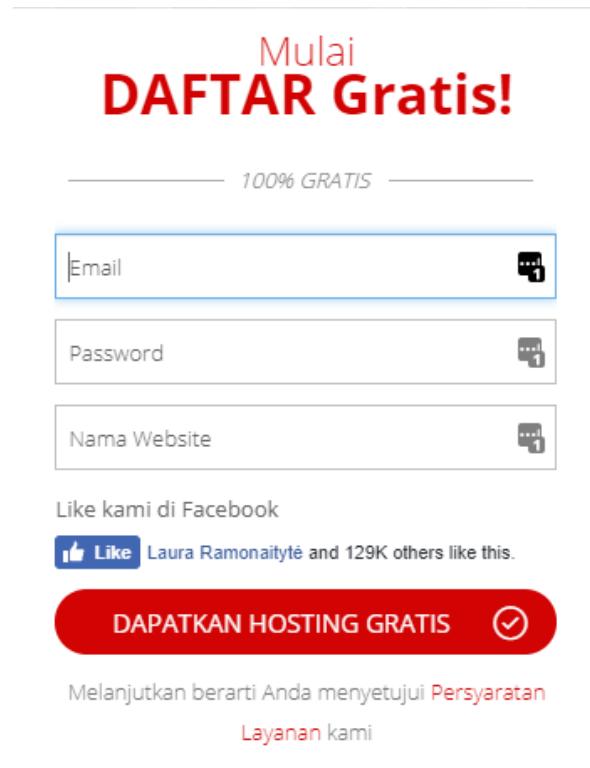
Ketika teman-teman ingin mengakses suatu website, maka teman-teman perlu mengetikkan alamat website pada browser yang teman-teman gunakan.

Kemudian, browser akan mengeksekusi perintah yang akan diteruskan dari internet ke server hosting sesuai permintaan. Hasilnya yaitu akan menampilkan gambar dan informasi website sesuai yang diakses dan akan diteruskan oleh internet agar tampil pada browser teman-teman. Jadi, begitulah cara kerja web hosting.

2.3 Cara membuat akun 000webhost

- Buka <https://id.000webhost.com/free-hosting>

2. Pilih opsi Daftar Hosting Gratis!. Anda akan diarahkan ke halaman di mana Anda harus memasukkan beberapa informasi penting, seperti alamat email, password, dan nama website. Klik Dapatkan Hosting Gratis.



Gambar 2.2 Masukan data-data untuk mendaftar webhost secara gratis

3. Setelah berhasil membuat akun, buka inbox dan cari email verifikasi. Kini Anda sudah bisa menggunakan layanan hosting gratis di 000webhost!

2.4 Fungsi Utama Web Hosting

1. Tempat Penyimpanan Data

Fungsi utama dari sebuah web hosting yaitu sebagai tempat penyimpanan data atau file dari sebuah aplikasi atau website secara online, sehingga bisa diakses oleh orang banyak dengan jaringan internet. Hosting dapat dikatakan sebagai sebuah perpustakaan dan website sebagai buku-bukunya, sehingga banyak orang yang bisa melihat atau membaca buku tersebut. Jadi pemilik website bisa memberikan informasi sesuai kebutuhan user.

2. Alamat Akses atau Link Website

Hosting juga berfungsi sebagai redireksi dari sebuah DNS. Contohnya ada sebuah situs 101.com, situs ini sudah teregistrasi dalam sebuah DNS. Ketika ada seseorang ingin mengakses situs tersebut, maka server akan mengakses link hosting yang digunakan dalam hosting. Jadi biarpun website atau situs kamu sudah terdaftar dalam DNS tetapi tidak dimasukkan dalam web hosting, website tersebut tidak akan bisa di akses oleh orang lain. Bahkan bisa memunculkan keterangan error saat ada orang ingin mengaksesnya.

3. Menjaga Website atau Aplikasi Tetap Online

Fungsi terakhir dari web hosting yang terakhir sangatlah penting untuk sebuah website atau aplikasi, bahkan fungsi ini hanya ada di hosting. Dimana website kamu akan selalu dalam kondisi online selama penyedia hosting website tetap terjaga dan tidak ada masalah, maka website kamu bisa di akses kapan saja dan dimana saja tanpa khawatir bila sewaktu-waktu mati. Walaupun ada beberapa kasus sebuah website mati, tetapi ini hanya sebagian kecil saja. Umumnya situs tersebut menggunakan web hosting free, maka ada baiknya menggunakan hosting berbayar yang lebih aman dan memiliki garansi untuk uptime. Bilamana website untuk keperluan bisnis atau profesi, gunakan web hosting sudah berpengalaman, sehingga berapa pun yang mengakses website tidak akan mendapat masalah. Sebelum memulai untuk web hosting, sebaiknya kamu mengetahui jenis-jenisnya. Jadi kamu bisa memilih hosting yang sesuai dengan kebutuhan website atau aplikasi.

2.5 Jenis-jenis dari Web Hosting

1. Virtual Hosting

Shared atau Virtual Hosting merupakan jenis web hosting yang paling umum digunakan oleh pemilik website. Dimana dalam sebuah server menawarkan beberapa hosting dan salah satu diantaranya miliki kamu sendiri serta beberapa hosting lain. Sementara untuk biaya hosting jenis virtual terbilang sangat murah. Oleh karena itu, banyak pengusaha ataupun perorangan lebih memilih hosting virtual, terutama untuk website yang digunakan setiap hari.

2. Free Hosting

Free hosting atau hosting gratis biasanya digunakan oleh pemula, sebab web hosting ini tidak memerlukan biaya sewa untuk hosting website atau aplikasi sesuai dengan namanya. Selain itu free hosting juga lebih minim resiko, terutama bila website hanya digunakan untuk sendiri.

Walaupun begitu, layanan ini biasanya akan menampilkan banyak iklan yang mengganggu saat mengaksesnya. Selain itu, free hosting memiliki beberapa keterbatasan seperti jumlah bandwidth, kapasitas penyimpanan, dan resiko website off. Jika ingin memilih web hosting untuk sebuah bisnis atau usaha pastikan backup database, layanan support, dan bandwidth yang digunakan oleh penye-

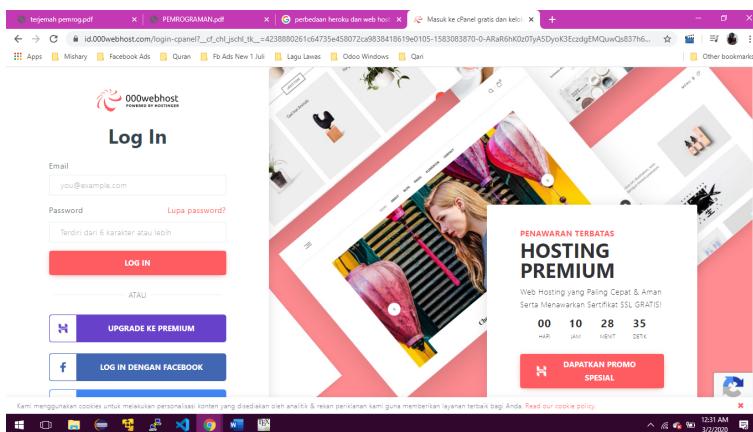
dia jasa. Jadi pastikan kebutuhan kamu bisa dipenuhi oleh jasa web hosting seperti IDCLOUDHOST, sebagai hosting terbaik.

2.6 Perbedaan Web Hosting dan Heroku

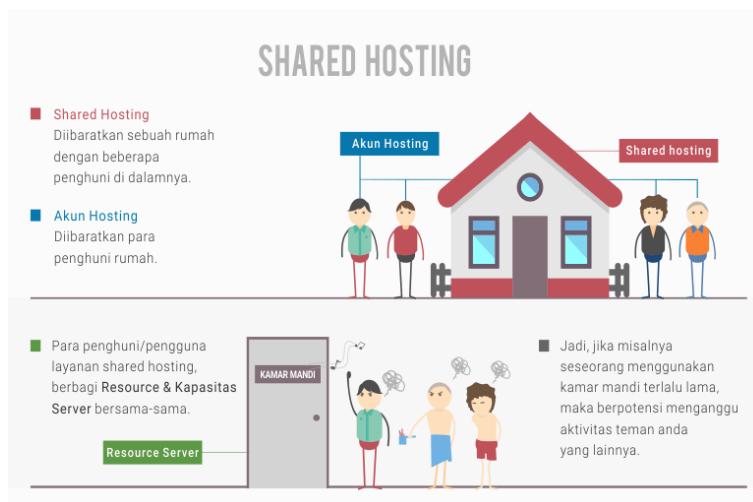
2.6.1 Web Hosting

Web hosting adalah layanan online untuk mengonlinekan website atau aplikasi web di internet. Ketika Anda membeli dan mendaftar di suatu layanan hosting, pada dasarnya Anda sedang meminjam space di server, tempat menyimpan semua file dan data yang dibutuhkan oleh website agar dapat bekerja sepenuhnya.

Server merupakan komputer fisik yang dijalankan tanpa adanya interupsi sehingga website Anda bisa diakses kapan saja oleh siapa pun. Web host bertugas untuk menjaga server agar tetap aktif dan berjalan, mengamankannya dari serangan cyber berbahaya, dan memindahkan konten (teks, gambar, file) dari server ke browser pengunjung situs Anda.



Gambar 2.3 Web hosting yang digunakan untuk membuat aplikasi Smart Gudang



Gambar 2.4 Imajinasi mempermudah cara kerja web hosting

2.6.2 Heroku

Heroku adalah sebuah cloud platform atau tempat penyimpanan yang menjalakan bahasa pemrograman tertentu. Sebelum mengenal lebih jauh mengenai Apa Itu Heroku dan Cara Deploy Proyek Pertama di Heroku sebaiknya kita juga perlu tahu apa itu Cloud Computing.

Cloud Computing adalah sebuah model untuk kenyamanan, akses jaringan on-demand untuk menyatukan pengaturan konfigurasi dari sumber daya komputasi seperti, jaringan, server, media penyimpanan, aplikasi, dan layanan yang dapat dengan cepat ditetapkan dan dirilis dengan usaha manajemen yang minimal atau interaksi dengan penyedia layanannya secara langsung.



Gambar 2.5 Ini heroku yang digunakan untuk membuat aplikasi kita

2.7 Web Hosting vs Heroku

2.7.1 Web Hosting

2.7.1.1 *Dari segi performa*

Web hosting menyediakan server yang digunakan oleh beberapa user sekaligus. Akibatnya kapasitas ruang penyimpanan serta bandwidth yang tersedia pun akan terbatas. Jadi tidak mengherankan jika kadang kala website butuh waktu lama untuk loading.

Di sisi lain, heroku tidak memiliki ruang penyimpanan fisik karena pada dasarnya layanan ini merupakan sebuah layanan virtual. Anda memang akan menggunakan layanan ini dengan user lain, namun jika terdapat kendala seperti loading yang lambat server heroku Anda akan segera dipindahkan oleh sistem. Hasilnya Anda tidak lagi merasakan kendala tersebut dan performa website pun bisa tetap prima.

2.7.1.2 *Dari segi kontrol data*

Pada web hosting Anda harus berbagi dengan user lain sehingga tidak ada kendali yang utuh terhadap server yang telah dibeli. Sebagai ilustrasi, apabila user lain menginginkan performa yang lebih baik maka website Anda pun akan berisiko menjadi lambat dan lebih susah diakses.

Lain halnya dengan heroku. Selama menyewa space tertentu dalam cloud server maka ruang tersebut berada dalam control Anda sepenuhnya. Jika dalam web hosting Anda bisa terdampak risiko performa web melambat karena bandwidth yang dibagi dengan user lain, maka hal tersebut tidak terjadi pada heroku. Ini karena server akan segera memindahkan Anda ke server yang lebih lapang saat bandwidth terbatas.

2.7.1.3 *Dari segi biaya*

Saat ingin menggunakan layanan web hosting Anda harus membayar server yang digunakan berikut dengan hardware. Padahal kenyataanya ruang tersebut juga digunakan oleh user lain. Biaya ini akan membengkak jika Anda memutuskan untuk membeli web hosting tipe dedicated.

Bagaimana dengan heroku? Dalam heroku, Anda cukup mengeluarkan dana sesuai dengan layanan yang didapat. Tidak perlu mengeluarkan dana tambahan untuk server yang mungkin tidak Anda gunakan. Jika diibaratkan, pembiayaan heroku serupa dengan membayar tagihan listrik atau biaya berlangganan operator seluler.

2.8 10 Kontrol Panel Hosting Populer Berbasis Open Source

Kontrol panel yang paling populer saat ini dan kuat brandingnya adalah cPanel dan Plesk. Kedua kontrol panel ini merupakan aplikasi berbayar yang dibayar setiap bulan bagi sebuah provider hosting untuk di install dalam servernya. Namun untungnya, ada beberapa kontrol panel alternatif yang bersifat open source yang tersedia untuk di download secara gratis dengan fitur hampir sama dengan yang berbayar, yaitu sebagai berikut:

1. Cpanel

Cpanel Adalah kontrol panel hosting yang berbasis Unix/Linux. Antarmuka grafisnya membantu Anda untuk mengelola website beserta account hosting Anda dengan sangat mudah dan cepat. Cpanel memberi Anda akses penuh atas berbagai elemen pengaturan dari situs web dan administrasi hostingnya melalui web browser misalnya seperti Membuat database, membuat account email, auto responder, dan mengelola file website.

2. Plesk

Plesk adalah control panel hosting yang mirip dengan cPanel. Plesk memungkinkan Anda untuk mengelola account hosting Anda melalui antarmuka berbasis web. Anda dapat menginstall kontrol panel ini didalam VPS atau dedicated server. Plesk juga memungkinkan Anda untuk mengontrol ribuan virtual host dalam satu mesin. Kontrol panel memungkinkan Anda untuk mengotomatisasi banyak tugas yang pada gilirannya mengurangi biaya dan sumber daya. Hal ini juga meningkatkan profitabilitas, efisiensi dan kepuasan pelanggan.

- (a) Membuat akun FTP.
- (b) Mengelola dan membuat akun email dan database seperti MySQL dan PostgreSQL.
- (c) Menambahkan domain dan subdomain.
- (d) Restore dan Backup data.
- (e) Mengelola DNS dan sumber daya lainnya.

3. ISPCConfig

ISPCConfig adalah kontrol panel open source multi bahasa yang memungkinkan Anda untuk mengelola beberapa server di bawah satu kontrol panel. ISPCConfig berlisensi di bawah lisensi BSD. Kontrol panel open source ini juga mampu mengelola FTP, SQL, BIND DNS, database dan virtual server.

Fitur yang disediakan oleh ISPCConfig adalah seperti berikut ini:

- (a) Dapat memanage lebih dari satu server dari satu panel kontrol.
- (b) Antarmuka web yang memudahkan untuk administrator, reseller dan klien login.
- (c) Mendukung webserver seperti Apache dan Nginx.
- (d) Konfigurasi mirroring dan cluster.
- (e) Mengelola akun email dan FTP.
- (f) Dan masih banyak lagi

4. Kloxo

Kloxo adalah salah satu kontrol panel website yang terbilang canggih dan disediakan secara gratis untuk distro Redhat dan CentOS. Memiliki fitur seperti FTP, spam filter, PHP, Perl, CGI, dan banyak lagi. Fitur seperti Messaging,

Backup restore dan modul Ticketing juga tersedia dalam kontrol panel tersebut. Ini membantu user untuk mengelola/menjalankan kombinasi Apache dengan BIND, dan beralih antarmuka antara program ini tanpa kehilangan data Anda.

5. Zpanel

Zpanel adalah kontrol panel hosting yang disediakan secara gratis dan sangat mudah digunakan pada kontrol panel webhosting kelas enterprise seperti Linux, UNIX, MacOS, dan Microsoft Windows. Zpanel ditulis dalam bahasa PHP murni dan berjalan dengan baik pada Apache, PHP dan MySQL. Muncul dengan serangkaian fitur inti penting untuk menjalankan layanan hosting web Anda. Fitur inti tersebut meliputi Apache Web Server, hMailServer, FileZilla Server, MySQL, PHP, Webalizer, RoundCube, phpMyAdmin, phpSysInfo, FTP Jailing dan masih banyak lagi.

6. Webmin

Webmin merupakan kontrol panel webhosting yang powerfull dan sangat fungsional. Software yang dirancang untuk platform Unix dan Linux dengan cara yang sederhana. Webmin cukup mampu untuk mengelola berbagai komponen lingkungan berbasis web dari pengaturan webserver untuk maintaining FTP dan Email Server.

Fitur yang disediakan pada Webmin, adalah sebagai berikut:

- (a) Mengkonfigurasi dan membuat server virtual pada Apache.
- (b) Mengelola, menginstal atau menghapus paket perangkat lunak (RPM format).
- (c) Untuk keamanan, Anda dapat menyetting fitur firewall.
- (d) Mengubah pengaturan DNS, alamat IP, konfigurasi routing.
- (e) Mengelola database, tabel dan field MySQL

7. EHCP

EHCP (Easy Hosting Control Panel) adalah software kontrol panel gratis untuk menjaga server hosting berbasis web. Dengan penggunaan EHCP Anda dapat mengelola database MySQL, account email, account domain, account FTP dan banyak lagi. Ini adalah satu-satunya control panel yang telah built-in support untuk Nginx dan PHP-FPM yang tidak menggunakan Apache dan memberikan kinerja yang baik untuk server low end.

8. DTC

Domain Technologie Control (DTC) adalah control panel hosting terutama untuk admin dan akuntansi layanan hosting GPL. Dengan bantuan interface web berbasis GUI, DTC dapat mendeklegasikan tugas seperti membuat email, account FTP, subdomain, database dan banyak lagi. Ia mengatur database MySQL yang berisi semua informasi hosting.

9. Interworx

Interworx adalah sistem manajemen server Linux dan kontrol panel webhosting. Interworx memiliki seperangkat tool yang memberikan kewenangan administrator untuk memerintah servernya sendiri dan end user dapat melihat atau meninjau hasil pengelolaan website mereka. Kontrol panel ini pada dasarnya dibagi menjadi dua mode operasi, yaitu:

- (a) Nodeworx, yaitu modus administrator yang membantu mengelolaan server.
- (b) SiteWorx, yaitu website owner view yang membantu end users untuk mengelola account mereka hosting dan fitur-fitur didalamnya.

10. Ajenti

Ajenti merupakan satu – satunya kontrol panel berbasis open source yang kaya fitur, kuat dan ringan. Kontrol panel yang menyediakan antarmuka web responsif untuk mengelola server kecil set-up dan juga paling cocok untuk Dedicated dan VPS hosting. Muncul dengan banyak built-in plugin untuk mengkonfigurasi dan mengelola perangkat lunak server dan layanan seperti Apache, Nginx, MySQL, FTP, Firewall, File System, Cron, Munin, Samba, Squid dan banyak program lainnya seperti File Manager, Kode Editor untuk developer serta akses Terminal.

2.9 Flowchart

2.9.1 Pengertian Flowchart

Flowchart (diagram alir) adalah suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. Flowchart biasanya digunakan sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kemudian diberikan kepada programmer, dengan begitu, flowchart dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, flowchart digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu, adapun untuk menghubungkan satu proses ke proses selanjutnya selanjutnya digambarkan dengan menggunakan garis penghubung.

Dengan adanya flowchart ini, setiap urutan proses dapat digambarkan secara jelas, maka dari itu ketika ada penambahan proses baru dapat dilakukan dengan mudah menggunakan flowchart ini, jika flowchart sudah selesai dibuat, maka giliran programmer yang akan menerjemahkan desain logis tersebut kedalam bentuk program dengan berbagai bahasa pemrograman yang telah disepakati.

2.9.2 Fungsi Flowchart

Flowchart memiliki fungsi diantaranya adalah untuk menggambarkan, menyederhanakan suatu rangkaian proses atau prosedur sehingga dapat mudah dipahami dan mudah dilihat berdasarkan urutan langkah dari suatu proses.

2.9.3 Simbol-simbol Flowchart

No.	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program
2		<i>Input / Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
3		<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya / tidak
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
7		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
8		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9		<i>Punch Tape</i>	
10		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer)
11		<i>Flow</i>	Menyatakan jalannya arus suatu proses

Gambar 2.6 Simbol flowchart dan penjelasannya

BAB 3

PENGENALAN XAMPP

3.1 XAMPP

3.1.1 Pengertian XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem p-namaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah cross platform sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris.

Sejarah mencatat, software XAMPP pertama kali dikembangkan oleh tim proyek bernama Apache Friends dan sampai saat ini sudah masuk dalam rilis versi 7.3.9 yang bisa didapatkan secara gratis dengan label GNU (General Public License).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini:

1. X = Cross Platform

Merupakan kode penanda untuk software cross platform atau yang bisa berjalan di banyak sistem operasi.

2. A = Apache

Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (open source).

3. M = MySQL / MariaDB

MySQL atau MariaDB merupakan aplikasi database server yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database.

4. P = PHP

Huruf "P" yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat website dinamis, contohnya dalam website berbasis CMS WordPress.

5. P = Perl

Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.

3.1.2 Fungsi XAMPP

Program aplikasi XAMPP berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya website offline. XAMPP bekerja secara offline layaknya web hosting biasa namun tidak bisa diakses oleh banyak orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para mahasiswa maupun pelajar untuk melihat hasil desain website sebelum akhirnya dibuat online menggunakan web hosting yang biasa dijual dipasaran.

