

**APLIKASI BAHASA PEMROGRAMAN JAVA  
DISERTAI COMPILER BERBASIS ANDROID  
(STUDI KASUS PEMROGRAMAN MENENGAH  
IIB DARMAJAYA)**



Disusun Oleh :

**GALUH WIYENDA MUHAMMAD  
1411010018**

**FAKULTAS ILMU KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA  
BANDAR LAMPUNG  
2019**



## PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 8 Maret 2019



**Galuh Wiyenda M**  
NPM. 1411010018

**HALAMAN PERSETUJUAN**

**Judul Skripsi : APLIKASI BAHASA PEMOGRAMAN JAVA  
DISERTAI COMPILER BERBASIS ANDROID  
(STUDI KASUS PEMOGRAMAN MENENGAH  
IIB DARMAJAYA)**

**Nama Mahasiswa : Galuh Wiyenda Muhammad**

**No. Pokok Mahasiswa : 1411010018**

**Jurusan : Teknik Informatika**



**Menyetujui :**

**Dosen Pembimbing**

**Ketua Jurusan Teknik Informatika**

*SABD*

*[Signature]*

**Apri Triansyah, S.Kom., M.T.I  
NIK 12720212**

**Yuni Arkhiansyah, S.Kom., M.Kom  
NIK 00480802**

## HALAMAN PENGESAHAN

Telah Diuji dan Dipertahankan Didepan Tim Penguji Skripsi  
Jurusan Teknik Informatika Institut Informatika dan Bisnis Darmajaya  
Bandar Lampung dan Dinyatakan Diterima untuk  
Memenuhi Syarat Guna Memperoleh  
Gelar Sarjana Komputer

### Mengesahkan

1. Tim Penguji

Tanda Tangan

Ketua : **Suhendro Yusuf Irianto., Ph.D**

Anggota : **Sulyono, S.Kom., M.T.I**

2. Dekan Fakultas Ilmu Komputer

**Sriyanto, S.Kom., M.M., Ph.D**  
NIK 00210800

Tanggal Lulus Ujian Skripsi : 8 Maret 2019

## RIWAYAT HIDUP

### I. IDENTITAS

1. Nama : Galuh Wiyenda Muhammad
2. Npm : 1411010018
3. Tempat/Tanggal Lahir : Bandung, 1 Juli 1997
4. Agama : Islam
5. Alamat : Jl. ST. Badarudin GG. Gunako Atas No. 11
6. Suku : Sunda
7. Kewarganegaraan : Indonesia
8. No. Telp/ Hp : 087898665754
9. Email : g.wiyenda@gmail.com

### II. Pendidikan yang pernah ditempuh oleh penulis, antara lain :

1. Sekolah Dasar Negeri (SDN) 2 Rawa Laut Bandar Lampung tamat tahun 2008.
2. Sekolah Menengah Pertama (SMP) 14 Bandung tamat tahun 2011.
3. Sekolah Menengah Kejuruan Negeri (SMKN) 4 Bandar Lampung tamat tahun 2014.
4. Pada tahun 2014 Penulis diterima di IIB Darmajaya Jurusan S-1 Teknik Informatika.

Yang menyatakan  
Bandar Lampung, 8 Maret 2019

Galuh Wiyenda M  
NPM.1411010018

## PERSEMBAHAN

Kupersembahkan karya ini untuk :

1. Allah *Subhanahu Wa Ta'ala* atas limpahan anugrah yang telah diberikan sehingga skripsi ini dapat terselesaikan dengan baik.
2. Kedua Orang Tuaku yang senantiasa mencurahkan kasih sayang untuk keberhasilan dan setiap doanya untuk kehidupanku.
3. Bapak Apri Triansah, S.Kom., M.T.I, selaku pembimbing skripsi yang selalu memberikan bimbingan dan masukan – masukan terbaik sehingga skripsi ini dapat selesai dengan baik.
4. Teman terdekat Teknik Informatika angkatan 2014 serta teman – teman pejuang skripsi DSC yang sama – sama berjuang dari pagi hingga malam, dan yang selalu mendukung baik langsung maupun tidak langsung.
5. Terima kasih banyak kepada semua dosen yang telah membimbingku dan memberikan ilmu pengetahuan, masukan dan dukungan.
6. Almamater tercinta Institut Informatika dan Bisnis (IIB) Darmajaya yang telah memberikanku banyak bekal ilmu pengetahuan.