3.1.3 Cara Install XAMPP di windows

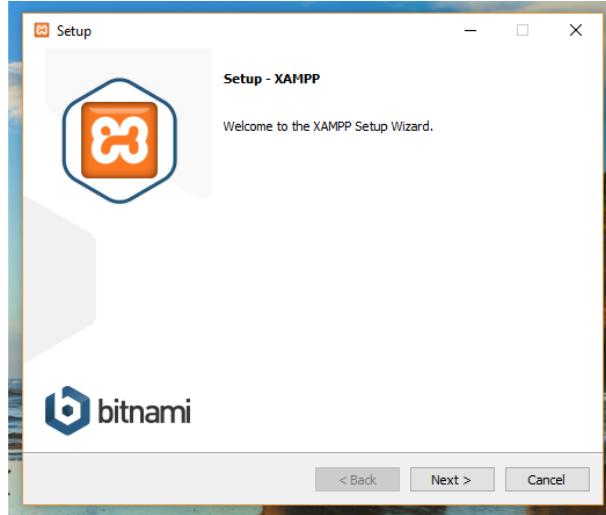
Setelah mengenal lebih jauh lewat pengertian XAMPP, teman-teman tentu penasaran dong bagaimana langkah-langkah cara menginstall XAMPP di Laptop atau PC? Berikut panduan instalasi XAMPP selengkapnya:

1. Teman-teman harus menyiapkan file XAMPP terlebih dahulu dengan cara mengunduh lewat link berikut ini <http://www.apachefriends.org/en/index.html>. File ini bersifat gratis, jadi teman-teman tidak perlu mengeluarkan biaya lisensi sepeserpun.



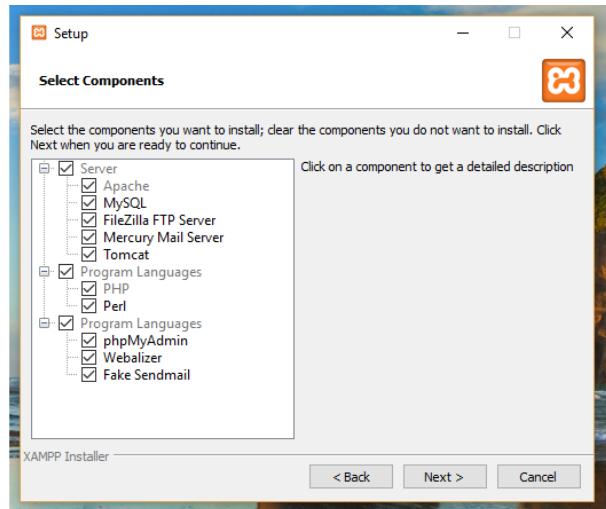
Gambar 3.1 Halaman download aplikasi XAMPP

2. Setelah file berhasil didownload dengan sempurna langkah selanjutnya adalah menyiapkan space/ruang hardisk di laptop atau komputer dengan minimal masih menyisakan ROM sebesar 1GB guna menampung gambar, teks, video, dan dokumentasi website.
3. Jalankan file xampp-windows-x64-7.3.9-0-VC15-installer atau versi lainnya yang lebih mutakhir.
4. Akan ada jendela baru yang terbuka sebagai indikasi dimulainya proses instalasi. Pada proses ini teman-teman akan diminta untuk memilih bahasa Indonesia atau English, pilih saja yang bahasa Indonesia kemudian klik next.



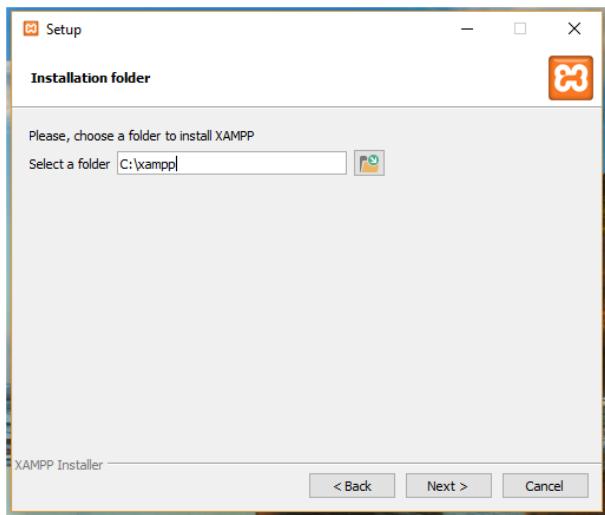
Gambar 3.2 Langkah pertama instal XAMPP

5. Setelah itu, teman-teman akan diminta untuk memilih komponen yang akan digunakan. Standar untuk server web berbasis CMS WordPress menggunakan MySQL, FileZilla FTP server, dan Apache, sedangkan dari bahasa pemrograman bisa menceklist pilihan phpMyAdmin, PHP, dan Perl. Kemudian klik next.



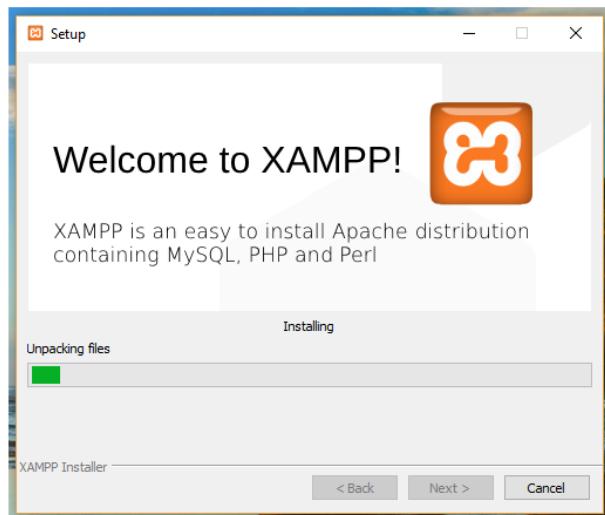
Gambar 3.3 Memilih komponen

6. Pada tahap ini kita diminta untuk memilih lokasi file XAMPP yang akan diinstall, gunakan saja pengaturan default lalu klik next/install.



Gambar 3.4 Lokasi file penyimpanan

7. Tunggu beberapa menit sampai proses instalasi XAMPP selesai yang ditandai dengan bar progress seluruhnya berubah menjadi warna hijau.

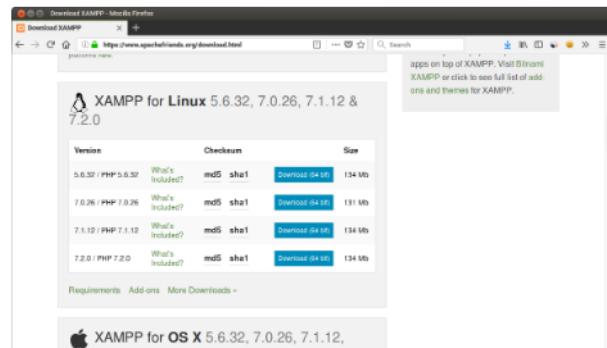


Gambar 3.5 Proses Install XAMPP

8. Teman-teman sudah bisa menjalankan program XAMPP di laptop atau PC Windows untuk membuat website secara offline menggunakan localhost.

3.1.4 Cara Installasi XAMPP pada Linux

1. Download XAMPP pada website resminya di <https://www.apachefriends.org/download.html>



Gambar 3.6 Pilih yang XAMPP 7

2. Setelah download selesai, kita akan mendapatkan file binari. File ini akan kita eksekusi dengan user root. Karena itu, teman-teman buka file manager sebagai root. Lalu, klik 2x pada file tersebut.



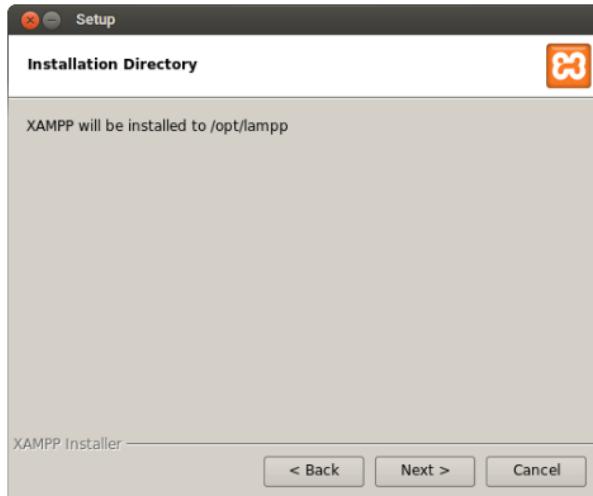
Gambar 3.7 Klik Next untuk melanjutkan

3. Teman-teman akan diminta untuk menentukan komponen yang akan diinstall. Biarkan saja, lalu klik next.



Gambar 3.8 Tentukan komponen

4. XAMP akan diinstall di direktori /opt/lampp/, klik next untuk melanjutkan.



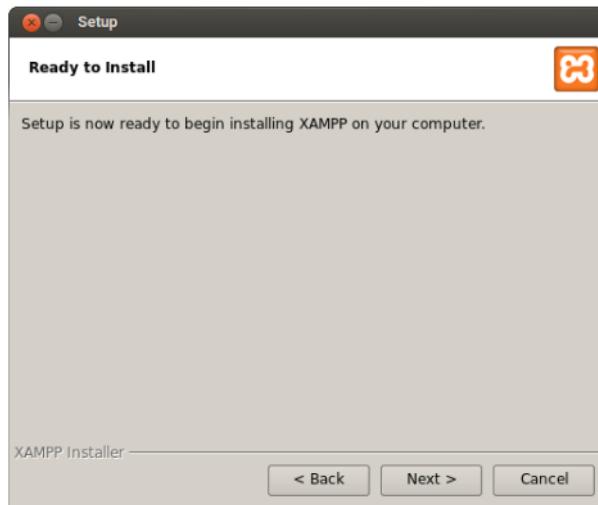
Gambar 3.9 Install XAMPP pada direktori

5. Lalu klik next pada bitnami for XAMPP



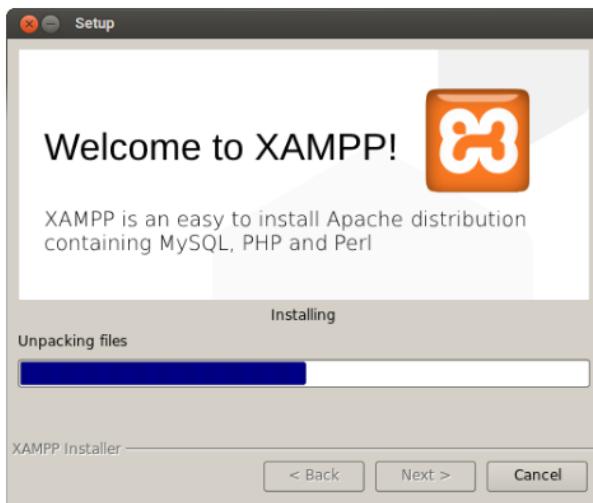
Gambar 3.10 Klik Next bitnami for XAMPP

6. Klik next untuk siap install



Gambar 3.11 Klik Next untuk siap install

7. Tunggu prosesnya hingga selesai



Gambar 3.12 Tunggu prosesnya sampe selesai

8. Klik finish untuk mengakhiri XAMPP



Gambar 3.13 Klik Finish

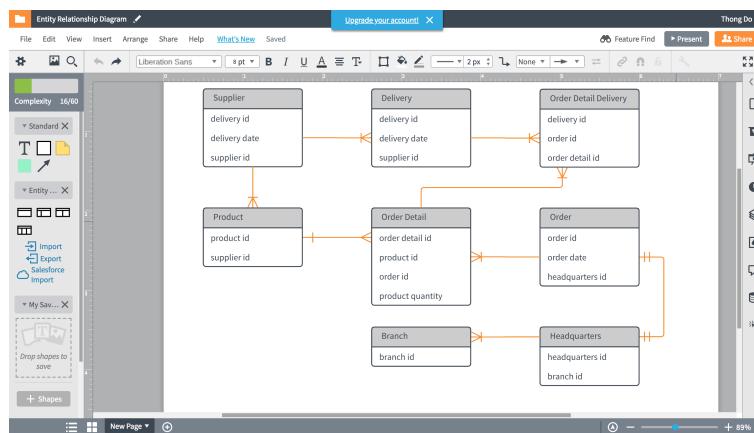
BAB 4

PENGENALAN DATABASE

4.1 Pengertian Database



Gambar 4.1 Halaman tampilan untuk Database



Gambar 4.2 Halaman tampilan untuk Relasi antar Data

Database atau basis data adalah kumpulan berbagai data dan informasi yang tersimpan dan tersusun di dalam komputer secara sistematis yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program komputer untuk mendapatkan informasi dari basis data tersebut.

Istilah database sendiri mengacu pada koleksi data-data yang saling terkait satu sama

lain dimana tujuan database dapat digunakan untuk mengelola data dengan lebih efektif dan efisien.

4.2 Pengertian Database menurut para ahli

1. Menurut Fathansyah (2007)

Basis data terdiri dari dua 2 kata, yakni Basis dan data. Basis kurang lebih dapat diartikan pula sebagai gudang atau markas, tempat bersarang atau berkumpul. Sedangkan Data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (Siswa, peagawai, pelanggan dan pelanggan), hewan, barang, peristiwa, keadaan, konsep, dan sebagainya yang terekam dalam bentuk huruf, simbol, teks, angka, gambar, bunyi, atau kombinasi lainnya

2. Menurut Indrajani (2011)

Berikut ini beberapa pengertian basis data, yakni :

(a) Sebuah kumpulan data yang berkaitan secara logis serta merupakan penjelasan dari data tersebut yang dibuat dengan tujuan agar menemukan data yang dibutuhkan oleh suatu perusahaan dan organisasi. Basis data juga dapat diartikan sebagai kumpulan data yang saling terintegrasi karena basis data dibuat untuk digunakan oleh banyak pengguna, memegang data operasional dan juga penjelasan tentang data tersebut, dan menghindari duplikasi data.

(b) Sebuah kumpulan elemen data yang terintegrasi serta berhubungan secara logika. Basis data menggabungkan berbagai catatan yang tadinya disimpan dalam file yang terpisah ke dalam suatu elemen data. Pendapat Connolly dan Begg (2010, p65), basis data merupakan suatu kumpulan data yang saling berelasi secara logika serta dirancang untuk melengkapi informasi yang dibutuhkan oleh organisasi.

3. Bambang Hariyanto (2008)

Basis Data ialah Kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena atau fakta secara terstruktur dalam domain tertentu agar mendukung aplikasi pada sistem-sistem tertentu.

4. Abdul Kadir dan Terra Ch. Triwayuni (2003)

Basis Data merupakan suatu pengorganisasian sekumpulan data yang saling berhubungan sehingga memudahkan aktivitas untuk mendapat informasi.

5. Linda Marlinda (2004)

Basis Data merupakan sebuah susunan dan kumpulan data operasional lengkap dari suatu organisasi serta perusahaan yang diorganisir atau dikelola lalu disimpan secara terintegrasi dengan memakai metode tertentu menggunakan komputer sehingga bisa menyediakan informasi optimal yang dibutuhkan oleh pemakainya.

6. Edhy Sutanta (2014)

Dalam bukunya yang berjudul Analisa Basis Data adalah sebagai berikut: “Basis data bisa dipahami sebagai suatu kumpulan data terhubung (interrelated data) yang disimpan dengan bersama-sama pada suatu media, tanpa mengatap satu dan yang lainnya atau tidak memerlukan suatu kerangkapan data (meskipun ada maka kerangkapan data itu harus seminimal mungkin dan terkontrol (controlled redundancy), data disimpan dengan cara tertentu sehingga mudah untuk digunakan dan ditampilkan kembali, data bisa digunakan satu atau bahkan lebih program-program aplikasi secara optimal, data disimpan tanpa ketergantungan dengan program yang akan menggunakannya, data disimpan sedemikian rupa sehingga proses pengambilan, penambahan, dan modifikasi data dapat dilakukan dengan sangat mudah dan terkontrol.”

7. Nugroho Adi (2005)

Dalam bukunya yang berjudul Konsep Pengembangan Sistem Basis Data menegaskan : “Basis data merupakan salah satu bagian dari sistem informasi secara keseluruhan.”

8. C.J. Date (2010)

Basis Data merupakan sekumpulan data persisten yang dipakai oleh system aplikasi dari perusahaan. System basis data pada dasarnya adalah suatu system penyimpanan record atau data yang terkomputerisasi.

9. Kusrini (2007)

Basis data merupakan kumpulan data yang berelasi. Data sendiri merupakan fakta mengenai obyek, orang dan lain-lain. Data dinyatakan dengan nilai (dengan karakter, angka atau symbol).

10. Connolly dan Begg (2010)

Basis data merupakan kumpulan data yang telah terbagi dan terhubung secara logikal serta deskripsi dari data yang dirancang untuk memenuhi keperluan informasi suatu organisasi.

4.3 Fungsi Database

Setelah memahami apa itu database, maka kita harus mengetahui apa itu fungsi database:

1. Mengelompokkan data dan informasi sehingga lebih mudah dimengerti
2. Mencegah terjadinya duplikat data maupun inkonsistensi data
3. Mempermudah proses penyimpanan, akses, pembaharuan, dan menghapus data.
4. Menjaga kualitas data dan informasi yang diakses sesuai dengan yang di-input.
5. Membantu proses penyimpanan data yang besar.
6. Membantu meningkatkan kinerja aplikasi yang membutuhkan penyimpanan data.

4.4 Manfaat Database

Berikut beberapa manfaat dari menggunakan database yang bisa didapatkan jika bekerja dengan sistem database:

1. Tidak terjadinya redundansi Basis Data

Database mampu meminimalkan terjadinya redundansi artinya redundansi sendiri itu merupakan terjadinya data-data ganda dalam berkas-berkas yang berbeda.

2. Integritas Data Terjaga

Database memastikan integritas data yang tinggi dimana database akan memastikan keakuratan, aksesibilitas, konsistensi dan juga kualitas tinggi pada suatu data.

3. Independensi Berbagai Data

Database menjaga independensi data dimana orang lain tidak dapat merubah data meskipun data bisa diakses.

4. Kemudahan berbagai Data

Menggunakan perangkat lunak database bisa digunakan untuk berbagi data atau informasi dengan sesama pengguna lainnya.

5. Menjaga Keamanan Data Database menjamin keamanan suatu informasi data, dimana anda bisa meyisipkan kode akses untuk data-data tertentu yang tidak bisa diakses bersama.

6. Kemudahan Akses Data

Dengan database bisa memudahkan untuk mengakses dan mendapatkan data karena semua data terorganisir dengan baik.

4.5 Tujuan Basis Data

1. Dalam melakukan penyimpanan cepat dan mudah, memanipulasi dan juga memunculkan kembali dari data tersebut.
2. Memberikan efisiensi ruang yang lega di memory penyimpanan karena dengan adanya basis data, redundansi data dapat terhindari
3. Akurasi (Accuracy) data.
4. Ketersediaan (Availability) data.
5. Kelengkapan (Completeness) data, dapat melakukan berupa perubahan struktur dalam basis data tersebut baik itu dari penambahan Objek baru (Tabel) hingga penambahan dari berbagai Field baru di Table
6. Security (Keamanan) data adalah dasar dari ketentuan di pengguna dimana boleh atau tidaknya dalam mengakses Basis Data serta berbagai objek yang

terdapat di dalamnya dan menentukan berbagai jenis apa saja yang boleh digunakan.

7. Kebersamaan Pemakai (Sharability), Pengguna dari basis data dapat digunakan lebih dari satu orang, namun tetap menjaga atau menghindari adanya masalah baru yang bisa saja terjadi di antaranya seperti inkonsistensi data hal ini dapat saja terjadi karena data yang sama di ubah oleh banyak pemakai di dalam waktu yang bersamaan dan juga kondisi deadlock hal ini tentu bisa juga dapat terjadi karena banyak pemakai yang saling menunggu (Antrian) untuk menggunakan data tersebut.

4.6 Maria DB

Disini kita akan mencoba untuk membahas berbagai macam tutorial mengenai Maria DB seperti cara install database MariaDB,Syntax,Tipe Data,koneksi,database,membuat database,memilih databsae,membuat tabel,operasi CRUD,cara insert,cara limit,cara update,cara delete,statement dan berbagai perintah bisa digunakan dalam MariaDB. Nah, sebelum masuk untuk mempelajari MariaDB ada baiknya jika teman-teman sudah mempelajari atau mengetahui dasar-dasar perintah MYSQL.

MariaDB adalah proyek berbasis komunitas dari sistem manajemen basis data relasional MYSQL. MariaDB adalah teknologi database open source dan relasional yang dapat digunakan sebagai pengganti MYSQL.Maria DB dikembangkan oleh pengembang asli MYSQL yang khawatir setelah MYSQL diakuisisi oleh Oracle.

Maria DB adalah relasional database manajemen sistem yang menyimpan data kedalam tabel-tabel yang ada didalam database.Primary Key dan Foreign Key digunakan untuk membangun relasi antar beberapa tabel yang berbeda.

Relasional databsae manajemen sistem (RDBMS) memiliki beberapa fitur seperti berikut ini:

1. RDBMS memfasilitasi Anda untuk menerapkan sumber data dengan tabel, kolom, dan indeks.
2. RDBMS menyediakan integritas referensi antar baris dari beberapa tabel.
3. Hal ini digunakan untuk secara otomatis untuk memperbarui indeks.
4. RDBMS dapat digunakan untuk menafsirkan query SQL dan operasi dalam memanipulasi atau sumber data dari tabel.

4.7 Istilah yang digunakan dalam RDBMS

Berikut ini adalah beberapa istilah yang digunakan dalam relasional database manajemen sistem pada MariaDB:

1. Database : Database merupakan suatu wadah yang berisi tabel-tabel yang berisi data

2. Table : Tabel merupakan struktur matrix yang berisi data.
3. Coloumn : Coloumn(kolom) adalah suatu elemen data. Coloumn merupakan suatu struktur yang menyimpan data dengan tipe yang sama.
4. Row : Row atau abris adalah struktur dimana suatu data disimpan, Row biasa disebut juga dengan tuple, entry atau record.
5. Primary Key : Primary Key merupakan suatu nilai yang unik. Nilai yang berupa Primary Key tidak dapat muncul dua kali didalam tabel yang sama.
6. Foreign Key : Foreign Key biasanya digunakan untuk menghubungkan dua buah tabel yang berbeda.

BAB 5

CONTOH STUDI KASUS

5.1 Latar Belakang

Aplikasi scan barcode barang masuk dan keluar pada warehousing berbasis android adalah aplikasi pengolahan data barang yang terdapat di dalam suatu sistem pergudangan, yang dirancang untuk mengontrol barang masuk dan barang yang terjual. Di aplikasi yang kami rancang akan dibuat tabel untuk menentukan barang masuk dan terjual terbanyak setiap bulannya. Pada proses yang dilakukan dalam mengelola warehousing bukanlah perkara yang mudah tapi tidak bisa dibilang cukup susah, semua tergantung pada jenis perusahaan yang dimilikinya. Beberapa kegiatan yang dilakukan dalam sistem warehousing pada umumnya meliputi pencatatan seperti produk, harga produk yang akan dijual, biaya persediaan, harga pokok persediaan, catatan pengeluaran dan permintaan barang serta perhitungan fisik persediaan (stock opname)[?]. Pada proses aplikasi kita meliputi proses pencatatan seperti produk, harga produk yang akan dijual, catatan pengeluaran dan permintaan barang serta perhitungan fisik persediaan (stock opname) dan tabel data untuk barang masuk dan terjual terbanyak setiap bulannya. Dalam dewasa ini semua perusahaan pasti memiliki persediaan. Oleh karena itu untuk menjamin lancar nya arus lintas e-commerce maka perlu diadakan pencatatan terhadap pengelolaan barang, baik barang keluar

dan barang masuk serta untuk menentukan barang mana yang harus di prioritaskan oleh pelaku e-commerce sehingga pelaku ecommerce tepat dalam mengambil keputusan.. Pengelolan warehosuing yang baik dan pengambilan keputusan yang tepat akan memberikan dampak yang besar bagi perkembangan bisnis perusahaan, dengan warehousing yang terkelola dengan baik maka sistem penjualan juga dapat berjalan dengan baik pula dan sebaliknya jika warehousing yang terkelola tidak berjalan dengan baik maka akan menghasilkan sesuatu yang kurang baik. Sehingga dari uraian permasalahan diatas akan dibuat suatu sistem untuk pengelolaan barang masuk dan barang keluar untuk mengetahui informasi stock yang ada guna memudahkan pemilik usaha dalam penjualan .

5.2 Metode Pengumpulan Data

1. Pre-Processing

Tahap pre-processing data merupakan proses untuk mempersiapkan data mentah sebelum dilakukan proses lain. Pada umumnya, pre-processing data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah di proses oleh system. Text Preprocessing menjadi tahap awal dalam text minning. Preprocessing dilakukan untuk menghilangkan bagian atau teks yang tidak diperlukan sehingga mendapatkan data yang berkualitas untuk dieksekusi.

Tahapan PreProcessing :

(a) Tokenizing

Tokenizing adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan.

(b) Filtering

Tahap Filtering adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma stoplist(membuang kata kurang penting atau wordlist(menyimpan kata penting). Stoplist/stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words.

(c) Stemming

Teknik stemming diperlukan selain untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda.

2. Analisis Data

Analisis Data adalah rangkaian kegiatan penelaahan, pengelompokan, sistematiasi, penafsiran dan verifikasi data agar sebuah fenomena memiliki nilai social, akademis, dan ilmiah.

Kegiatan analisis data adalah mengelompokkan data berdasarkan variabel dan jenis responden, mentabulasi data berdasarkan variabel dan seluruh responded,

menyajikan data tiap variabel yang diteliti, melakukan perhitungan untuk men-guji hipotesis.

5.3 Contoh studi kasus penggunaan aplikasi ini



Gambar 5.1 Usaha sembako kelas menengah-bawah

suatu pengusaha ingin memulai wirausaha jual beli barang secara online di market online. Untuk memanajemen usahanya pengusaha masimenggunakan cara maual yaitu dengan menulis di buku catatan baik itu barang yang terjual, barang masuk dan stock. seiring berkembangnya usaha jumlah barang yang semakin meningkat dan sulit untuk di manajemen dan karena khawatir data ilang atau rusak maka pengusaha memutuskan untuk beralih cara manajemen dari tradisional ke modern.

Smart Gudang merupakan sebuah aplikasi yang dirancang khusus untuk manajemen barang dimana cara kerja dari aplikasi ini dengan menunjukan scan barcode ke arah barang. Aplikasi ini ditujukkan bagi masyarakat yang memiliki usaha kecil yang terbatas biaya karena dilihat dari permasalahan yang ada dalam lingkup pergudangan bawah menengah masih banyak sekali kesulitan dalam hal pencatatan data sehingga tidak mengefisiensikan waktu. Dan dengan adanya aplikasi ini masyarakat yang memiliki usaha kecil tidak perlu merogoh banyak biaya unutk membeli alat scanner, cukup dengan menggunakan smartphone yang telah terinstall aplikasi scanner ini.

Tujuan :

1. Mengefisiensikan waktu dengan mengubah teknologi tradisional menjadi teknologi modern

2. Mengganti device khusus scanner dengan android yang telah terinstall aplikasi smart Gudang
3. Mempermudah manajemen barang masuk dan barang keluar

Manfaat :

1. Mengefisiensi waktu dalam memanajemen barang masuk dan barang keluar
2. Mengurangi biaya dalam penuhan alat device (alat scanner)
3. Masyarakat di permudah dalam memanajemen barang

5.4 Jurnal Yang Terkait Dengan Penilitian

1. Nama Peneliti : I Putu Alit Putra Yudha, Made Sudarma, Putu Arya Mertasana
Judul Peneliti : Perancangan Aplikasi Sistem Inventory Barang menggunakan barcode Scanner berbasis Android

Tujuan Penelitian : k merancang aplikasi sistem inventory barang yang menggunakan barcode scanner di perangkat Android.

Hasil Penelitian : aplikasi ini dapat melakukan pencarian data stok barang di seluruh outlet Krisna Oleh-Oleh Bali karena memiliki database yang dapat diakses oleh semua user yang sudah diregistrasi oleh Admin. Aplikasi ini juga dapat digunakan pada smartphone Android, sehingga dalam pemakaiannya dapat mempermudah user dalam melakukan pengecekan stok barang tanpa harus melalui Operator toko ataupun mengecek secara manual [?].

2. Nama Peneliti : Dini Hamidin Sntoso Paujiah Mutianingsih.

Judul Penelitian : .Rancang Bangun Aplikasi Warehouse Berbasis Web Terintegrasi Dengan QrCode.

Tujuan Penelitian : Aplikasi dapat melakukan pengelolaan terhadap data barang, pengelolaan data barang bertujuan agar pengelola dapat melihat informasi aliran barang masuk dan keluar di warehouse; Aplikasi dapat mengelola data barang masuk. Proses kelola barang dimulai dari mencatat barang yang dikirim supplier kemudian dilanjutkan dengan proses penginputan informasi barang, poses penyimpanan barang, dan mendata barang yang akan keluar dari warehouse; Aplikasi dapat menampilkan data barang keluar dengan informasi barang; Aplikasi dapat menampilkan posisi barang pada rak yang tersedia di warehouse; Aplikasi dapat melihat status rak penuh dan tersedia, dan menampilkan kapasitas rak yang masih kosong

3. Nama Peneliti : Agustian Noor, Herpendi, Radna Nurmaliana.

Judul Penelitian : Perencanaan dan Pengembangan Aplikasi Stok Barang dan Penjualan pada UPT.Kewirausahaan Menggunakan Barcode dan Smart Card.

Tujuan Penelitian : Unit Pelaksana Teknis (UPT) Kewirausahaan Politeknik Negeri Tanah Laut adalah sebuah unit yang bergerak di bidang wirusaha yang

bertujuan mewadahi minat dan bakat mahasiswa dalam bidang entrepreneurship. Barcode Scanner dan RFID akan mempermudah setiap transaksi yang dilakukan yaitu hanya dengan menscan kode barang dan kartu RFID untuk identitas pembeli maka data akan tampil dilayar komputer.

Hasil Penelitian : Aplikasi Stok Barang dan Penjualan berhasil dibangun untuk menunjang kinerja UPT. Kewirausahaan. Dengan aplikasi ini proses transaksi penjualan dilakukan secara terkomputerisasi hanya dengan memindai barang yang akan dijual. Stok barang secara otomatis mengalami pengurangan setiap terjadinya transaksi penjualan sehingga mempermudah dalam pengelolaannya. Data pelanggan disimpan dalam Smart Card untuk mempermudah proses pencatatan yaitu hanya dengan memindai Smart Card maka data akan tampil di aplikasi. Dengan aplikasi ini pengelolaan data lebih teratur sehingga proses pelaporan juga akan menjadi lebih baik.

4. Nama Peneliti : Arie Setya Putra, Ochi Marshella Febriani.

Judul Penelitian : Sistem Informasi Monitoring Inventory Barang pada Balai Riset Standarisasi Industri Bandar Lampung.

Tujuan Penelitian : Balai Riset Standardisasi dan Industri Bandar Lampung perlu melakukan evaluasi data yang baik dengan sistem informasi monitoring inventori barang guna mengontrol peminjaman dan pengembalian barang melalui monitoring inventori.

Hasil Penelitian : Jika sewaktu-waktu diperlukan informasi tentang monitoring inventori barang maka dapat menyajikan secara cepat dan tepat. Berbeda dengan penggerjaan sebelumnya yaitu proses penggerjaanya masih dilakukan dengan cara pencatatan sehingga data yang dihasilkan memakan waktu yang lama juga tingkat keakuratan data kecil dengan program aplikasi ini proses penggerjaan cepat, data yang dihasilkan memiliki tingkat keakuratan data yang tinggi.

5. Nama Peneliti : Putu Alit Putra Yudha, Made Sudarma, Putu Arya Martasana.

Judul Penelitian : Perancangan Aplikasi Sistem Inventory Barang Menggunakan Barcode Scanner Berbasis Android.

Tujuan Penelitian : Penelitian ini bertujuan untuk merancang aplikasi sistem inventory barang yang menggunakan barcode scanner di perangkat Android.

Aplikasi ini diharapkan nantinya dapat membantu pegawai Krisna Oleh-Oleh Khas Bali untuk meningkatkan pelayanan kepada pelanggan dalam hal penginformasian data stok inventory barang. Hasil Penelitian : Kesimpulan yang dapat diambil setelah melakukan perancangan dan pembuatan aplikasi sistem inventory barang di Krisna Oleh-Oleh berbasis Android yaitu, aplikasi ini dapat melakukan pencarian data stok barang di seluruh outlet Krisna Oleh-Oleh Bali karena memiliki database yang dapat diakses oleh semua user yang sudah diregistrasi oleh Admin. Aplikasi ini juga dapat digunakan pada smart-phone Android, sehingga dalam pemakaiannya dapat mempermudah user dalam melakukan pengecekan stok barang tanpa harus melalui Operator toko ataupun

mengecek secara manual.

6. Nama Peneliti : Michael Christian Wibisono, Agustinus Noertjahiyana, Andreas Handojo.

Judul Penelitian : Pembuatan Aplikasi Pencatatan Stock dengan Menggunakan Barcode Pada Android.

Tujuan Penelitian : Stock opname pada barang dilakukan untuk memeriksa perbedaan antara jumlah barang pada gudang dengan jumlah yang tercatat pada database.

Hasil Penelitian : Aplikasi dapat melakukan pencatatan jumlah stok barang dan aset pada periode tertentu.

5.5 Siapakah pengguna aplikasi smart gudang

Untuk pengguna aplikasi android smart gudang ini khususnya lebih ditujukan kepada usaha kelas menengah-bawah. Karena apa? karena ,Aplikasi ini mungkin masih tergolong aplikasi yang sederhana sehingga untuk orang awam pun masih sanggup untuk menggunakan aplikasi ini tanpa kebingungan atau kesulitan.Aplikasi ini tidak di rekomendasikan bagi usaha menengah-atas karena untuk usaha yang menengah-atas aplikasi ini bisa dikatakan simple atau mungkin tidak bisa menganalisis pendataan yang lebih spesifik lagi. Mengapa kita membuat aplikasi ini untuk usaha menengah-bawah?

1. Tidak menyulitkan pemilik usaha menengah-bawah karena dengan alat ini pengusaha hanya perlu menunjukkan barcode ke hp lalu data akan tersimpan, tidak lagi pengusaha harus mencatat pada kertas.
2. Tempat penyimpanan data lebih aman.
3. Lebih simple dan mudah dibawa kemana-mana.

Contoh di bawah ini merupakan usaha kelas menengah-bawah



Gambar 5.2 Contoh usaha kelas menengah-bawah

5.6 Tentang kelas Menengah-bawah

5.6.1 Kelas Menengah

Kelas menengah bisa dikatakan mendominasi masyarakat Indonesia saat ini. Masyarakat kelas menengah memiliki pengaruh yang besar dalam pertumbuhan perekonomian di Indonesia. Bisa dibilang, kehidupan kelas menengah cenderung stabil dan teratur. Mereka bisa memenuhi kebutuhan dan pengeluaran sehari-hari. Meningkatnya jumlah kelas menengah di Indonesia turut mempengaruhi penurunan angka kemiskinan. Pertumbuhan kelas menengah sendiri dapat diukur dengan kenaikan PDB (Pendapatan Domestik Bruto).

Kelas menengah di Indonesia masih dibagi ke dalam beberapa golongan berdasarkan jumlah pengeluaran mereka, yaitu:

1. Poor middle dengan pengeluaran bulanan di bawah Rp 1.000.000.
2. Aspirant middle dengan pengeluaran bulanan antara Rp 1.000.000 sampai Rp 1.500.000.
3. Emerging middle dengan pengeluaran bulanan antara Rp 1.500.000 sampai Rp 2.000.000.
4. Middle dengan pengeluaran bulanan antara Rp 2.000.000 sampai Rp 3.000.000.
5. Upper middle dengan pengeluaran bulanan antara Rp 3.000.000 sampai Rp 5.000.000.

6. Affluent dengan pengeluaran bulanan antara Rp 5.000.000 sampai Rp 7.500.000.
7. Elite dengan pengeluaran bulanan lebih dari Rp 7.500.000.

5.6.2 Kelas Bawah

Sebagai negara berkembang, jumlah masyarakat kelas bawah di Indonesia masih dominan. Dilihat dari sisi pendapatan, golongan masyarakat ini memiliki angka penghasilan di bawah rata-rata. Itulah mengapa masyarakat kelas bawah biasanya hidup di bawah garis kemiskinan. Di Indonesia, kehidupan masyarakat kelas bawah disubsidi oleh pemerintah dalam berbagai aspek.

Mulai dari program pendidikan, kesehatan, dan kesejahteraan, masyarakat kelas bawah mendapatkan perhatian khusus dari pemerintah. Jumlah masyarakat kelas bawah mempengaruhi angka kemiskinan di Indonesia.

Terjadi kesenjangan sosial yang tajam antara kelas bawah dengan kelas lain di masyarakat. Kelas bawah mengalami kesulitan untuk mengakses berbagai layanan publik, sumber daya, dan lapangan pekerjaan. Masyarakat kelas bawah tersebar di berbagai wilayah, mulai dari perkotaan, pedesaan, hingga masyarakat yang hidup jauh dari pusat pembangunan.

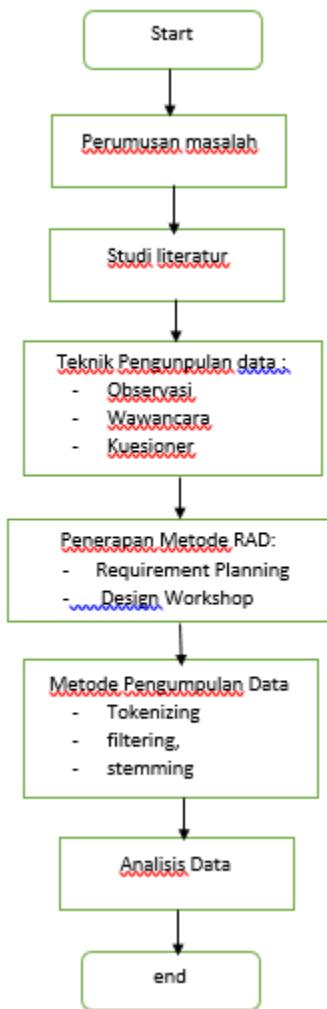
BAB 6

METODOLOGI PENELITIAN

6.1 Metodologi Penelitian

Metodologi Penelitian adalah sekumpulan peraturan, kegiatan, dan prosedur yang digunakan oleh pelaku suatu disiplin. Metodologi juga merupakan analisis teoritis mengenai suatu cara atau metode. Penelitian merupakan suatu penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban. Keinginan untuk memperoleh dan mengembangkan pengetahuan merupakan kebutuhan dasar manusia yang umumnya menjadi motivasi untuk melakukan penelitian.

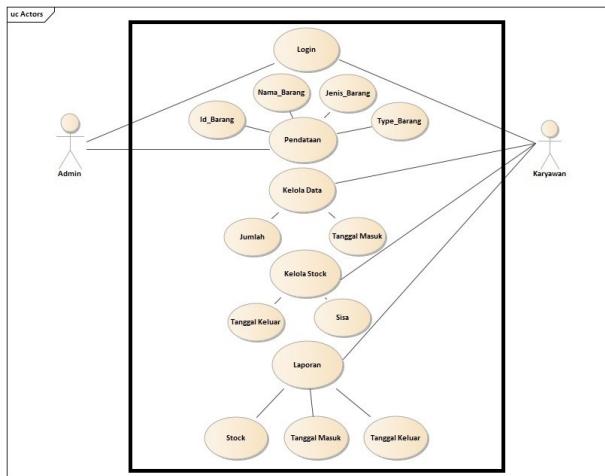
Berikut ini adalah alur dari metodologi penelitian yang dilakukan di dalam penelitian ini ;



Gambar 6.1 Alur metodologi penelitian

6.2 Analisis dan Perancangan

6.2.0.1 Use Case Diagram Berikut gambar dari Use Case Diagram

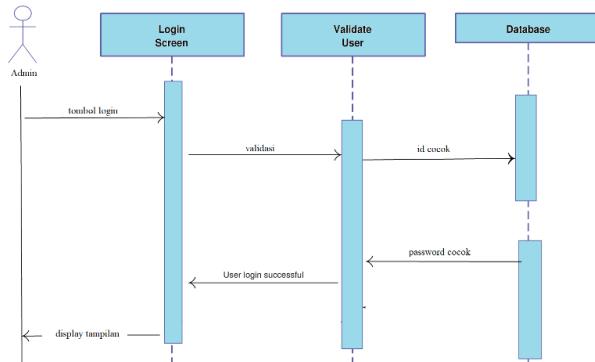


Gambar 6.2 Relationship use case kelompok dan actor

6.2.1 Sequence Diagram

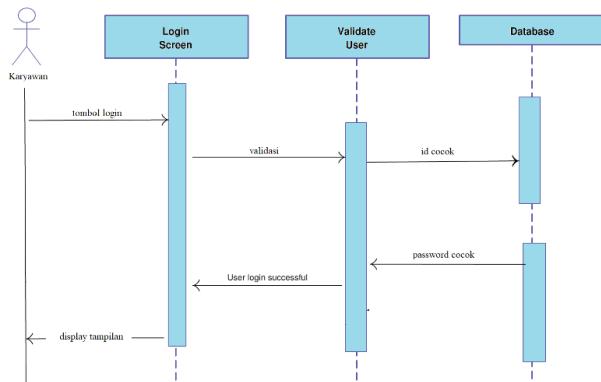
Sequence diagram disini adalah untuk menggambarkan kolaborasi dinamis antara sejumlah object, yang termasuk ke dalam sistem yang akan dibangun

1. Sequence Diagram Login Admin



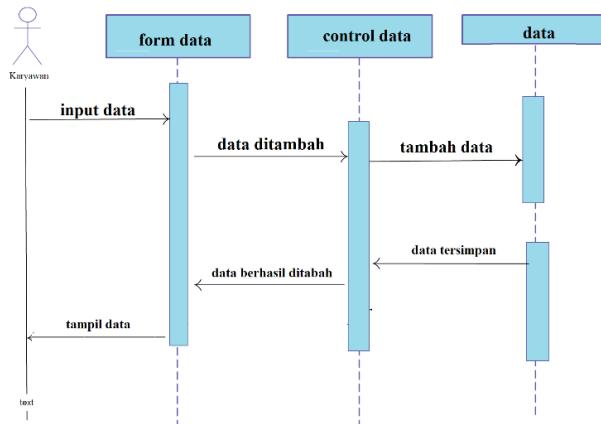
Gambar 6.3 Sequence Diagram Login Admin

2. Sequence Diagram Login Karyawan



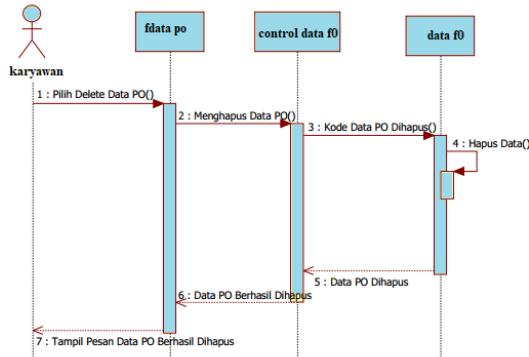
Gambar 6.4 Sequence Diagram Login Karyawan