## HALAMAN MOTTO

**“Semua hal yang kita lakukan tidak akan ada artinya jika kita tidak melakukannya”**

-Woody J.

## ABSTRACT

### JAVA PROGRAMMING LANGUAGE APPLICATION ACCOMPANIED WITH ANDROID-BASED COMPILERS (IIB Darmajaya Intermediate Programming Case Study)

By:

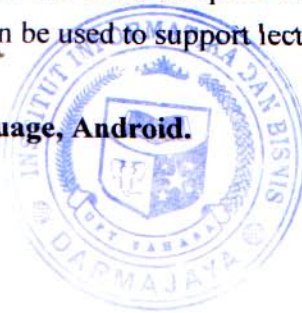
**Galuh Wiyenda Muhammad**

A compiler is a program that translates program languages (source code) into object languages (object code). The compiler combines the entire program language, collects it, and then rearranges it. One program language that uses compilers is Java. In Java language programming learning the students are required to have a personal computer or laptop that has the minimum specifications needed to run it. They are also required to understand the programming concepts within 3 hours of the week.

The java language application accompanied by an android-based compiler is one form of application tool that aims to help students to compile and evaluate the material given by the lecturer by working on quizzes.

The research development of java language applications used the *prototype* method, this method is suitable to be used to develop a software that will be developed again. This method starts with communication with the user. Then, it continued to make a design and modeling to be made the prototype formation before being produced. The result of this study indicates that the Java compiler on Android runs in a good category so that the application can be used to support lectures.

**Keywords: Compile, Java Language, Android.**



8



## **PRAKATA**

Puji syukur dipanjatkan kepada Tuhan YME, yang telah melimpahkan rahmat-Nya sehingga dapat menyelesaikan laporan Skripsi. Laporan ini disusun sebagai laporan tugas akhir. Dalam ini judul yang diangkat ialah “Aplikasi Bahasa Pemrograman Java Disertai Compiler Berbasis Android (Studi Kasus Pemrograman Menengah IIB Darmajaya)”.

Penelitian ini disusun sebagai syarat untuk mencapai gelar sarjana strata satu (S1) pada jurusan Teknik Informatika IIB Darmajaya Bandar Lampung. Dalam penulisan penelitian ini saya banyak memperoleh bimbingan, dorongan dan bantuan dari berbagai pihak. Untuk itu saya mengucapkan terima kasih setulus – tulusnya kepada :

1. Ketua Yayasan Alfian Husein Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
2. Bapak Ir. Firmansyah YA, MBA, M.Sc, Selaku Rektor Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
3. Bapak Sriyanto, S.Kom., M.M, Selaku Dekan Fakultas Ilmu Komputer Institut Informatika dan Bisnis Darmajaya Bandar Lampung.
4. Bapak Yuni Arkhiyansyah, M.Kom, Selaku Ketua Jurusan Teknik Infomatika Institut Informatika dan Bisnis Darmajaya.
5. Bapak Apri Triansah, S.Kom., M.T.I, Selaku Dosen Pembimbing yang telah banyak membimbing, mengarahkan, dan memotivasi penulis dalam penyusunan penelitian ini.
6. Para Dosen, Staf dan karyawan Institut Informatika dan Bisnis Darmajaya Bandar Lampung yang telah memberi bantuan baik langsung maupun tidak langsung selama saya menjadi mahasiswa.
7. Semua pihak yang telah memberikan bantuan dan petunjuk sehingga saya dapat lebih mudah dalam menyusun skripsi ini.
8. Almamater tercinta Institut Informatika dan Bisnis (IIB) Darmajaya yang telah memberikan banyak bekal ilmu pengetahuan.