3. Sequence Diagram Pendataan



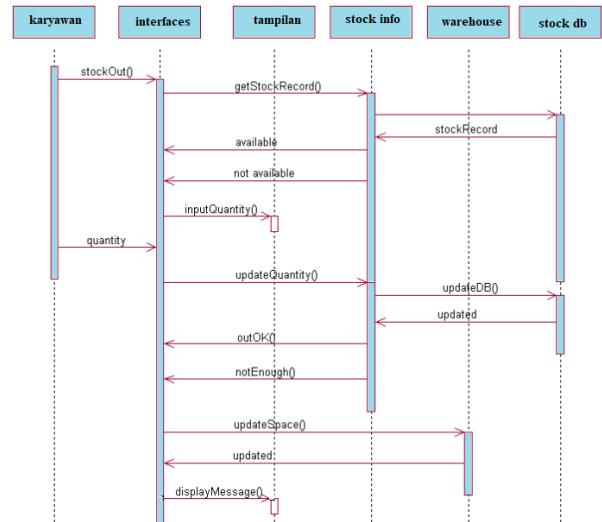
Gambar 6.5 Sequence Diagram Pendataan

4. Sequence Diagram Kelola Data



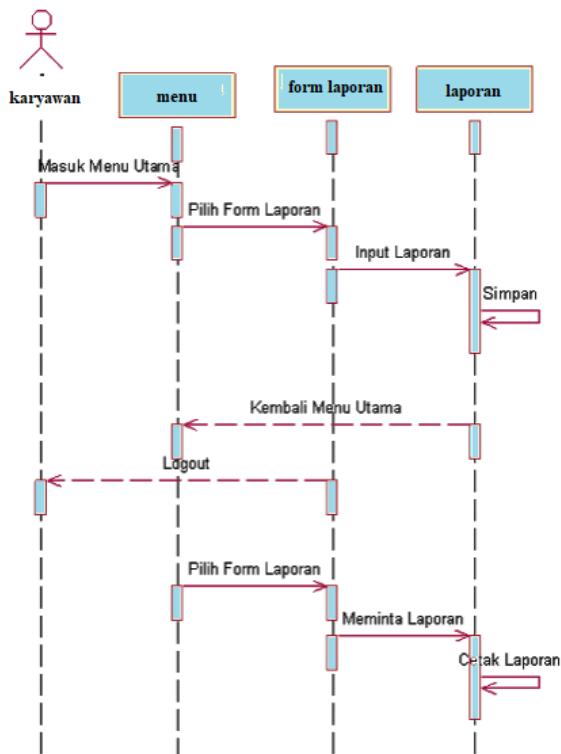
Gambar 6.6 Sequence Diagram Kelola Data

5. Sequence Diagram Kelola Stock



Gambar 6.7 Sequence Diagram Kelola Stock

6. Sequence Diagram Laporan

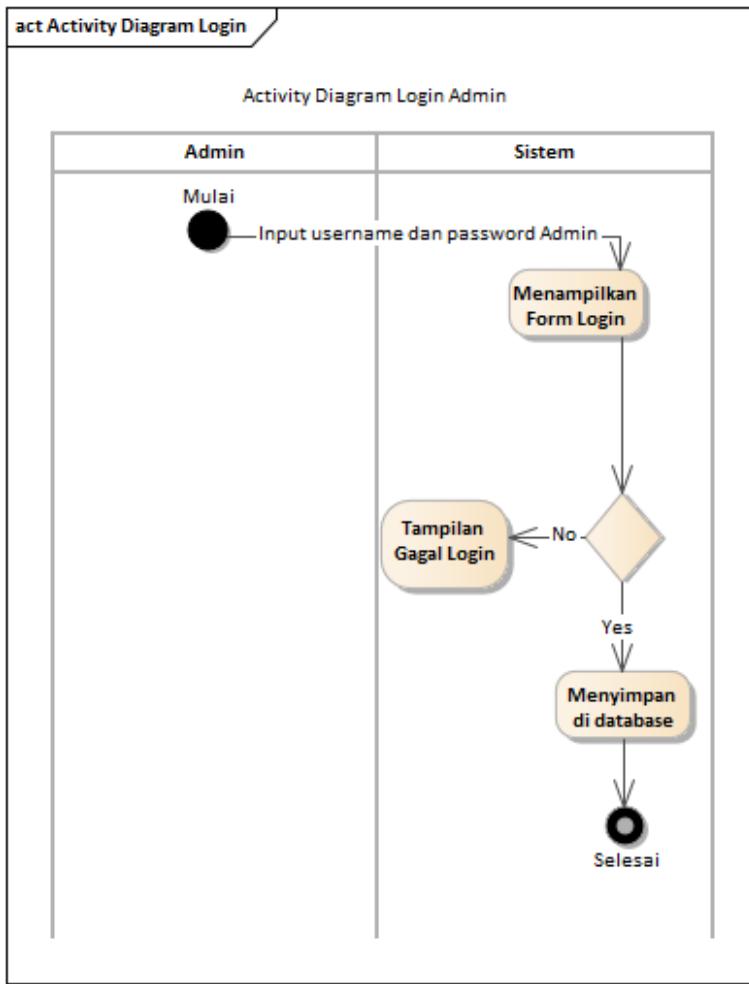


Gambar 6.8 Sequence Diagram Laporan

6.2.2 Activity Diagram

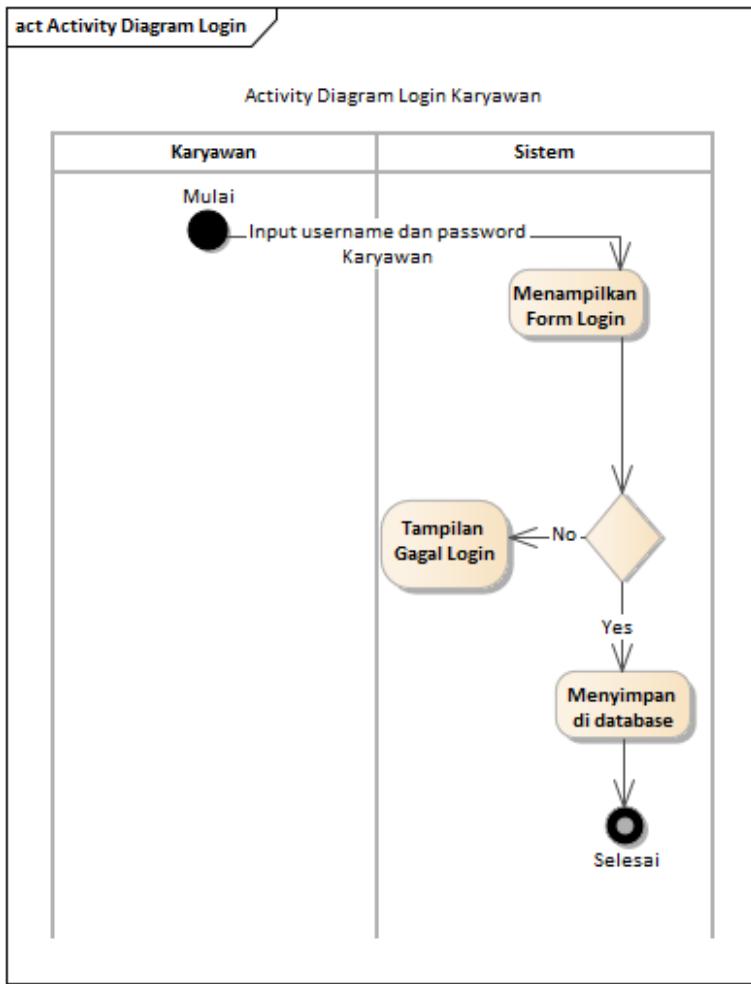
Activity diagram disini berfungsi untuk menggambarkan rangkaian aliran aktivitas yang ada di dalam sistem yang akan dibangun .

1. Activity Diagram Login Admin



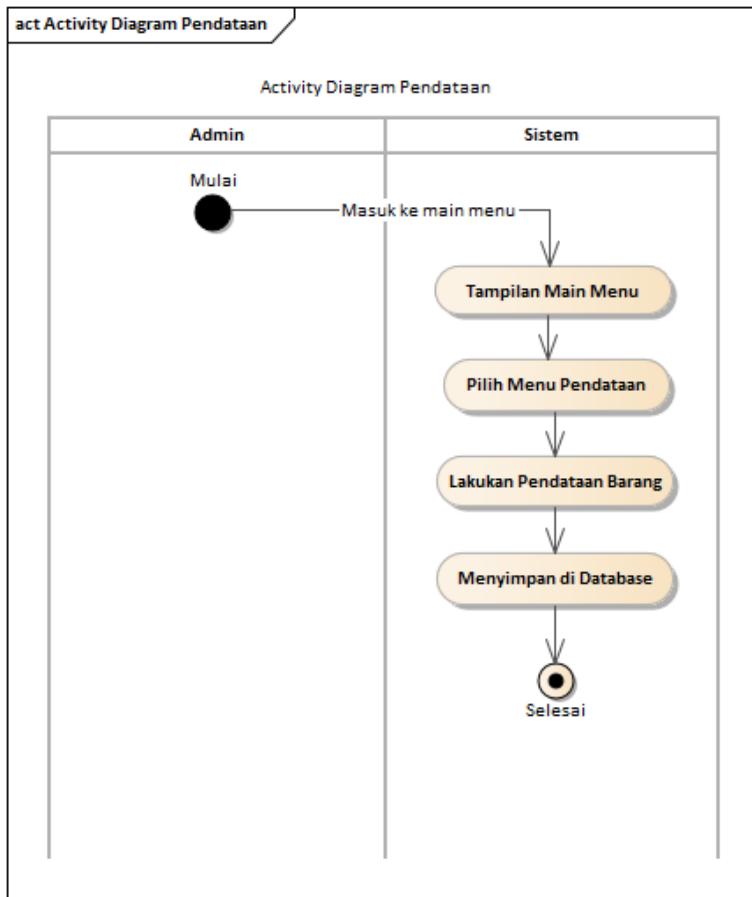
Gambar 6.9 Activity Diagram Login Admin

2. Activity Diagram Login Karyawan



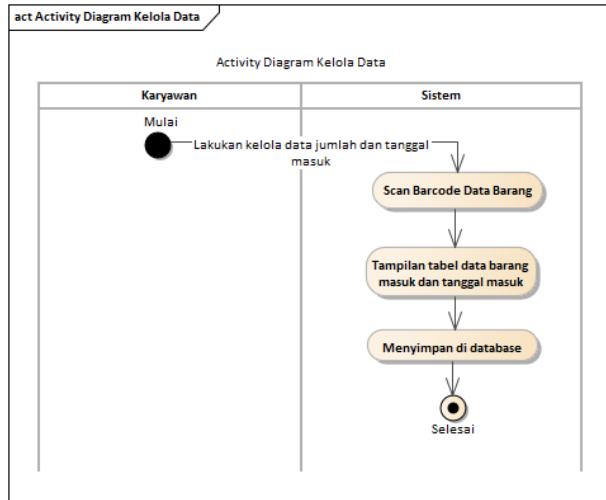
Gambar 6.10 Activity Diagram Login Karyawan

3. Activity Diagram Pendataan



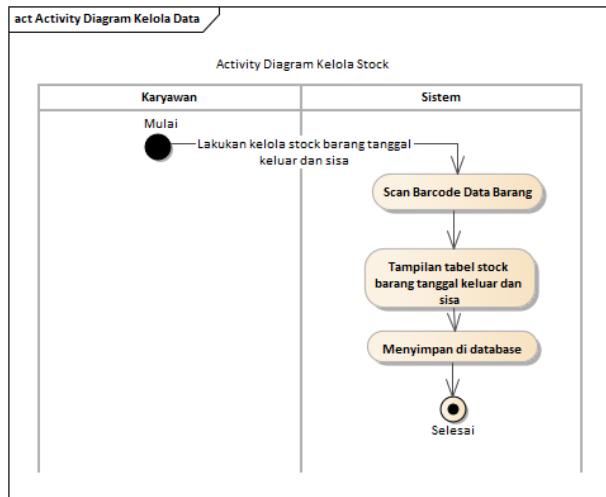
Gambar 6.11 Activity Diagram Pendataan

4. Activity Diagram Kelola Data



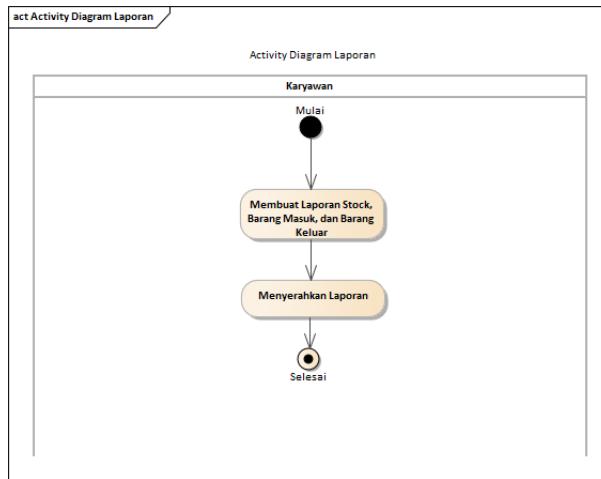
Gambar 6.12 Activity Diagram Kelola Data

5. Activity Diagram Kelola Stock



Gambar 6.13 Activity Diagram Kelola Stock

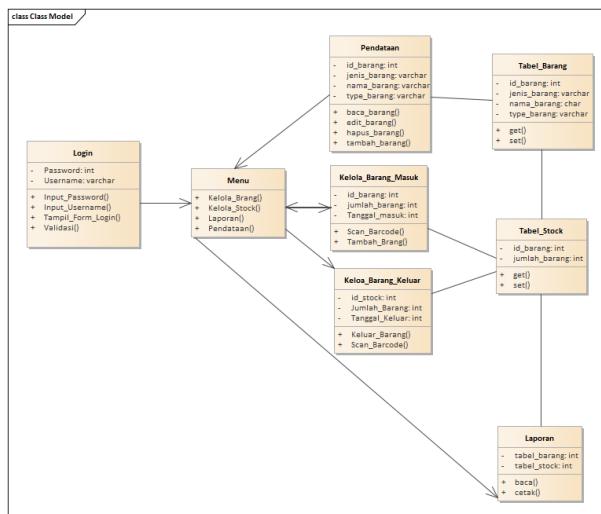
6. Activity Diagram Laporan



Gambar 6.14 Activity Diagram Laporan

6.2.3 Class Diagram

Class Diagram dibawah ini adalama menunjukkan semua entitas sistem yang akan dibangun, dan juga general relationship-nya.

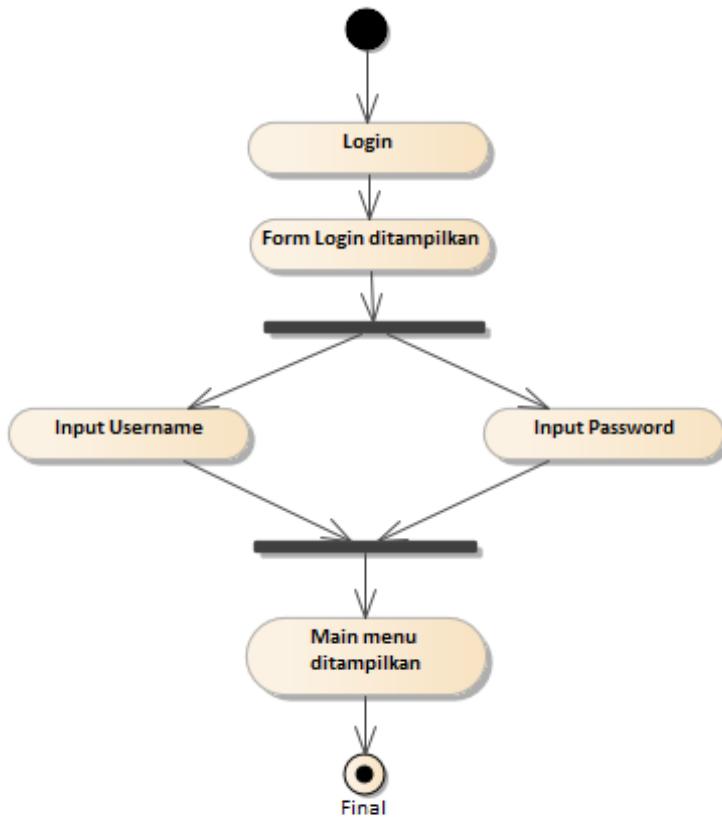


Gambar 6.15 Class diagram

6.2.4 Statechart Diagram

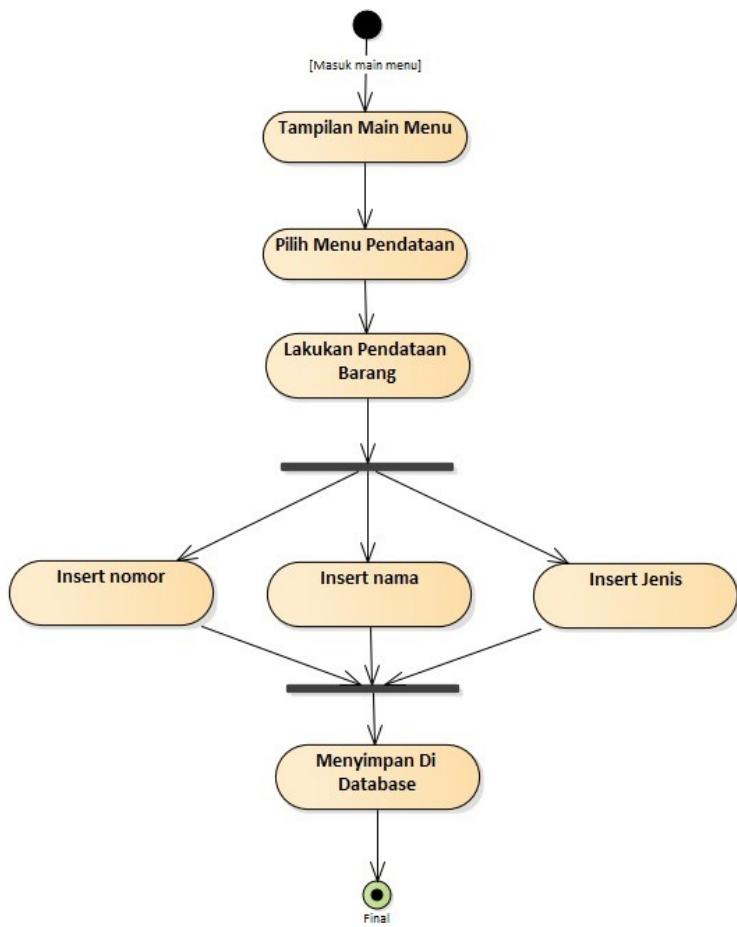
Statechart diagram disini adalah untuk menampilkan state machine, yang terdiri dari states, transitions, events, dan activities.

1. Statechart Diagram Login



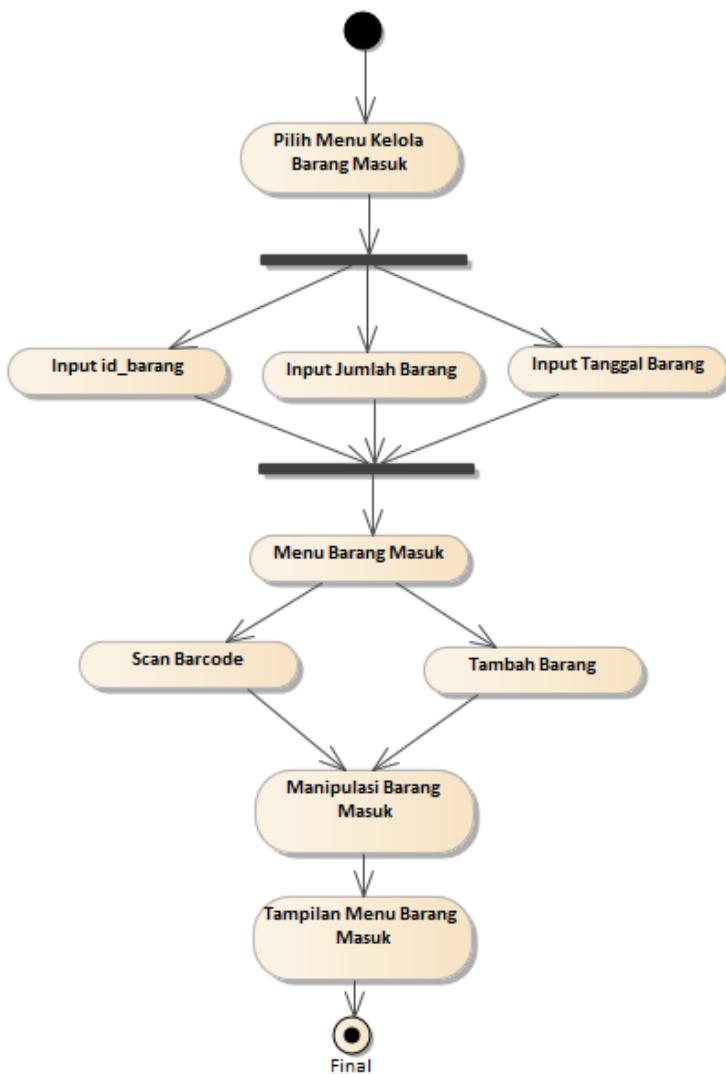
Gambar 6.16 Statechart Diagram Login

2. Statechart Diagram Pendataan



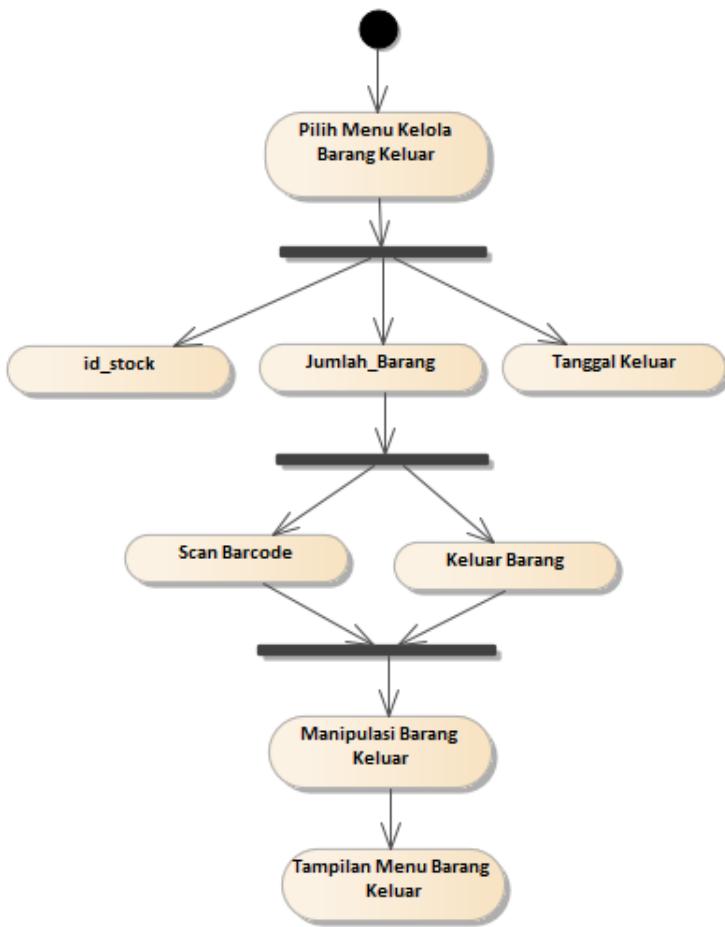
Gambar 6.17 Statechart Diagram Pendataan

3. Statechart Diagram Kelola Barang Masuk



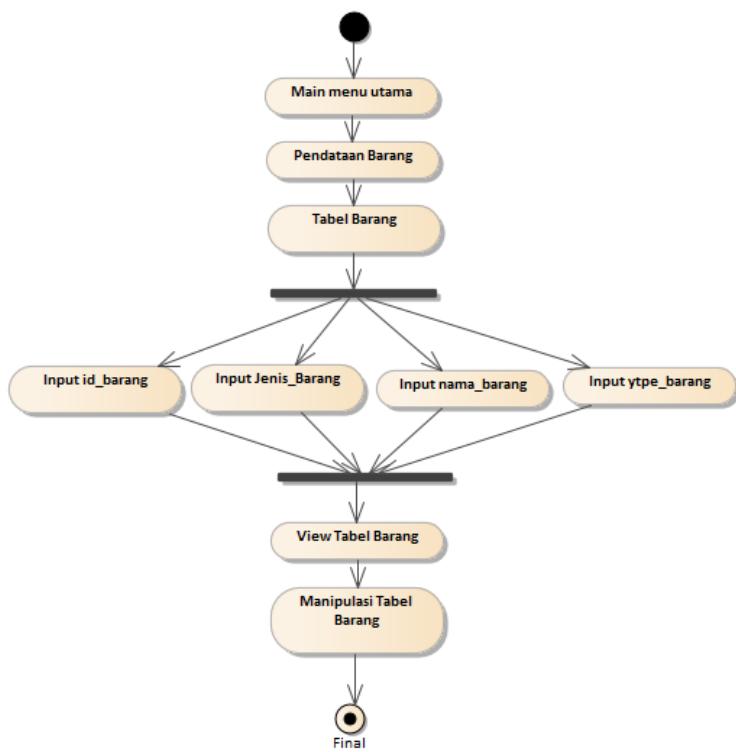
Gambar 6.18 Statechart Diagram Kelola Barang Masuk

4. Statechart Diagram Kelola Barang Keluar



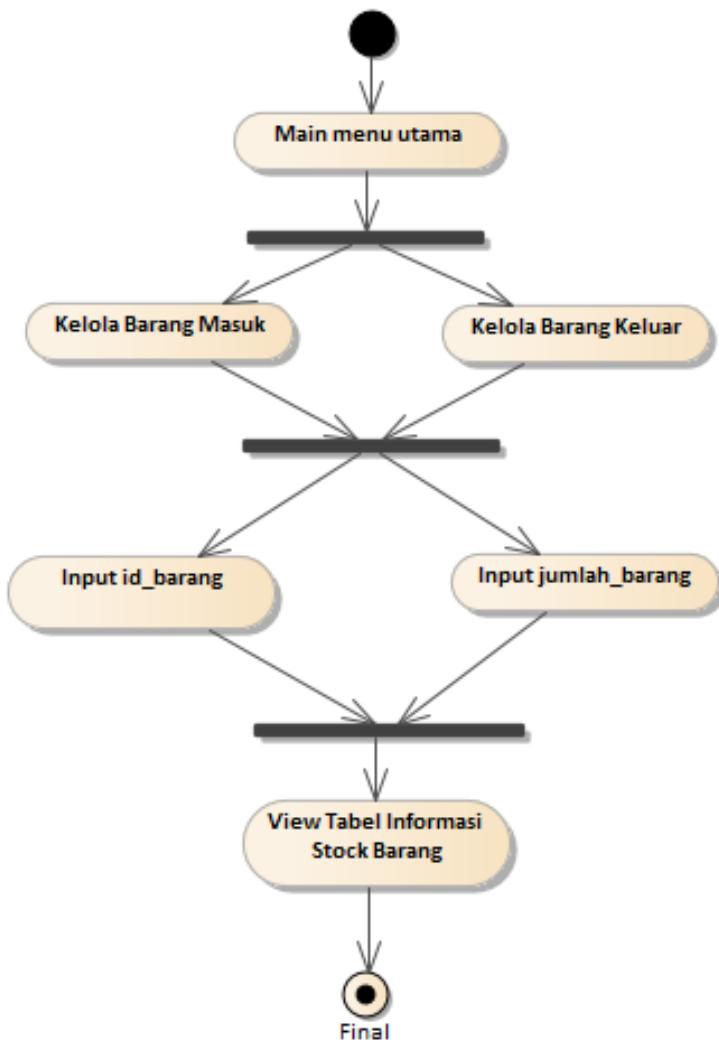
Gambar 6.19 Statechart Diagram Kelola Barang Keluar

5. Statechart Diagram Kelola Tabel Barang

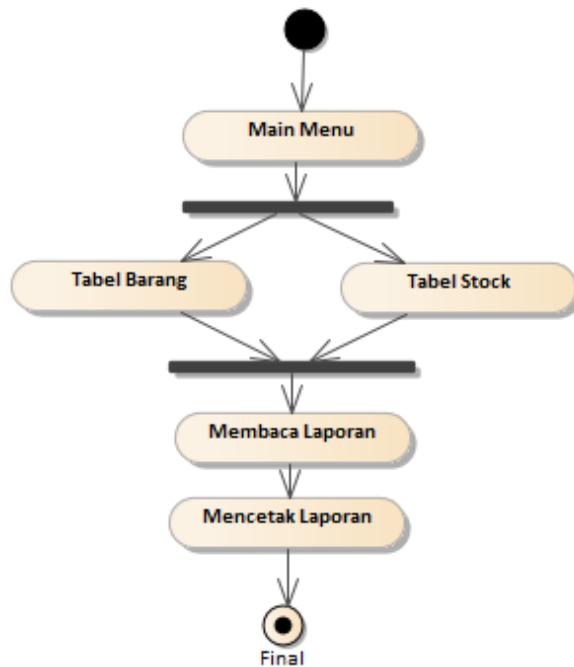


Gambar 6.20 Statechart Diagram Kelola Tabel Barang

6. Statechart Diagram Kelola Tabel Stock



Gambar 6.21 Statechart Diagram Kelola Tabel Stock



Gambar 6.22 Statechart Diagram Laporan

6.3 Implementasi

Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberikan masukan kepada pembangun sistem.

6.3.1 Lingkungan Tujuan

Aplikasi ini akan berjalan dengan baik karena didukung oleh perangkat lunak dan perangkat keras yang memadai.

1. Kebutuhan Perangkat Keras

No	Nama Perangkat	Spesifikasi	Keterangan
1.	Hard Disk	500 GB	Media untuk menyimpan data aplikasi
2.	Memory	8 GB	Memory Sistem yang digunakan
3.	Processor	<i>Intel® core™ 2 duo CPU T6570 @2,10 Ghz</i>	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan processor.
4.	Infrastruktur Jaringan		Server Dan Host

Gambar 6.23 Tabel Perangkat Keras

2. Kebutuhan Perangkat Lunak

No	Tools / Software	Fungsi	Keterangan
1.	<i>Windows 10</i>	Sistem Operasi	Windows 10 unggul dibanding versi sebelumnya
2.	<i>Xampp 1.7.3</i>	Server Basis Data	XAMPP adalah software perangkat lunak yang memiliki kemampuan atau dukungan sistem operasi yang banyak dalam perancangan web.
3.	<i>PHP, Java</i>	<i>Bahasa Pemograman Yang digunakan</i>	Bahasanya mudah dipahami dan mengatur desain seefisien mungkin.
4.	<i>Sublime text, Android Studio</i>	<i>Pengkodingan</i>	Aplikasi text editor untuk menulis kode.
5.	<i>Google Crome</i>	<i>Browser</i>	Tidak berat atau tidak memakan banyak penggunaan RAM.

Gambar 6.24 Tabel Perangkat Lunak

6.4 Pengujian dan Hasil Pengujian

Pengujian merupakan hal yang penting untuk dilakukan karena memiliki tujuan untuk menemukan sekurangan yang ada pada sistem yang dibuat. Pengujian memiliki maksud untuk mengetahui apakah sistem/ aplikasi yang dibuat telah sesuai dengan kriteria yang diinginkan.

6.4.1 Identifikasi dan Rencana Pengujian

Pengujian yang dilakukan yaitu dengan pengujian blackbox yang digunakan untuk menguji fungsi-fungsi dari perangkat lunak yang telah dirancang. Berikut merupakan tabel dari identifikasi dan rencana pengujian aplikasi yang telah dibuat:

No	Kelas Uji	Butir Uji	Jenis Pengujian
1.	Barang Masuk	Validasi barang masuk	<i>Black Box</i>
2.	Barang Keluar	Validasi barang keluar	<i>Black Box</i>

Gambar 6.25 Identifikasi dan rencana pengujian

6.4.2 Deskripsi dan Hasil Uji

Hasil uji ini merupakan proses yang telah berhasil pada sistem yang telah dibuat. Hasil uji yang diharapkan dan kesimpulan pada saat pengujian adalah sebagai berikut:

6.4.2.1 Pengujian Halaman Barang Masuk Berikut hasil pengujian dari halaman barang masuk:

Kelas Uji	Butir Uji	Hasil	Keterangan
Halaman Barang Masuk	Validasi Barang Masuk	Menampilkan Halaman tabel barang masuk	[√] Berhasil

Gambar 6.26 Pengujian halaman barang masuk

6.4.2.2 Pengujian Halaman Barang Keluar Berikut hasil pengujian dari halaman barang keluar:

Kelas Uji	Butir Uji	Hasil	Keterangan
Halaman Barang Keluar	Validasi Barang Keluar	Menampilkan Halaman barang keluar	[√] Berhasil

Gambar 6.27 Pengujian halaman barang keluar

BAB 7

BAGIAN-BAGIAN ANDROID STUDIO

7.1 Bagian-bagian pada Android Studio

7.1.1 Activity Main.xml

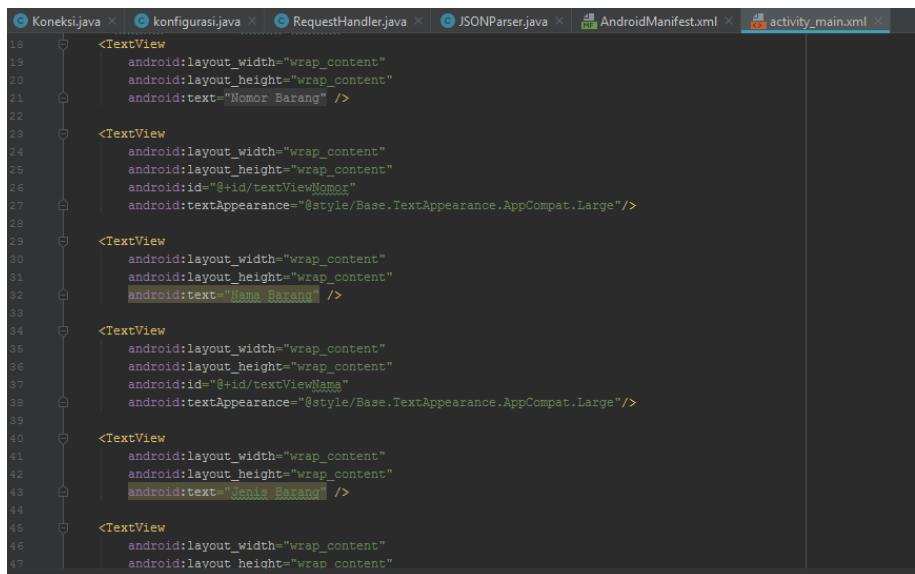
Jadi Activity_Main bisa di definisikan Aktivitasnya adaalah java dan tata letak nya adalah file XML. Jadi, nama Activity_Main akan membuat kelas Java, nah kelas java biasa disebut Main_Activity.java. dan file XML disebut dengan activity-main.xml yang mana nanti ketika sudah menyelesaikan source code nya lalu di jalankan maka Android Studio akan membangun Aplikasi yang dibuat.

1. Membuat posisi tampilan menjadi Relative Layout

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project2.MainActivity">
```

Gambar 7.1 Untuk membuat posisi tampilan

- Setelah membuat Layout lanjut untuk membuat TextView nya. Text View disini digunakan untuk menambahkan tulisan pada kotak.



```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project2.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nomor Barang" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textViewNomor"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Barang" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textViewNama"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Large"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Jenis Barang" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Barcode" />

```

Gambar 7.2 Membuat icon dan menambahkan text view

- Membuat button pada icon barang masuk, barang keluar dan infomasi stok barang.

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Barang Masuk"  
    android:id="@+id/buttonAdd" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Barang Keluar"  
    android:id="@+id/buttonOut" />  
  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Informasi Stock Barang"  
    android:id="@+id/buttonTabel" />  
</LinearLayout>  
  
<Button  
    android:id="@+id/buttonScan"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:text="Scan QR Code" />  
  
</RelativeLayout>
```

Gambar 7.3 Membuat Button

7.1.2 Main Activity.java

Class ini meng extens dari class activity.xml. Main Activity.java ini digunakan untuk memanggil fungsi. Fungsi ini dubuat ketika class sudah dibuat.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    private Button buttonScan;  
    private TextView textViewNama, textViewNomor, textViewJenis;  
    private Button buttonAdd;  
    private Button buttonOut;  
    private Button buttonTabel;  
  
    //qr code scanner object  
    private IntentIntegrator intentIntegrator;
```

Gambar 7.4 Main Activity meng extends dari ActivityMain.xml

1. Selanjutnya setelah kita membuat text view dan button pada activity main maka teman teman akan mengintialize object.

```

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

    // initialize object
    buttonScan = (Button) findViewById(R.id.buttonScan);
    textViewNomor = (TextView) findViewById(R.id.textViewNomor);
    textViewNama = (TextView) findViewById(R.id.textViewNama);
    textViewJenis = (TextView) findViewById(R.id.textViewJenis);
    buttonAdd = (Button) findViewById(R.id.buttonAdd);
    buttonOut = (Button) findViewById(R.id.buttonOut);
    buttonTabel = (Button) findViewById(R.id.buttonTabel);
}

```

Gambar 7.5 Initialize object

2. selanjutnya setelah teman-teman membuat button dan text view sebelumnya, maka teman teman membutuhkan perintah berikut agar button dan text view bisa dijalankan.

```

//Setting listeners to button
buttonAdd.setOnClickListener(this);
buttonOut.setOnClickListener(this);
buttonTabel.setOnClickListener(this);

// attaching onclickListener
buttonScan.setOnClickListener(this);
}

//Dibawah ini merupakan perintah untuk Menambahkan Pegawai (CREATE)
public void reset(){
    textViewNomor.setText("");
    textViewNama.setText("");
    textViewJenis.setText("");
}

```

Gambar 7.6 Setting button dan text view

3. Selanjutnya agar data bisa di eksekusi maka teman teman membutuhkan perintah sebagai berikut:

```

private void additem(){
    final String nomor = textViewNomor.getText().toString();
    final String nama = textViewNama.getText().toString();
    final String jenis = textViewJenis.getText().toString();

    class Additem extends AsyncTask{
        ProgressDialog loading;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(context MainActivity.this, null, "Menambahkan...", message: "Tunggu...", indeterminate: false, cancelable: false);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            Toast.makeText(context MainActivity.this, s, Toast.LENGTH_LONG).show();
            react();
        }
    }
}

```

Gambar 7.7 Untuk eksekusi barang masuk

4. lanjutan source code diatas

```

    @Override
    protected String doInBackground(Void... v) {
        HashMap<String, String> params = new HashMap<>();
        params.put(konfigurasi.KEY_EMP_NOMOR,nomor);
        params.put(konfigurasi.KEY_EMP_NAMA,nama);
        params.put(konfigurasi.KEY_EMP_JENIS,jenis);

        RequestHandler rh = new RequestHandler();
        String res = rh.sendPostRequest(konfigurasi.URL_ADD, params);
        return res;
    }
}

AddItem ae = new AddItem();
ae.execute();
}
}

```

Gambar 7.8 Untuk eksekusi barang masuk

5. selanjutnya untuk eksesksi barang keluar hampir sama dengan eksekusi barang masuk
6. gunakan source code berikut

```

private void addOut(){
    final String nomor = textViewNomor.getText().toString().trim();
    final String nama = textViewNama.getText().toString().trim();
    final String jenis = textViewJenis.getText().toString().trim();

    class AddOut extends AsyncTask<Void,Void,String>{
        ProgressDialog loading;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(MainActivity.this, "Menunggu...", "Tunggu...", false, true);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            Toast.makeText(MainActivity.this,s,Toast.LENGTH_LONG).show();
            reset();
        }
    }
}

```

Gambar 7.9 Untuk eksekusi barang keluar

7. lanjutan untuk source code diatas

```

    @Override
    protected String doInBackground(Void... v) {
        HashMap<String, String> params = new HashMap<>();
        params.put(konfigurasi.KEY_EMP_NOMOR,nomor);
        params.put(konfigurasi.KEY_EMP_NAMA,nama);
        params.put(konfigurasi.KEY_EMP_JENIS,jenis);

        RequestHandler rh = new RequestHandler();
        String res = rh.sendPostRequest(konfigurasi.URL_DEL, params);
        return res;
    }
}

AddOut ae = new AddOut();
ae.execute();
}
}

```

Gambar 7.10 Untuk eksekusi barang keluar

8. Untuk bisa mendapatkan scan dan mendapatkan hasil scan QR Code bisa meggunakan perintah berikut ini.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if (result != null) {
        if (result.getContents() == null) {
            Toast.makeText(context, this, text: "Hasil tidak ditemukan", Toast.LENGTH_SHORT).show();
        } else {
            // jika barcode berisi data
            try {
                // converting the data json
                JSONObject object = new JSONObject(result.getContents());
                // Atur nilai ke TextViews
                textViewNomor.setText(object.getString(name: "nomor"));
                textViewNama.setText(object.getString(name: "nama"));
                textViewJenis.setText(object.getString(name: "jenis"));
            } catch (JSONException e) {
                e.printStackTrace();
                // jika format encoded tidak sesuai maka hasil
                // ditampilkan ke toast
                Toast.makeText(context, this, result.getContents(), Toast.LENGTH_SHORT).show();
            }
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}

```

Gambar 7.11 Untuk Mendapatkan Hasil Scan

9. Agar source code diatas mampu dijalankan untuk di klik pada icon gunakan script dibawah ini.

```

@Override
public void onClick(View v) {
    // inisialisasi IntentIntegrator(scanQR)
    if (v == buttonScan) {
        intentIntegrator = new IntentIntegrator(activity, this);
        intentIntegrator.initiateScan();
    }

    if (v == buttonAdd) {
        addItem();
    }

    if (v == buttonOut) {
        addOut();
    }

    if (v == buttonTabel) {
        startActivity(new Intent(packageContext, Tampil.class));
    }
}

```

Gambar 7.12 Untuk button dapat di klik

10. selanjutnya untuk membuat activity_tampil.xml. Di activity ini digunakan untuk membuat tampilan webview.



```

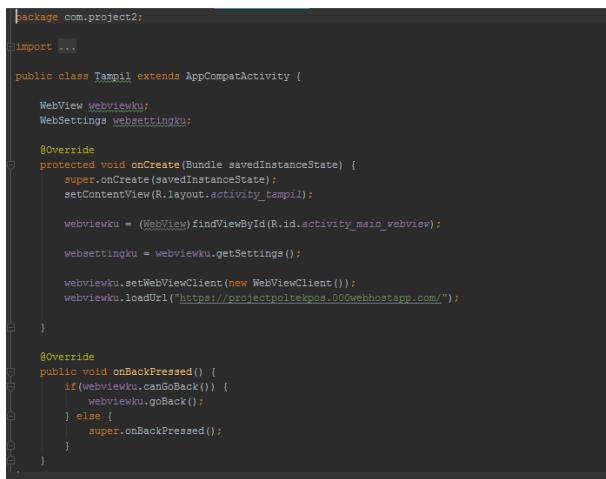
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.project2.Tampil">

    <WebView
        android:id="@+id/activity_main_webview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>

```

Gambar 7.13 Membuat tampilan web view

11. Selanjutnya untuk program tersebut dapat di eksekusi gunakan perintah berikut ini.



```

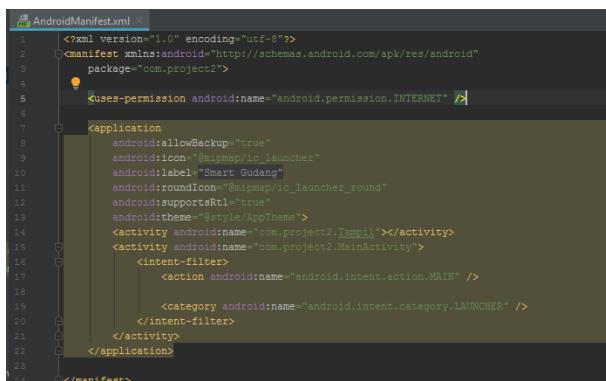
1 package com.project2;
2
3 import ...
4
5 public class Tampil extends AppCompatActivity {
6
7     WebView webviewku;
8     WebSettings websettingku;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_tampil);
14
15        webviewku = (WebView)findViewById(R.id.activity_main_webview);
16
17        websettingku = webviewku.getSettings();
18
19        webviewku.setWebClient(new WebviewClient());
20        webviewku.loadUrl("https://projectpoltekppos.000webhostapp.com/");
21    }
22
23    @Override
24    public void onBackPressed() {
25        if(webviewku.canGoBack()){
26            webviewku.goBack();
27        } else {
28            super.onBackPressed();
29        }
30    }
31
32}

```

Gambar 7.14 Meng-eksekusi tampilan web view

7.1.3 Android Mannifest.xml

Android Manifest.xml berisi informasi dari source code yang telah dibuat, yang bisa mencakup komponen aplikasi seperti kegiatan, layanan, penerima, dan penyedia konten. Karena setiap aplikasi pada android studio pasti memiliki manifest android. Karena jika android tidak memiliki file ini maka android tidak bisa di eksekusi.