Saya menyadari bahwa skripsi ini masih jauh dari kesempurnaan, baik dalam pembahasan materi maupun dalam penyajiannya, oleh karena itu kritik dan saran yang sifatnya membangun merupakan masukan yang sangat berarti bagi penyempurnaan dimasa yang akan datang.

Semoga skripsi ini bermanfaat dan dapat dijadikan bahan pertimbangan informasi bagi pihak yang berkepentingan.

Yang menyatakan

Bandar Lampung, 8 Maret 2019

Galuh Wiyenda M

NPM.1411010018

## DAFTAR ISI

	<b>Halaman</b>
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PERNYATAAN</b> .....	<b>ii</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>RIWAYAT HIDUP</b> .....	<b>v</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vi</b>
<b>MOTTO</b> .....	<b>vii</b>
<b>ABSTRAK</b> .....	<b>viii</b>
<b>PRAKATA</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xvi</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	2
1.3 Ruang Lingkup Penelitian.....	2
1.3.1 Tempat Penelitian.....	2
1.3.2 Waktu Penelitian .....	
1.3.3 Batasan Objek Penelitian .....	2
1.4 Tujuan Dan Manfaat .....	3
1.4.1 Tujuan .....	3
1.4.2 Manfaat .....	3
1.5 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Object Oriented Programming .....	5
2.2 Java.....	6
2.2.1 Pengertian Java.....	6
2.2.2 Arsitektur Java.....	6

2.3 Kompilasi .....	7
2.4 Android.....	7
2.5 Android Studio .....	8
2.6 Elemen Android .....	9
2.7 Metode Prototype .....	10
2.8 Unified Modeling Language (UML) .....	12
2.8.1 Use case Diagram .....	12
2.8.2 Activity Diagram .....	12
2.8.3 Sequence Diagram .....	14
2.8.4 Class Diagram .....	14
2.9 Database .....	14
2.9.1 Sqlite.....	14
2.10 Black Box Testing.....	15
2.11 Penelitian Terkait .....	15

### **BAB III METODOLOGI PENELITIAN**

3.1 Metode Pengumpulan Data .....	17
3.1.1 Observasi .....	17
3.1.2 Wawancara.....	17
3.1.3 Studio Literatur .....	17
3.2 Metode Pengembangan Perangkat Lunak .....	17
3.2.1 Komunikasi .....	17
3.2.2 Perencanaan Cepat .....	18
3.2.2.1 Analisa Kebutuhan <i>Non Fungsional</i> .....	18
a. Analsis Kebutuhan Pengguna ( <i>User</i> ) .....	18
b. Analisis Kebutuhan Perangkat Lunak ( <i>Software</i> ) .....	18
c. Analisis Kebutuhan Perangkat Keras ( <i>Hardware</i> ) .....	19
3.2.3 Pemodelan Desain .....	19
3.2.3.1 Desain UML .....	19
a. Use Case Diagram .....	20
b. Rancangan <i>Activity Diagram</i> Menu Utama .....	20
c. Rancangan <i>Activity Diagram</i> Materi .....	21

d. Rancangan <i>Activity Diagram</i> Quiz.....	21
e. Rancangan <i>Activity Diagram</i> Compiler.....	21
f. Rancangan <i>Activity Diagram</i> Tentang Aplikasi .....	22
3.2.3.2 <i>Class Diagram</i> .....	23
3.2.3.3 Desain Antar Muka Aplikasi .....	23
a. Rancangan Tampilan Halaman Splash Screen .....	23
b. Rancangan Tampilan Halaman Menu Utama .....	24
c. Rancangan Tampilan Halaman Materi.....	24
d. Rancangan Tampilan Halaman Isi Materi.....	24
e. Rancangan Tampilan Halaman Compiler .....	25
f. Rancangan Tampilan Halaman Quiz .....	25
g. Rancangan Tampilan Halaman Tentang Aplikasi.....	26
3.2.4 Pembentukan <i>Prototype</i> .....	27
3.2.5 Penyerahan Sistem dan Umpan Balik .....	28

## **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

4.1 Hasil Penelitian .....	29
4.1.1 Implementasi Basis Data .....	29
4.1.2 Implementasi Tampilan Aplikasi.....	29
4.1.2.1 Tampilan Halaman Splash Screen .....	29
4.1.2.2 Tampilan Halaman Menu Utama.....	30
4.1.2.3 Tampilan Halaman Compiler .....	31
4.1.2.4 Tampilan Halaman Materi.....	31
4.2.2.5 Tampilan Halaman Isi Materi.....	32
4.2.2.6 Tampilan Halaman Quiz .....	32
4.2.2.7 Tampilan Halaman Tentang Aplikasi .....	33
4.2 Pengujian Software .....	33
4.2.1 Hasil Pengujian Pada Menu Utama .....	34
4.2.2 Hasil Pengujian Halaman Compiler.....	35
4.2.3 Hasil Pengujian Halaman Materi .....	36
4.2.4 Hasil Pengujian Halaman Quiz .....	37
4.2.5 Hasil Pengujian Halaman Tentang Aplikasi .....	38

4.3 Kelebihan dan Kelemahan Aplikasi.....	39
4.3.1 Kelebihan Aplikasi .....	39
4.3.2 Kelemahan Aplikasi .....	39

## **BAB V SIMPULAN DAN SARAN**

5.1 Simpulan.....	41
5.2 Saran .....	41

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## Daftar Tabel

	<b>Halaman</b>
Tabel 2.1 Simbol <i>Use Case</i> .....	12
Tabel 2.2 Simbol <i>Activity Diagram</i> .....	13
Tabel 2.3 Penelitian Terkait .....	15
Tabel 4.1 Hasil Pengujian Menu Utama .....	34
Tabel 4.2 Hasil Pengujian Halaman <i>Compiler</i> .....	35
Tabel 4.3 Hasil Pengujian Halaman Materi .....	36
Tabel 4.4 Hasil Pengujian Halaman Quiz .....	37
Tabel 4.5 Hasil Pengujian Tentang Aplikasi .....	39

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2.1 Metode <i>Prototype</i> .....	11
Gambar 3.1 Use Case Diagram.....	20
Gambar 3.2 Rancangan <i>Activity Diagram</i> Menu Utama .....	20
Gambar 3.3 Rancangan <i>Activity Diagram</i> Materi.....	21
Gambar 3.4 Rancangan <i>Activity Diagram</i> Quiz.....	21
Gambar 3.5 Rancangan <i>Activity Diagram</i> Compiler .....	22
Gambar 3.6 Rancangan <i>Activity Diagram</i> Tentang Aplikasi.....	22
Gambar 3.7 Class Diagram Aplikasi Bahasa Pemrograman Java .....	23
Gambar 3.8 Rancangan Tampilan Splash Screen .....	23
Gambar 3.9 Rancangan Tampilan Halaman Menu Utama .....	24
Gambar 3.10 Rancangan Tampilan Halaman Materi.....	24
Gambar 3.11 Rancangan Tampilan Halaman Isi Materi.....	25
Gambar 3.12 Rancangan Tampilan Halaman Compiler .....	25
Gambar 3.13 Rancangan Tampilan Halaman Quiz.....	26
Gambar 3.14 Rancangan Tampilan Tentang Aplikasi .....	26
Gambar 4.1 Implementasi Basis Data .....	29
Gambar 4.2 Tampilan Halaman Splash Screen .....	39
Gambar 4.3 Tampilan Halaman Menu Utama .....	40
Gambar 4.4 Tampilan Halaman Compiler .....	40
Gambar 4.5 Tampilan Halaman Materi .....	41
Gambar 4.6 Tampilan Halaman Isi Materi .....	41
Gambar 4.7 Tampilan Halaman Quiz .....	42
Gambar 4.8 Tampilan Halaman Tentang Aplikasi .....	43



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Belajar merupakan kegiatan yang berproses dan merupakan unsur yang sangat fundamental dalam penyelenggaraan setiap jenis jenjang pendidikan. Dalam dunia pendidikan, tercapainya suatu tujuan dari pendidikan kepada mahasiswa sangat ditentukan oleh seorang dosen sebagai pusat pembelajarannya. Dosen berperan dalam membimbing dan melaksanakan pembelajaran agar pembelajaran dapat berlangsung efektif.