Gambar 7.15 Manifest Android Studio

7.1.4 Membuat fitur scan barcode kamera menggunakan library ZXING

Ada beberapa hal yang harus dilakukan yaitu :

1. Menambahkan permission untuk menggunakan Camera di file AndroidManifest.xml

Untuk menambahkan permission Camera dapat menambahkan baris script tersebut diatas dengan tag `<application>` / `</application>` di file AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA"/>
```

Gambar 7.16 Gunakan script diatas

2. Menambahkan ZXING library

Tambahkan ZXING library di proyek anda untuk mengimplementasikan pemindai, dengan cara buka file build.gradle (Modul:app) dan tambahkan kode berikut pada bagian dependencies :

```
compile 'me.dm7.barcodescanner:zxing:1.9'
```

Gambar 7.17 Gunakan script diatas

3. Menginisialisasi Scanner View

Untuk memecahkan kode-kode barcode, kita perlu menerapkan pemindai. Kita inisialisasikan ZXingScannerView pada MainAcivity.java.

```
mScannerView = new ZXingScannerView(this);
setContentView(mScannerView);
```

Gambar 7.18 Gunakan script diatas

4. Menerapkan ResultHandler

Untuk mendapatkan hasil pindai QR Code, kita akan menerapkan class ZXingScannerView.ResultHandler. Class ini akan menangani hasilnya dengan menggunakan metode handleResult(). Jadi, kita perlu menambahkan metode ini pada file MainActivity.java. Method handleResult() mendapatkan hasilnya dan menampilkannya pada alert dialog.

```
@Override
public void handleResult(Result rawResult) {
    Log.v("TAG", rawResult.getText());
    // Prints scan results
```

```
        Log.v("TAG", rawResult.getBarcodeFormat()
              .toString());
        AlertDialog.Builder builder =
            new AlertDialog.Builder(this);
        builder.setTitle("Scan Result");
        builder.setMessage(rawResult.getText());
        AlertDialog alert1 = builder.create();
        alert1.show();

        mScannerView.resumeCameraPreview(this);
    }
```

5. Menambahkan Handler untuk hasil scan

Kita tambahkan handler untuk memindai dan memulai kamera pada saat activity onResume() dan berhenti pada saat activity onPause().

```
@Override
    public void onResume() {
        super.onResume();
        mScannerView.
            setResultHandler(this);
        mScannerView.startCamera();
    }
@Override
public void onPause() {
    super.onPause();
    mScannerView.stopCamera();
}
```

Berikut tampilan source code lengkap pada MainActivity.java :

```
import android.os.Bundle;
import android.support.v7.
app.AlertDialog;
import android.support.v7.
app.AppCompatActivity;
import android.util.Log;

import com.google.zxing.Result;

import me.dm7.barcodescanner.
zxing.ZXingScannerView;

public class MainActivity extends AppCompatActivity
```

```
implements ZXingScannerView.ResultHandler {  
    private ZXingScannerView mScannerView;  
  
    @Override  
    protected void onCreate  
(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        mScannerView = new ZXingScannerView(this);  
        setContentView(mScannerView);  
    }  
    @Override  
    public void onResume() {  
        super.onResume();  
        mScannerView.setResultHandler(this);  
        mScannerView.startCamera();  
    }  
  
    @Override  
    public void onPause() {  
        super.onPause();  
        mScannerView.stopCamera();  
    }  
  
    @Override  
    public void handleResult(Result rawResult) {  
        Log.v("TAG", rawResult.getText());  
        // Prints scan results  
        Log.v("TAG", rawResult.getBarcodeFormat()  
.toString());  
        AlertDialog.Builder builder =  
        new AlertDialog.Builder(this);  
        builder.setTitle("Scan Result");  
        builder.setMessage(rawResult.getText());  
        AlertDialog alert1 = builder.create();  
        alert1.show();  
  
        mScannerView.resumeCameraPreview(this);  
    }  
}
```

BAB 8

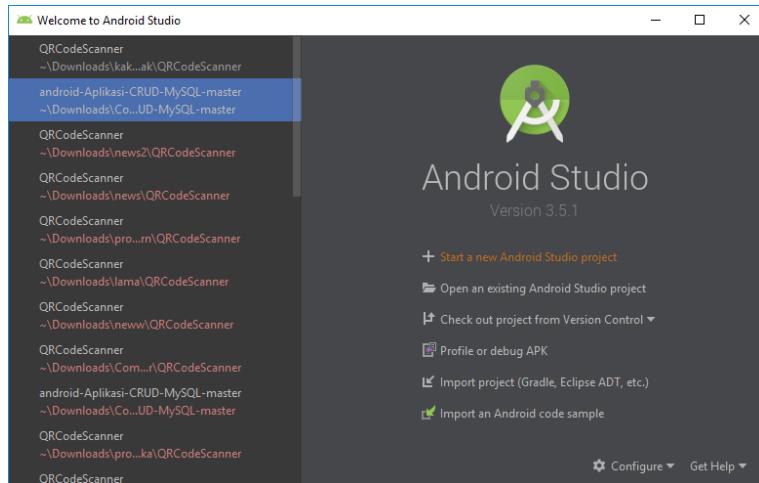
CARA MENDEBUG APLIKASI

8.1 Cara Mendebug Aplikasi

8.2 Contoh Cara Mendebug Aplikasi Hello Word

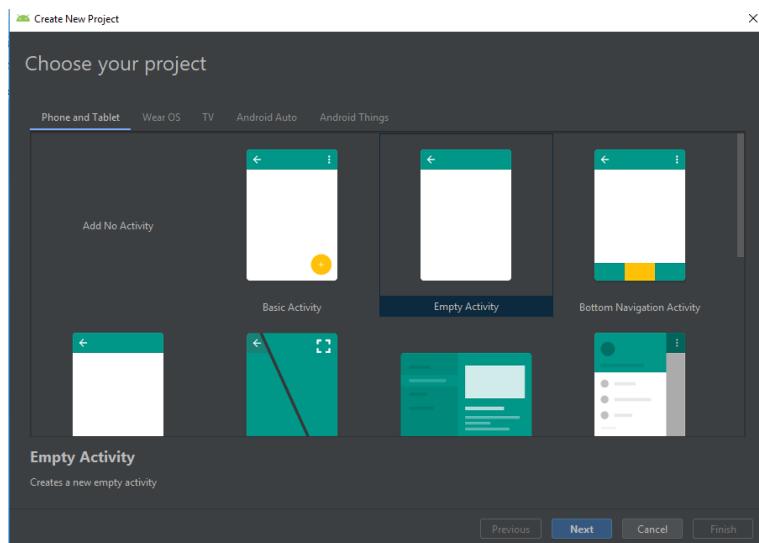
1. Pertama buka Software Android Studio

2. Pilih Start New Android Studio



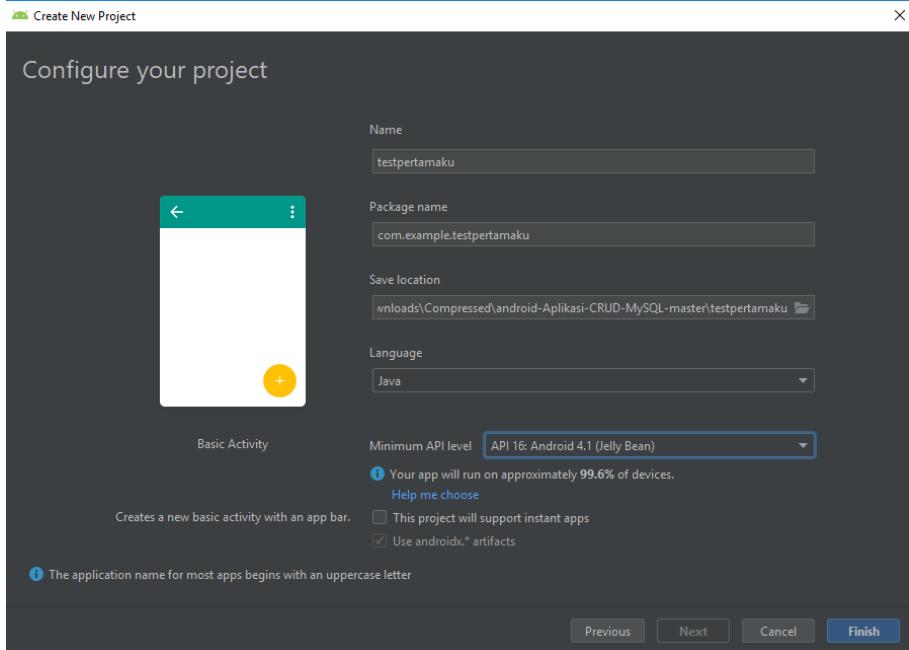
Gambar 8.1 Software Android Studio

3. Lalu pilih basic activity untuk test pertama



Gambar 8.2 Klik Empty Activity

4. Setelah itu setting configure your project dan klik finish

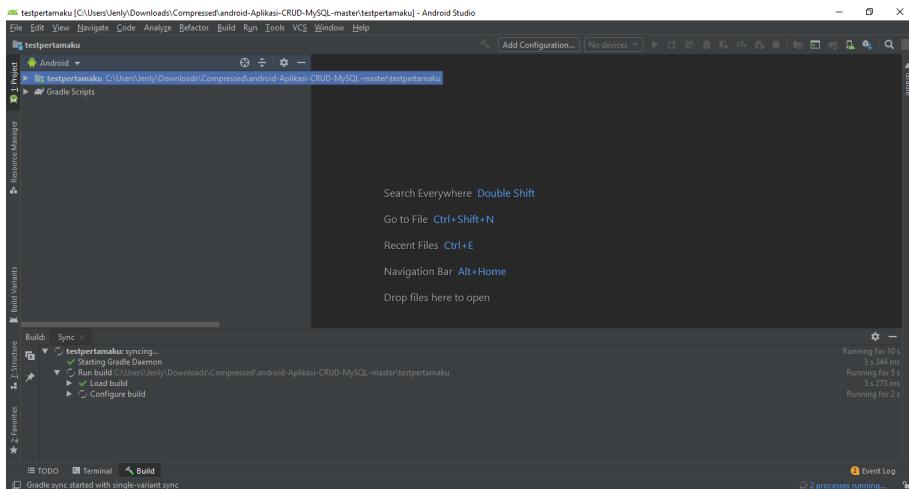


Gambar 8.3 Setting configure your project

Keterangan :

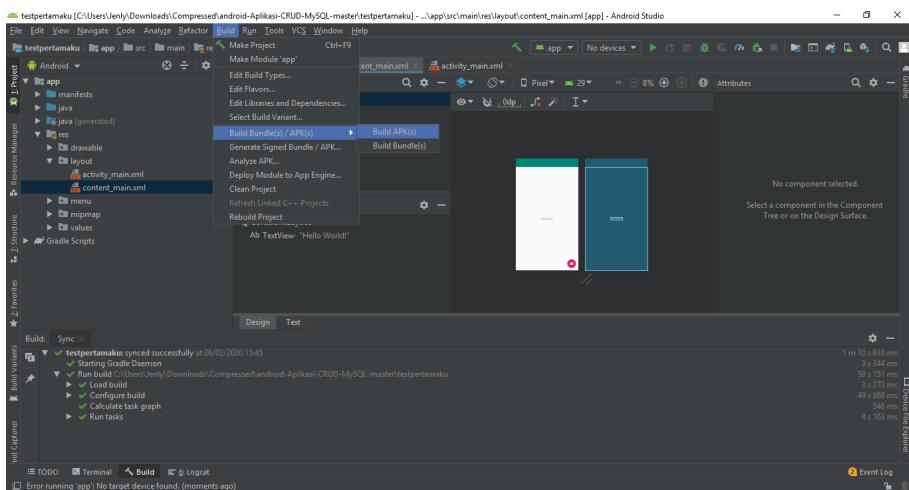
- (a) Name : Nama aplikasi yang akan dibuat
 - (b) Package Name : Merupakan paket namespace seperti di Java. Nama ini juga harus unik, biasanya Package Name akan tergenerate secara otomatis ketika sudah menginputkan Application Name.
 - (c) Save Location : Lokasi folder dari Project Android
 - (d) Minimum API level : Saya memilih hanya Phone and Tablet dengan minimum SDK API 16: Android 4.1 (Jelly Bean).
5. Tunggu build gradle sampe selesai

170 CARA MENDEBUG APLIKASI



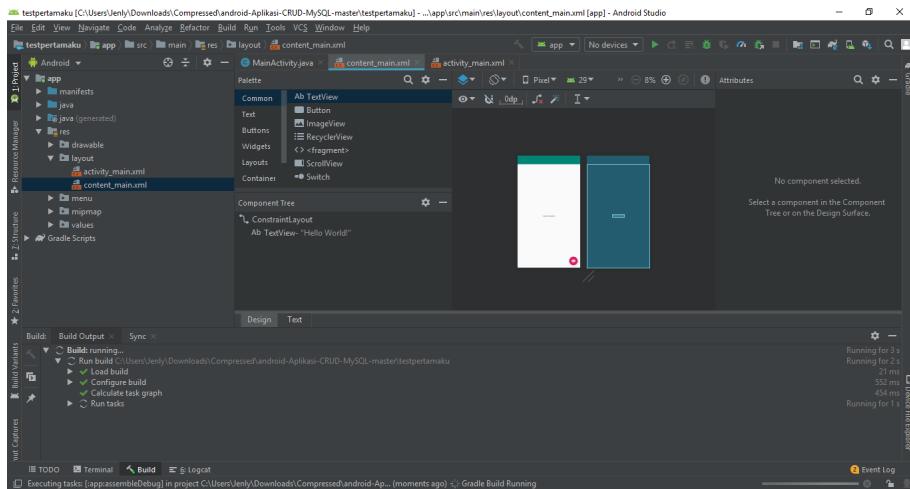
Gambar 8.4 Sedang proses build gradle

6. Setelah build gradle selesai lalu klik build lalu pilih build bundle apk kemudian build apk.



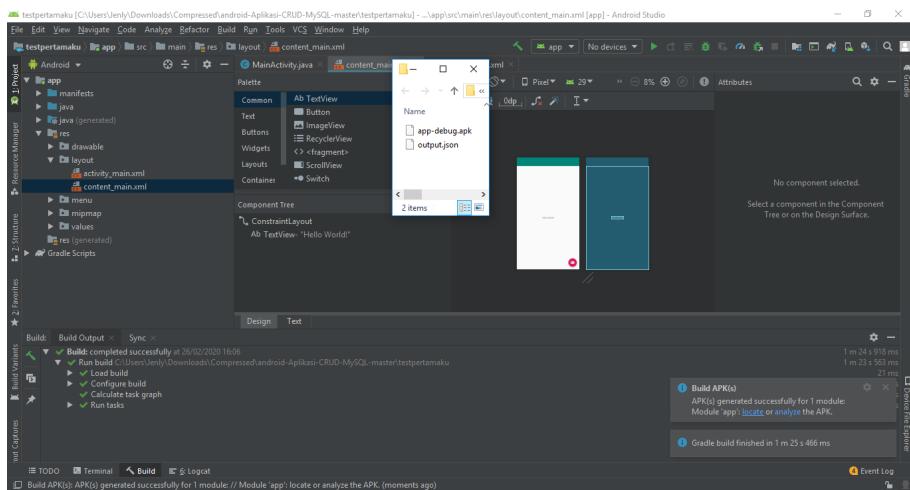
Gambar 8.5 Klik build bundle apk

7. Tunggu build apk sampai selesai.



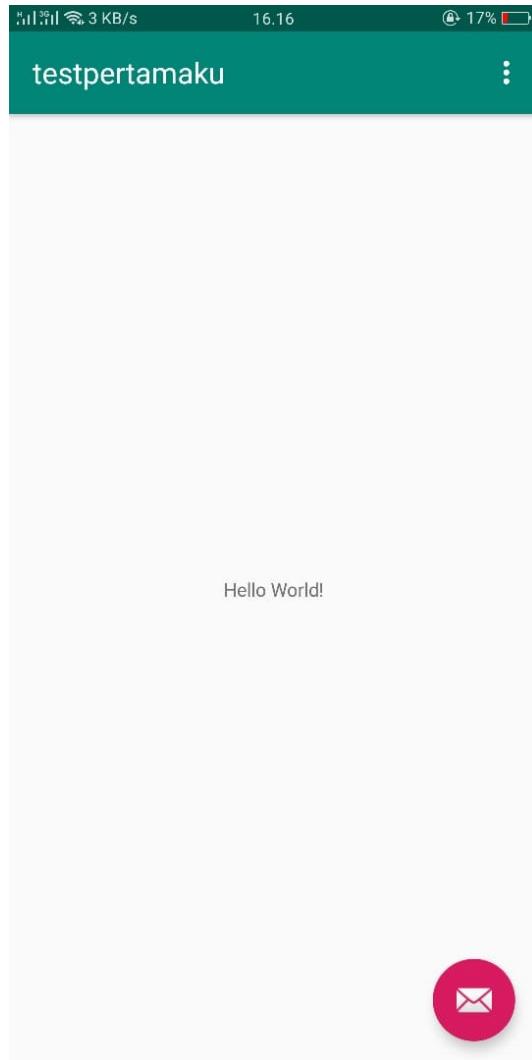
Gambar 8.6 Tunggu proses selesai

- Setelah build apk selesai lalu klik locate kemudian pindahkan app-debug.apk ke android dan install.



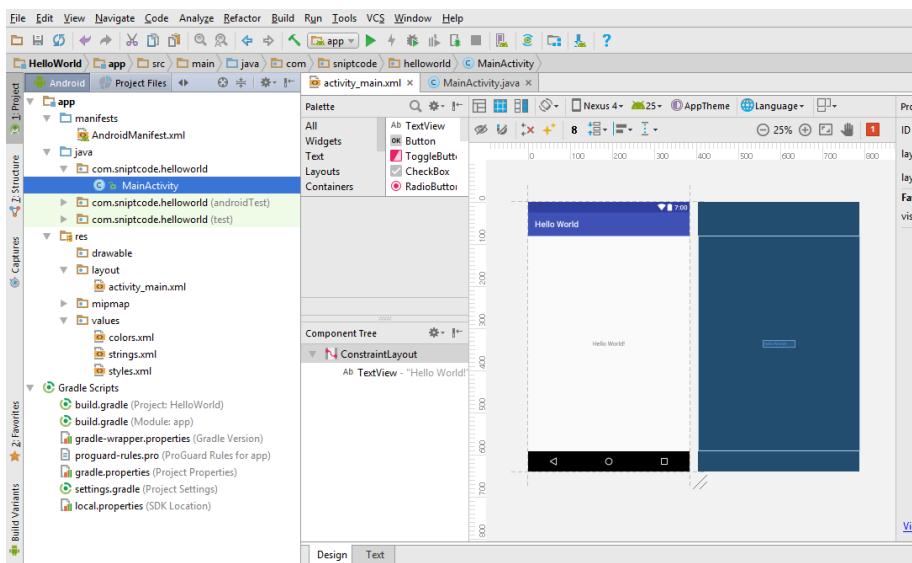
Gambar 8.7 Pindahkan app debug.apk ke android

- berikut tampilan aplikasi ketika di install.



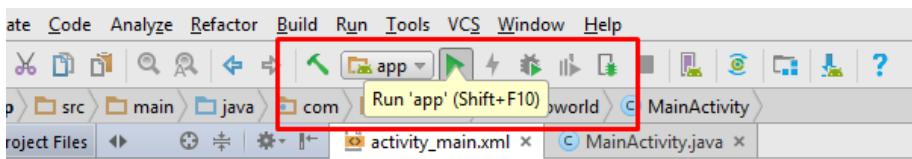
Gambar 8.8 Tampilan Hello Word pada Android

10. Selanjutnya bisa dilakukan dengan langkah sebagai berikut



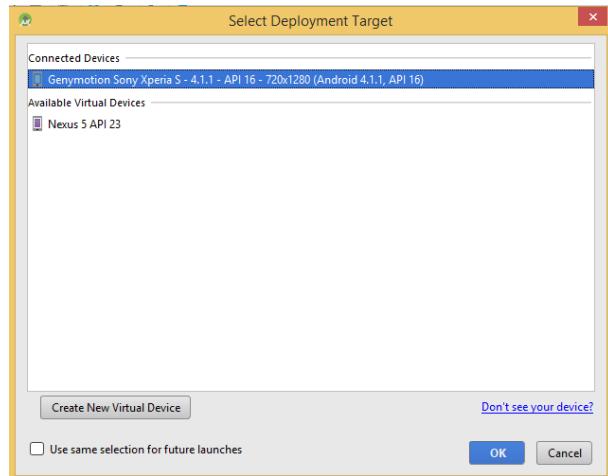
Gambar 8.9 Pilih Main Activity nya

11. Untuk Menjalankan Aplikasi Android Hello World Pertama di Android Studio bisa menekan Shift+F10 atau seperti pada gambar di bawah.



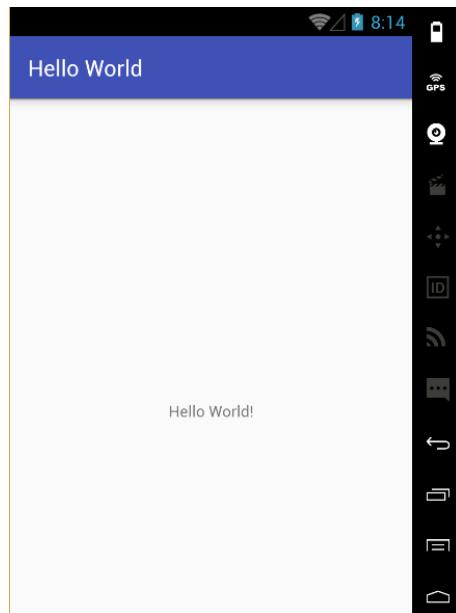
Gambar 8.10 Tekan Shift+F10

12. Pilih Devices yang akan digunakan, disini saya memakai Emulator Genymotion.



Gambar 8.11 Pilih device yang akan digunakan

13. Hasilnya seperti berikut ini:

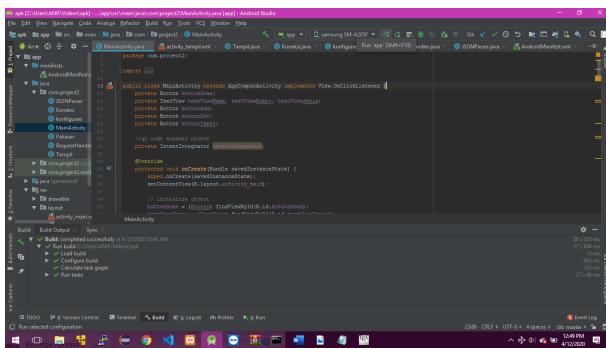


Gambar 8.12 Berikut tampilan aplikasi nya

8.3 Cara Mendebug Aplikasi Smart Gudang

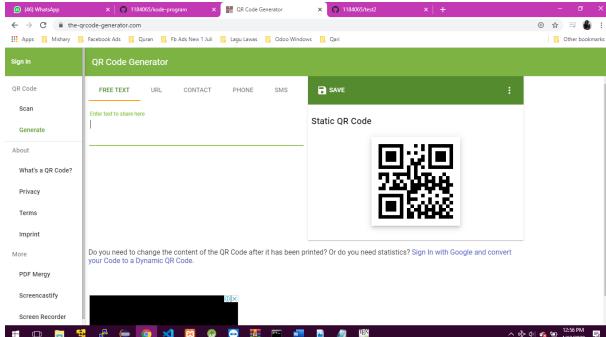
8.3.1 Cara Menambahkan Barang pada Smart Gudang

1. Buka Source Code pada MainActivity.java pada Android Studio



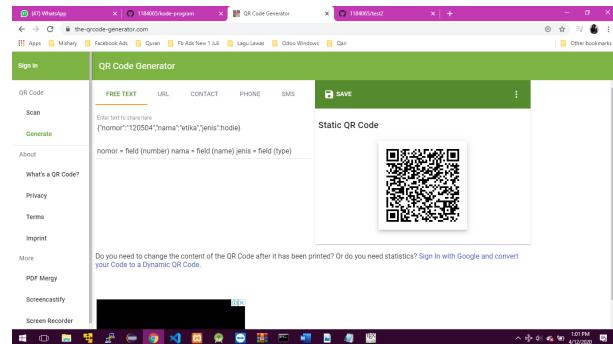
Gambar 8.13 Buka MainActivity.java lalu klik runs 'app'

2. Setelah aplikasi dirun dan sudah ter-install maka, buka website QR Code Generator pada google.
3. Buka QR Code Generator pada google



Gambar 8.14 Ini website QR Code Generator

4. Selanjutnya tuliskan perintah pada Free Text sesuai dengan yang ada kode-program



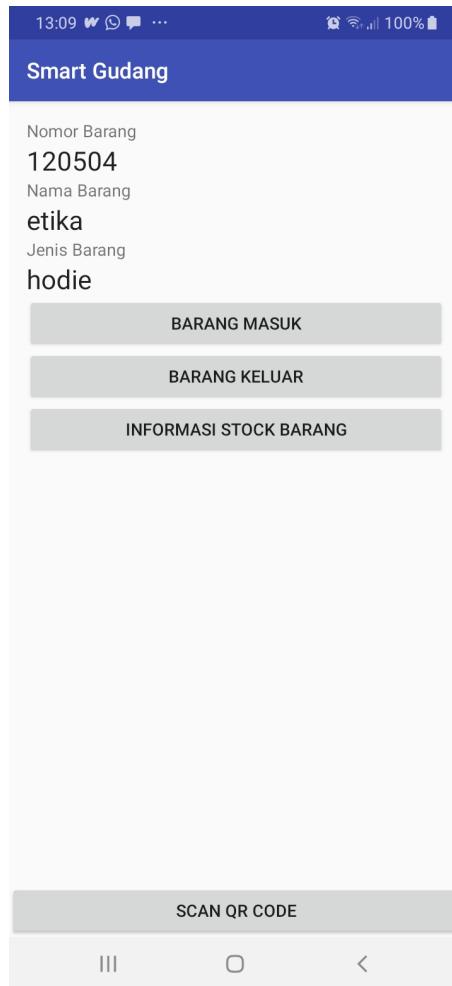
Gambar 8.15 Tuliskan perintah pada Free Text kemudian pada Scanner QR Code di save

5. QR Code pada data yang telah dibuat



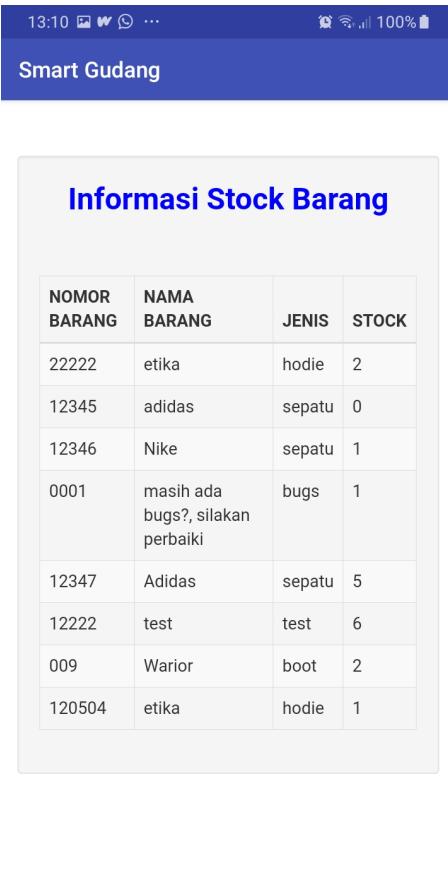
Gambar 8.16 QR Code yang telah dibuat

6. Kemudian klik icon Scan QR Barcode pada Handphone ke arah QR Code yang telah dibuat tadi
7. Lalu Data yang telah dibuat tadi akan muncul



Gambar 8.17 Data Masuk ke dalam aplikasi

8. Kemudian teman-teman klik icon Barang Masuk
9. setelah nya teman-teman bisa melihat barang yang telah ditambahkan pada icon Informasi Stock Barang



Gambar 8.18 Data Masuk ke dalam tabel Informasi Stock Barang

8.3.2 Cara Mengurangi Barang pada Smart Gudang

1. Langkah-langkahnya hampir sama dengan menambahkan barang masuk
2. pertama-tama teman-teman harus run dan install aplikasi pada Handphone
3. Klik Icon QR Code Scanner pada Handphone kemudian arahkan QR Code pada QR Code data yang telah dibuat tadi
4. Klik Icon barang keluar pada aplikasi
5. Kemudian akan muncul notifikasi bahwa barang telah berkurang



Gambar 8.19 Notifikasi barang keluar

6. Untuk melihat barang keluar bisa dilihat pada icon Informasi Stock Barang



Gambar 8.20 Tampilan barang keluar pada Infromasi Stock Barang

BAB 9

INTERFACES APLIKASI

9.1 Interfaces Aplikasi

1. Buat Source Code Untuk tampilan utamanya seperti berikut. Pada Source Code ini digunakan untuk membuat sebuah tampilan atau layout pada android dengan perintah.xml

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
  /android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/activity_main"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context="com.project2.MainActivity">
7   // ini adalah contoh terjadinya redundansi
8 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  /android"
9   xmlns:tools="http://schemas.android.com/tools" android:layout_
  _width="match_parent"
10  android:orientation="vertical"
11  android:layout_height="match_parent" android:paddingLeft="16
  dp"
12  android:paddingRight="16dp"
```

```
13     android:paddingTop="16dp"
14     android:paddingBottom="16dp" tools:context="com.project2.
>MainActivity">
15
16     // test push menambahkan konten
17     //Merubah hardcode dengan membuat string.xml
18     <TextView
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/nomor_barang" />
22
23     <TextView
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:id="@+id/textViewNomor"
27         android:textAppearance="@style/Base.TextAppearance.
>AppCompat.Large"/>
28
29     <TextView
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:text="Nama Barang" />
33
34     <TextView
35         android:layout_width="wrap_content"
36         android:layout_height="wrap_content"
37         android:id="@+id/textViewNama"
38         android:textAppearance="@style/Base.TextAppearance.
>AppCompat.Large"/>
39
40     <TextView
41         android:layout_width="wrap_content"
42         android:layout_height="wrap_content"
43         android:text="Jenis Barang" />
44
45     <TextView
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
48         android:id="@+id/textViewJenis"
49         android:textAppearance="@style/Base.TextAppearance.
>AppCompat.Large"/>
50
51     <Button
52         android:layout_width="match_parent"
53         android:layout_height="wrap_content"
54         android:text="Barang Masuk"
55         android:id="@+id/buttonAdd" />
56     <Button
57         android:layout_width="match_parent"
58         android:layout_height="wrap_content"
59         android:text="Barang Keluar"
60         android:id="@+id/buttonOut" />
61
62     <Button
63         android:layout_width="match_parent"
64         android:layout_height="wrap_content"
```

```
65         android:text="Informasi Stock Barang"
66         android:id="@+id/buttonTabel" />
67     </LinearLayout>
68
69 <Button
70     android:id="@+id/buttonScan"
71     android:layout_width="match_parent"
72     android:layout_height="wrap_content"
73     android:layout_alignParentBottom="true"
74     android:text="Scan QR Code" />
75
76 </RelativeLayout>
```

Listing 9.1 Source Code untuk membuat layout atau tampilan utama.

2. setelah buat tampilan utama atau bisa disebut dengan activity main nya selanjutnya buat MainActivity nya. Main Activity disini dimaksudkan agar ketika teman-teman membuat layout pada activity main bisa di eksekusi apakah hasilnya bisa berhasil jalan atau gagal.

```
1 package com.project2;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.app.AlertDialog;
6 import android.content.Intent;
7 import android.os.AsyncTask;
8
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import com.google.zxing.integration.android.IntentIntegrator;
15 import com.google.zxing.integration.android.IntentResult;
16 import com.project2.R;
17
18 import org.json.JSONException;
19 import org.json.JSONObject;
20
21 import java.util.HashMap;
22
23 public class MainActivity extends AppCompatActivity implements
24     View.OnClickListener {
25     private Button buttonScan;
26     private TextView textViewNama, textViewNomor, textViewJenis;
27     private Button buttonAdd;
28     private Button buttonOut;
29     private Button buttonTabel;
30
31     //qr code scanner object
32     private IntentIntegrator intentIntegrator;
33
34     @Override
35     protected void onCreate(Bundle savedInstanceState) {
36         super.onCreate(savedInstanceState);
```

```
36         setContentView(R.layout.activity_main);
37
38     // initialize object
39     buttonScan = (Button) findViewById(R.id.buttonScan);
40     textViewNomor = (TextView) findViewById(R.id.
41     textViewNomor);
42     textViewNama = (TextView) findViewById(R.id.textViewNama)
43     ;
44     textViewJenis = (TextView) findViewById(R.id.
45     textViewJenis);
46     buttonAdd = (Button) findViewById(R.id.buttonAdd);
47     buttonOut = (Button) findViewById(R.id.buttonOut);
48     buttonTabel = (Button) findViewById(R.id.buttonTabel);
49
50     // Setting listeners to button
51     buttonAdd.setOnClickListener(this);
52     buttonOut.setOnClickListener(this);
53     buttonTabel.setOnClickListener(this);
54
55     // attaching onclickListener
56     buttonScan.setOnClickListener(this);
57 }
58 //Dibawah ini merupakan perintah untuk Menambahkan Pegawai (CREATE)
59 public void reset(){
60     textViewNomor.setText("");
61     textViewNama.setText("");
62     textViewJenis.setText("");
63 }
64 private void addItem(){
65
66     final String nomor = textViewNomor.getText().toString().
67     trim();
68     final String nama = textViewNama.getText().toString().
69     trim();
70     final String jenis = textViewJenis.getText().toString().
71     trim();
72
73     class AddItem extends AsyncTask<Void,Void,String>{
74
75         ProgressDialog loading;
76
77         @Override
78         protected void onPreExecute() {
79             super.onPreExecute();
80             loading = ProgressDialog.show(MainActivity.this ,
81             "Menambahkan...","Tunggu...",false ,false );
82         }
83
84         @Override
85         protected void onPostExecute(String s) {
86             super.onPostExecute(s);
87             loading.dismiss();
88             Toast.makeText(MainActivity.this ,s ,Toast.LENGTH_
89             LONG).show();
90 }
```

```
83         reset();
84     }
85
86
87     @Override
88     protected String doInBackground(Void... v) {
89         HashMap<String ,String> params = new HashMap<>();
90         params .put(konfigurasi .KEY_EMP_NOMOR,nomor);
91         params .put(konfigurasi .KEY_EMP_NAMA,nama);
92         params .put(konfigurasi .KEY_EMP_JENIS,jenis);
93
94         RequestHandler rh = new RequestHandler();
95         String res = rh.sendPostRequest(konfigurasi .URL_
ADD, params );
96         return res;
97     }
98 }
99
100
101     AddItem ae = new AddItem();
102     ae.execute();
103 }
104
105     private void addOut(){
106
107         final String nomor = textViewNomor.getText().toString() .
trim();
108         final String nama = textViewNama.getText().toString() .
trim();
109         final String jenis = textViewJenis.getText().toString() .
trim();
110
111         class AddOut extends AsyncTask<Void ,Void ,String >{
112
113             ProgressDialog loading;
114
115             @Override
116             protected void onPreExecute() {
117                 super.onPreExecute();
118                 loading = ProgressDialog.show(MainActivity .this ,”
Mengurangi...”,”Tunggu...”,false ,false );
119             }
120
121             @Override
122             protected void onPostExecute(String s) {
123                 super.onPostExecute(s);
124                 loading.dismiss();
125                 Toast.makeText(MainActivity .this ,s ,Toast.LENGTH_
LONG).show();
126                 reset();
127             }
128
129             @Override
130             protected String doInBackground(Void... v) {
131                 HashMap<String ,String> params = new HashMap<>();
132                 params .put(konfigurasi .KEY_EMP_NOMOR,nomor);
133                 params .put(konfigurasi .KEY_EMP_NAMA,nama);
```

```

133         params.put(konfigurasi.KEY_EMP_JENIS, jenis);
134
135         RequestHandler rh = new RequestHandler();
136         String res = rh.sendPostRequest(konfigurasi.URL_
137         DEL, params);
138         return res;
139     }
140
141     AddOut ae = new AddOut();
142     ae.execute();
143 }
144 // Mendapatkan hasil scan
145
146
147 @Override
148 protected void onActivityResult(int requestCode, int
149 resultCode, Intent data) {
150     IntentResult result = IntentIntegrator.
151     parseActivityResult(requestCode, resultCode, data);
152     if (result != null) {
153         if (result.getContents() == null) {
154             Toast.makeText(this, "Hasil tidak ditemukan",
155             Toast.LENGTH_SHORT).show();
156         } else {
157             // jika qrcode berisi data
158             try {
159                 // converting the data json
160                 JSONObject object = new JSONObject(result.
161                 getContents());
162                 // atur nilai ke textviews
163                 textViewNomor.setText(object.getString("nomor"
164                 ""));
165                 textViewNama.setText(object.getString("nama")
166                 );
167                 textViewJenis.setText(object.getString("jenis"
168                 ""));
169             } catch (JSONException e) {
170                 e.printStackTrace();
171                 // jika format encoded tidak sesuai maka
172                 hasil
173                     // ditampilkan ke toast
174                     Toast.makeText(this, result.getContents(),
175                     Toast.LENGTH_SHORT).show();
176             }
177         }
178     } else {
179         super.onActivityResult(requestCode, resultCode, data)
180     }
181 }
182
183 @Override
184 public void onClick(View v) {
185     // inisialisasi IntentIntegrator(scanQR)
186     if (v == buttonScan) {
187

```

```

178         intentIntegrator = new IntentIntegrator(this);
179         intentIntegrator.initiateScan();
180     }
181
182     if (v == buttonAdd) {
183         addItem();
184     }
185
186     if (v == buttonOut) {
187         addOut();
188     }
189
190     if (v == buttonTabel) {
191         startActivity(new Intent(this, Tampil.class));
192     }
193 }
194 }
```

Listing 9.2 Source Code untuk mengeksekusi layout atau tampilan utama.

3. kemudian buat `tampil_activity.xml` untuk membuat tampilan pada `web_view` nya

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
   /android"
2   xmlns:tools="http://schemas.android.com/tools" android:layout_
   _width="match_parent"
3   android:layout_height="match_parent"
4   tools:context="com.project2.Tampil">
5
6   <WebView
7     android:id="@+id/activity_main_webview"
8     android:layout_width="match_parent"
9     android:layout_height="match_parent" />
10
11 </RelativeLayout>
```

Listing 9.3 Source Code untuk membuat layout tampil web view.

4. setelah dibuat source code untuk layout tampil maka untuk mengeksekusi nya kita membuat `source_code tampil.java`

```

1 package com.project2;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 import android.webkit.WebSettings;
7 import android.webkit.WebView;
8 import android.webkit.WebViewClient;
9
10 import com.project2.R;
11
12 public class Tampil extends AppCompatActivity {
13
14     WebView webviewku;
15     WebSettings websettingku;
```

```

16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_tampil);
21
22         webviewku = (WebView) findViewById(R.id.activity_main_
23             webview);
24
25         websettingku = webviewku.getSettings();
26
27         webviewku.setWebViewClient(new WebViewClient());
28         webviewku.loadUrl("https://projectpoltekpos.000webhostapp
29             .com/");
30
31     }
32
33     @Override
34     public void onBackPressed() {
35         if(webviewku.canGoBack()) {
36             webviewku.goBack();
37         } else {
38             super.onBackPressed();
39     }

```

Listing 9.4 Source Code untuk mengeksekusi layout tampil atau web view nya.

5. Selanjutnya buat koneksi, untuk koneksi disini dimaksudkan untuk menyambungkan dari android studio supaya bisa terhubung atau terkoneksi dengan website pada web hosting yang telah kita buat.

```

1 package com.project2;
2
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.io.InputStreamReader;
6 import java.net.HttpURLConnection;
7 import java.net.URL;
8 import java.net.URLConnection;
9
10 public class Koneksi {
11
12     public String call(String url) {
13         int BUFFER_SIZE = 2000;
14         InputStream in = null;
15         try {
16             in = OpenHttpConnection(url);
17         } catch (IOException e) {
18             e.printStackTrace();
19             return "";
20         }
21         InputStreamReader isr = new InputStreamReader(in);
22         int charRead;
23         String str = "";

```

```

24     char[] inputBuffer = new char[BUFFER_SIZE];
25     try {
26         while ((charRead = isr.read(inputBuffer)) > 0) {
27             String readString = String.valueOf(
28                 inputBuffer, 0, charRead);
29             str += readString;
30             inputBuffer = new char[BUFFER_SIZE];
31         }
32     } catch (IOException e) {
33         // Handle Exception
34         e.printStackTrace();
35         return "";
36     }
37     return str;
38 }
39
40 private InputStream OpenHttpConnection(String url) throws
41 IOException {
42     InputStream in = null;
43     int response = -1;
44     URL url1 = new URL(url);
45     URLConnection conn = url1.openConnection();
46     if (!(conn instanceof HttpURLConnection))
47         throw new IOException("Not An Http Connection");
48     try {
49         HttpURLConnection httpconn = (HttpURLConnection) conn
50 ;
51         httpconn.setAllowUserInteraction(false);
52         httpconn.setInstanceFollowRedirects(true);
53         httpconn.setRequestMethod("GET");
54         httpconn.connect();
55         response = httpconn.getResponseCode();
56         if (response == HttpURLConnection.HTTP_OK) {
57             in = httpconn.getInputStream();
58         }
59     } catch (Exception e) {
60         throw new IOException("Error connecting 2");
61     }
62     return in;
63 }

```

Listing 9.5 Source Code Koneksi.

6. selanjutnya source_code konfigurasi nya

```

1 package com.project2;
2
3 public class konfigurasi {
4
5     // ini adalah tujuan request
6     public static final String URL_ADD="https://projectpoltekpos
7     .000webhostapp.com/android/tambah.php";
8     public static final String URL_DEL = "https://
9     projectpoltekpos.000webhostapp.com/android/tambah2.php";

```

```

8     public static final String URL_GET_ALL = "https ://  

9         projectpoltekpos .000 webhostapp .com / android / tampilSemua .php";  

10    public static final String URL_GET_EMP = "https ://  

11        projectpoltekpos .000 webhostapp .com / android / tampilPgw .php?id  

12        =";  

13    public static final String URL_UPDATE_EMP = "https ://  

14        projectpoltekpos .000 webhostapp .com / android / updatePgw .php";  

15    public static final String URL_DELETE_EMP = "https ://  

16        projectpoltekpos .000 webhostapp .com / android / hapusPgw .php?id=";  

17  

18 //Dibawah ini merupakan Kunci yang akan digunakan untuk  

19 mengirim permintaan ke Skrip PHP  

20 // ini adalah nama kolom yang harus disesuaikan dengan  

21 database  

22 public static final String KEY_EMP_ID = "id";  

23 public static final String KEY_EMP_NOMOR = "nomor";  

24 public static final String KEY_EMP_NAMA = "nama";  

25 public static final String KEY_EMP_JENIS = "jenis"; //desg  

26 itu variabel untuk posisi  

27  

28 //JSON Tags  

29 public static final String TAG_JSON_ARRAY="result";  

30 public static final String TAG_ID = "id";  

31 public static final String TAG_NAMA = "nama";  

32 public static final String TAG_JENIS = "jenis";  

33  

34 //ID  

35 //emp itu singkatan dari Employee  

36 public static final String EMP_ID = "emp_id";  

37 }

```

Listing 9.6 Source Code Konfigurasi.

7. selanjutnya buat requestHandler

```

1 package com . project 2;  

2  

3 import java . io . BufferedReader;  

4 import java . io . BufferedWriter;  

5 import java . io . InputStreamReader;  

6 import java . io . OutputStream;  

7 import java . io . OutputStreamWriter;  

8 import java . io . UnsupportedEncodingException;  

9 import java . net . HttpURLConnection;  

10 import java . net . URL;  

11 import java . net . URLEncoder;  

12 import java . util . HashMap;  

13 import java . util . Map;  

14  

15 import javax . net . ssl .HttpsURLConnection ;  

16  

17 public class RequestHandler {  

18     //Metode Untuk mengirim httpPostRequest  

19     //Metode ini mengambil 2 Argumen  

20     //Metode Pertama adalah URL dari Skrip yang digunakan untuk  

21     mengirimkan permintaan

```

```
21 //Yang lainnya adalah HashMap dengan nilai pasangan nama yang
22 //berisi data yang akan dikirim dengan permintaan
23
24 public String sendPostRequest(String requestURL,
25                                 HashMap<String , String>
26                                 postDataParams) {
27     //Membuat URL
28     URL url ;
29
30     //Objek StringBuilder untuk menyimpan pesan diambil dari
31     //server
32     StringBuilder sb = new StringBuilder();
33     try {
34         //Inisialisasi URL
35         url = new URL(requestURL);
36
37         //Membuat Koneksi HttpURLConnection
38         HttpURLConnection conn = (HttpURLConnection) url.
39         openConnection();
40
41         //Konfigurasi koneksi
42         conn.setReadTimeout(15000);
43         conn.setConnectTimeout(15000);
44         conn.setRequestMethod("POST");
45         conn.setDoInput(true);
46         conn.setDoOutput(true);
47
48         //Membuat Keluaran Stream
49         OutputStream os = conn.getOutputStream();
50
51         //Menulis Parameter Untuk Permintaan
52         //Kita menggunakan metode getpostDataString yang
53         //didefinisikan di bawah ini
54         BufferedWriter writer = new BufferedWriter(
55             new OutputStreamWriter(os , "UTF-8"));
56         writer.write(getpostDataString(postDataParams));
57
58         writer.flush();
59         writer.close();
60         os.close();
61         int responseCode = conn.getResponseCode();
62
63         if (responseCode ==HttpsURLConnection.HTTP_OK) {
64
65             BufferedReader br = new BufferedReader(new
66             InputStreamReader(conn.getInputStream()));
67             sb = new StringBuilder();
68             String response;
69             //Reading server response
70             while ((response = br.readLine()) != null){
71                 sb.append(response);
72             }
73         }
74
75     } catch (Exception e) {
76         e.printStackTrace();
77     }
```

```
71     }
72     return sb.toString();
73 }
74
75 public String sendGetRequest(String requestURL){
76     StringBuilder sb =new StringBuilder();
77     try {
78         URL url = new URL(requestURL);
79         HttpURLConnection con = (HttpURLConnection) url.
openConnection();
80         BufferedReader bufferedReader = new BufferedReader(
new InputStreamReader(con.getInputStream()));
81
82         String s;
83         while((s=bufferedReader.readLine())!=null){
84             sb.append(s+"\n");
85         }
86     }catch(Exception e){
87     }
88     return sb.toString();
89 }
90
91 public String sendGetRequestParam(String requestURL, String
id){
92     StringBuilder sb =new StringBuilder();
93     try {
94         URL url = new URL(requestURL+id);
95         HttpURLConnection con = (HttpURLConnection) url.
openConnection();
96         BufferedReader bufferedReader = new BufferedReader(
new InputStreamReader(con.getInputStream()));
97
98         String s;
99         while((s=bufferedReader.readLine())!=null){
100            sb.append(s+"\n");
101        }
102    }catch(Exception e){
103    }
104    return sb.toString();
105 }
106
107 private String getPostDataString(HashMap<String , String>
params) throws UnsupportedEncodingException {
108     StringBuilder result = new StringBuilder();
109     boolean first = true;
110     for (Map.Entry<String , String> entry : params.entrySet())
{
111         if (first)
112             first = false;
113         else
114             result.append("&");
115
116         result.append(URLEncoder.encode(entry.getKey() , "UTF
-8"));
117         result.append("=");
118 }
```

```
118     result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
119 }
120
121     return result.toString();
122 }
123 }
```

Listing 9.7 Source Code RequestHandler.

8. Buat JSONnya, fungsi dari JSON sendiri yaitu JSON (JavaScript Object Notation) adalah sebuah format data yang digunakan untuk pertukaran dan penyimpanan data.

```
1 package com.project2;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.InputStreamReader;
6 import java.io.OutputStream;
7 import java.io.OutputStreamWriter;
8 import java.io.UnsupportedEncodingException;
9 import java.net.HttpURLConnection;
10 import java.net.URL;
11 import java.net.URLEncoder;
12 import java.util.HashMap;
13 import java.util.Map;
14
15 import javax.net.ssl.HttpsURLConnection;
16
17 public class RequestHandler {
18     //Metode Untuk mengirim httpPostRequest
19     //Metode ini mengambil 2 Argumen
20     //Metode Pertama adalah URL dari Skrip yang digunakan untuk
21     //mengirimkan permintaan
22     //Yang lainnya adalah HashMap dengan nilai pasangan nama yang
23     //berisi data yang akan dikirim dengan permintaan
24
25     public String sendPostRequest(String requestURL,
26                                     HashMap<String, String>
27                                     postDataParams) {
28         //Membuat URL
29         URL url;
30
31         //Objek StringBuilder untuk menyimpan pesan diambil dari
32         //server
33         String Builder sb = new String Builder();
34         try {
35             //Inisialisasi URL
36             url = new URL(requestURL);
37
38             //Membuat Koneksi HttpURLConnection
39             HttpURLConnection conn = (HttpURLConnection) url.
40             openConnection();
41
42             //Konfigurasi koneksi
```

```

38         conn . setReadTimeout(15000) ;
39         conn . setConnectTimeout(15000) ;
40         conn . setRequestMethod("POST") ;
41         conn . setDoInput(true) ;
42         conn . setDoOutput(true) ;
43
44         // Membuat Keluaran Stream
45         OutputStream os = conn . getOutputStream() ;
46
47         // Menulis Parameter Untuk Permintaan
48         // Kita menggunakan metode getPostDataString yang
didefinisikan di bawah ini
49         BufferedWriter writer = new BufferedWriter(
50             new OutputStreamWriter(os , "UTF-8")) ;
51         writer . write(getpostDataString(postDataParams)) ;
52
53         writer . flush() ;
54         writer . close() ;
55         os . close() ;
56         int responseCode = conn . getResponseCode() ;
57
58         if (responseCode ==HttpsURLConnection.HTTP_OK) {
59
60             BufferedReader br = new BufferedReader(new
InputStreamReader(conn . getInputStream())) ;
61             sb = new StringBuilder() ;
62             String response;
63             // Reading server response
64             while ((response = br . readLine()) != null){
65                 sb.append(response) ;
66             }
67         }
68     } catch (Exception e) {
69         e . printStackTrace() ;
70     }
71     return sb . toString() ;
72 }
73
74 public String sendGetRequest(String requestURL){
```

Listing 9.8 Source Code JSONParser.java.

9. Source Code Manifest Android, Source Code ini merupakan bawaan dari Aplikasi Android sendiri yang mana dalam Source Code ini berpengaruh terhadap activity main dan main activity nya.

```

1 package com . project 2;
2
3 import java . io . BufferedReader ;
4 import java . io . BufferedWriter ;
5 import java . io . InputStreamReader ;
6 import java . io . OutputStream ;
7 import java . io . OutputStreamWriter ;
8 import java . io . UnsupportedEncodingException ;
9 import java . net . HttpURLConnection ;
```

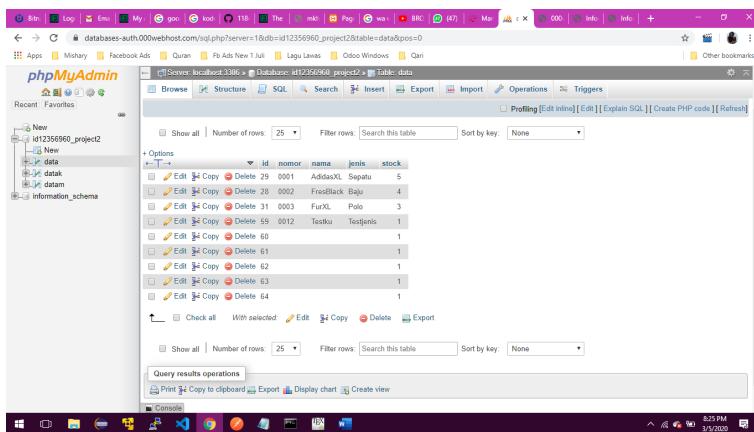
```

10 import java.net.URL;
11 import java.net.URLEncoder;
12 import java.util.HashMap;
13 import java.util.Map;
14
15 import javax.net.ssl.HttpsURLConnection;
16
17 public class RequestHandler {
    //Metode Untuk mengirim httpPostRequest
    //Metode ini mengambil 2 Argumen
    //Metode Pertama adalah URL dari Skrip yang digunakan untuk
    mengirimkan permintaan
    //Yang lainnya adalah HashMap dengan nilai pasangan nama yang
    berisi data yang akan dikirim dengan permintaan
18
19    public String sendPostRequest(String requestURL,
                                   HashMap<String, String>
20                                   postDataParams) {

```

Listing 9.9 Source Code Manifest Android.

10. Halaman untuk membuat database dengan PHP My Admin.



Gambar 9.1 Database yang digunakan

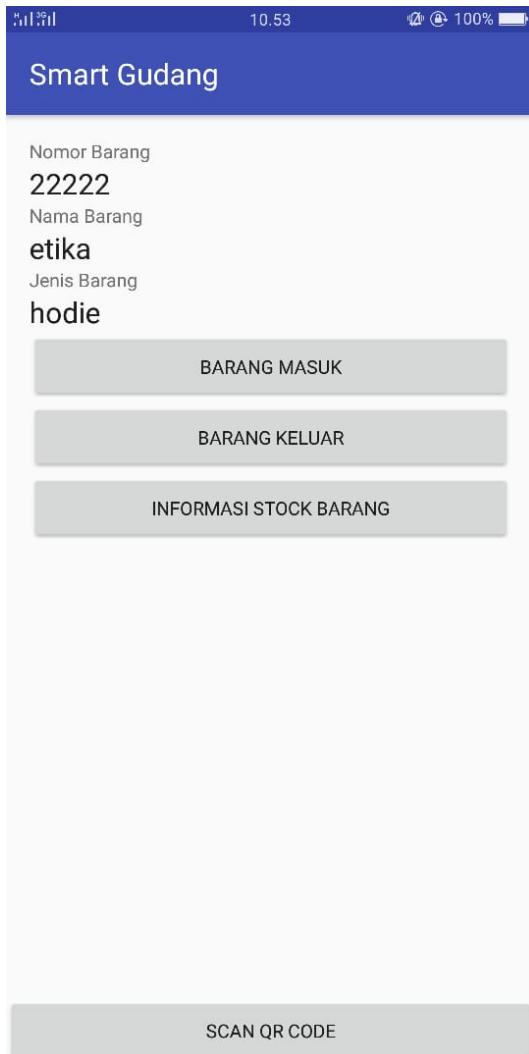
9.2 Tampilan Aplikasi

1. Tampilan untuk Aplikasi yang sudah jadi. Setelah semua source code selesai dan di debug aplikasi nya tidak ada yang error maka akan muncul tampilan seperti ini.
2. Tampilan Aplikasi yg belum terisi data



Gambar 9.2 Tampilan yang belum terisi data

3. Tampilan aplikasi yang sudah terisi data



Gambar 9.3 Tampilan yang sudah terisi data

Pertama kali user masuk Android akan disuguhkan pada halaman utama yaitu dapat dilihat seperti gambar diatas. Setelah itu jika user ingin menginputkan barang masuk yang diinginkan maka user masuk terlebih dahulu pada halaman website yang sudah kita sediakan. Kemudian arahkan scan barcode pada kamera lalu akan di proses kemudian data akan masuk ke tabel input barang dan untuk melihat hasilnya yaitu dengan masuk ke tabel informasi stock barang.

4. Informasi dari stock barang yang ada.

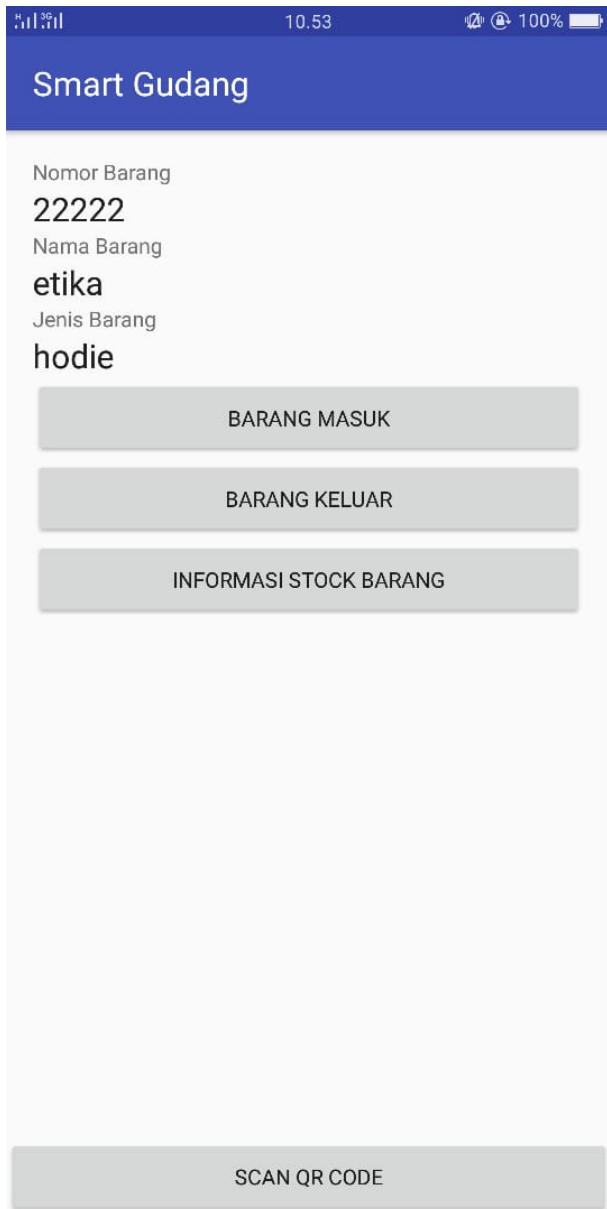


NOMOR BARANG	NAMA BARANG	JENIS	STOCK
22222	etika	hodie	2
12345	adidas	sepatu	0
12346	Nike	sepatu	1
0001	masih ada bugs?, silakan perbaiki	bugs	1
12347	Adidas	sepatu	5
12222	test	test	6
009	Warior	boot	2
120504	etika	hodie	1



Gambar 9.4 Informasi Stock dengan web view

5. Tampilan Aplikasi dengan potret. Untuk tampilan aplikasi pada hanphone apabila di buat dengan posisi potret maka hasilnya akan seperti ini.



Gambar 9.5 Tampilan Aplikasi Potret

6. Tampilan Aplikasi dengan landscape. Untuk tampilan aplikasi pada hanphone apabila di buat dengan posisi landscape maka hasilnya akan seperti ini.



Gambar 9.6 Tampilan aplikasi landscape

9.3 Tata Cara Penggunaan Aplikasi

1. Pertama, pastikan teman-teman sudah mendownload aplikasi smart gudang ini
2. kemudian, pastikan semua barang sudah terdata
3. Arahkan barcode pada Aplikasi Smart Gudang
4. Kedua, setelah diarahkan data yang telah di scan akan terdaftar pada barang masuk
5. Ketiga, untuk barang keluar juga sama, arahkan barcode ke aplikasi smart gudang
6. Keempat, setelah diarahkan data akan berhasil untuk barang keluar

9.4 Mengatasi Error

1. Issues 1 (Jenis Error : Hardcoded text)

Activity_main.xml

```
<Button  
    android:id="@+id/buttonScan"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:text="Scan QR Code" />
```

String.xml

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

Keterangan dan pemecahan masalah : Ini hanya peringatan, fungsi tetap bekerja sebagai mana mestinya. direkomendasikan untuk mendefine string "text" di file string.xml . alasannya agar membuat perubahan di masa depan lebih sederhana dan lebih mudah untuk referensi di beberapa halaman

Perbaikan kode progam : Define string, ubah file activity_main.xml menjadi:

```
<TextView
    android:layout_width=
    "match_parent"
    android:layout_height=
    "wrap_content"
    android:text=
    "@string/string_name"
    />
```

Membuat string di file strings.xml. ubah file strings.xml menjadi:

```
<resources>
    <string name="app_name">QRCodeScanner</string>
    <string name="scan_qr_code"
        >Scan Qr Code</string>
</resources>
```

2. Issues 2 (Jenis Error : MissingPrefix)

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text=Tabel
    android:id="@+id/buttonTabel" />
```

Perbaikan kode progam :

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tabel"
    android:id="@+id/buttonTabel" />
```

3. Issues 3 (Jenis Error : variable 'x' is already defined in scope)

```
RequestHandler rh = new RequestHandler();
String rh = rh.sendPostRequest(konfigurasi.URL_ADD, params);
return rh;
```

Keterangan dan solusi : tidak dapat memanggil variabel sendidiri dan membuat variabel lokal dengan nama yang sama di java. solusi yang dapat dilakukan adalah mengganti nama salah variable

Perbaikan Kode Program

```
RequestHandler rh = new RequestHandler();
String res = rh.sendPostRequest(konfigurasi.URL_ADD, para
return res;
```

4. Issues 4 (Jenis Error : Incompatible Types) Lokasi File MainActivity.java

```
buttonScan = (\Button") findViewById(R.id.buttonScan);
buttonAdd = (\Button") findViewById(R.id.buttonAdd);
buttonTabel = (\Button") findViewById(R.id.buttonTabel);
```

Keterangan dan pemecahan masalah : ini merupakan sebuah error pada MainActivity.java karena penggunaan dalam syntak salah yaitu dengan menggunakan tanda kutip pada kode Button.

Perbaikan dalam kode program :

```
buttonScan = (Button) findViewById(R.id.buttonScan);
buttonAdd = (Button) findViewById(R.id.buttonAdd);
buttonTabel = (Button) findViewById(R.id.buttonTabel);
```

5. Issues 5 (Jenis Error : Incompatible Types)

Lokasi File MainActivity.java

```
buttonScan = (\Button") findViewById(R.id.buttonScan);
buttonAdd = (\Button") findViewById(R.id.buttonAdd);
buttonTabel = (\Button") findViewById(R.id.buttonTabel);
```

Keterangan dan pemecahan masalah : ini merupakan sebuah error pada MainActivity.java karena penggunaan dalam syntak salah yaitu dengan menggunakan tanda kutip pada kode Button.

Perbaikan dalam kode program :

```
buttonScan = (Button) findViewById(R.id.buttonScan);
buttonAdd = (Button) findViewById(R.id.buttonAdd);
buttonTabel = (Button) findViewById(R.id.buttonTabel);
```

6. Issues 6 (Field can be converted to a local variable) Lokasi File MainActivity.java

Keterangan dan pemecahan masalah : ini merupakan sebuah warning atau peringatan pada MainActivity.java karena pada IntentIntegrator kelebihan dalam bidang kelasnya dan jika kelebihan bisa diganti dengan variable lokal

7. Issues 7 (Cannot resolve symbol ‘parent’)

```
protected void onActivityResult(int requestCode,
int resultCode, Intent data) {
IntentResult result = IntentIntegrator.parseActivityResult
(requestCode, resultCode, data);
if (result != null)
```

Keterangan dan Solusi: Library belum diimport solusinya dengan cara mengimport library jenis zxing: import com.google.zxing.integration.android.IntentIntegrator; import com.google.zxing.integration.android.IntentResult;

8. Issues 8(Error php : Undefined variable: row in X on line Y)

```
<tbody>
<?php
include('conn.php');
$query=mysqli_query("select
* from `data`");
while($row=mysqli_fetch_array($query))
{
?>
<tr>
<td><?php echo $row['nomor']; ?><td>
<td><?php echo $row['nama']; ?></td>
<td><?php echo $row['jenis']; ?><td>
<td><?php echo $row['stock']; ?><td>
</tr>
<?php
}
?>
</tbody>
..
```

Perbaikan Kode Program:

```
.. <?php
```

```
include('conn.php');
$query=mysqli_query($conn,
"select      *from `data`");
while ($row=mysqli_fetch_array($query))  {
?>
<tr>
```

9. Issues 9 (Error Installing APK)

Masalah : USB debugging belum diaktifkan. Buka Settings perangkat Anda, lalu pilih Developer Options, dan pastikan USB Debugging diaktifkan. Jika Anda tidak melihat Developer Options di menu Settings, pilih About Phone dan tekan terus Build Number sampai pemberitahuan You are now a developer muncul. Kembali ke layar Settings utama, dan Anda harus menemukan bahwa Developer Options telah ditambahkan.

10. Issues 10(Cannot Resolve Symbol R) Masalah Cannot Resolve Symbol R ini bisa juga terjadi setelah kita memindahkan file atau direktori java ke lokasi lain.

- (a) Klik menu File pada Android Studio.
- (b) lalu Klik Invalidate Cache/ Restart

BAB 10

KESIMPULAN DAN SARAN

10.1 Kesimpulan dan Saran

10.1.1 Kesimpulan

Setelah melakukan studi literatur, observasi, perancangan, analisis, implementasi dan pengujian website untuk sistem informasi stok persediaan barang maka dapat disimpulkan :

1. Dengan aplikasi yang di rancang dapat mempermudah pengolahan data persediaan barang.
2. Dengan menggunakan android untuk merancang sistem informasi persediaan barang ini sehingga memudahkan pengaksesan sistem yang di rancang.
3. Sistem informasi persediaan barang dagang ini berhasil melakukan proses pengolahan data barang persediaan barang .

10.1.2 Saran

Berdasarkan kesimpulan diatas, maka saran yang diharapkan yaitu dilakukan suatu pengembangan sistem informasi persediaan barang :

1. Menambahkan fitur notifikasi dering pemberitahuan apabila ada barang masuk yang di terima oleh admin dari sistem gudang.
2. Menambah fasilitas keamanan data, seperti fasilitas backup data.
3. Menambah fasilitas help pada aplikasi yang dibuat agar lebih user friendly.
4. Interface sistem yang dibangun masih tampak sederhana dapat dikembangkan lebih menarik.

DAFTAR PUSTAKA

1. @articleyudha2017perancangan, title=Perancangan Aplikasi Sistem Inventory Barang Menggunakan Barcode Scanner Berbasis Android, author=Yudha, I Putu Alit Putra and Sudarma, Made and Mertasana, Putu Arya, journal=Jurnal SPEKTRUM, volume=4, number=2, pages=72–80, year=2017
2. PetaniKode, www.petanikode.com/tutorial/java
3. CodePolitan, www.codepolitan.com
4. , Dicoding.com, <https://www.dicoding.com>