Dalam pembelajaran mata kuliah pemrograman menengah yang mempelajari bahasa pemrograman java yang dilaksanakan di Program Studi Teknik Informatika Institut Informasi dan Bisnis Darmajaya, mahasiswa dituntut untuk memahami dan mempraktekkan konsep – konsep pemrograman yang diajarkan dengan waktu 4 sks, yang berarti mahasiswa hanya mendapat total waktu 3 jam selama seminggu. Bahasa pemrograman java memerlukan sebuah IDE (*Integrated Development Environment*) untuk mempraktekkan bahasa sumber java. Dengan terbatasnya waktu untuk memahami konsep – konsep pembelajaran, serta terbatasnya perangkat komputer yang memiliki IDE (*Integrated Development Environment*) dengan spesifikasi yang memadai, membuat mahasiswa tidak memahami semua materi yang telah diberikan.

Indikasi ini dimungkinkan karena beberapa faktor, pertama dalam menyampaikan materi sebagian dosen hanya menggunakan salah satu buku untuk memberikan materi, kedua tidak semua mahasiswa mampu memahami materi dengan hanya membaca buku, diperlukan pembahasan ulang materi yang ada pada buku, ketiga diperlukannya perangkat komputer dengan spesifikasi yang memadai untuk mempraktekkan bahasa sumber java. Dalam penelitian yang dilakukan oleh Bella Hardiyana, S.Kom dan Yayang Nopandi, S.Kom yang berjudul “Rancang Bangun Aplikasi Pembelajaran Pemrograman Berorientasi Objek Dengan Bahasa

Java” disebutkan bahwa mahasiswa dituntut untuk memahami konsep – konsep pemrograman yang diajarkan dengan jatah waktu selama 1,5 jam dalam seminggu. Dari faktor – faktor inilah yang menyebabkan belajar mahasiswa kurang efektif, sehingga menyebabkan mahasiswa kesulitan dalam mempelajari bahasa pemrograman java dan membuat mata kuliah ini sulit dimengerti.

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Dengan memanfaatkan teknologi telepon pintar, inovasi dalam pembelajaran dan strategi pembelajaran perlu dilakukan agar proses belajar mengajar lebih menarik serta mudah diterima. Upaya untuk menambah pemahaman mahasiswa dalam belajar dapat dilakukan dengan memberikan materi – materi dan soal untuk mengasah kemampuan mahasiswa melalui media pembelajaran.

Berdasarkan hal tersebut maka perlu dibangun sebuah aplikasi yang berjudul **“APLIKASI BAHASA PEMROGRAMAN JAVA DISERTAI COMPILER BERBASIS ANDROID”**.

Aplikasi ini memiliki kelebihan yang dapat membantu mahasiswa mempelajari bahasa pemrograman java dan mencoba *source code* dengan android tanpa harus menggunakan perangkat komputer.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah, maka perumusan masalah dalam penelitian ini adalah bagaimana membangun dan merancang sebuah aplikasi bahasa pemrograman java yang mampu melakukan *compiling* secara langsung berbasis android?

## **1.3 Ruang Lingkup Penelitian**

### **1.3.1 Tempat Penelitian**

Tempat penelitian dilaksanakan di Institut Informatika dan Bisnis Darmajaya, yang beralamat di Jalan Z.A. Pagar Alam, No.93 Labuhan Ratu, Bandar Lampung.

### **1.3.2 Waktu Penelitian**

Waktu penelitian dilaksanakan selama 90 hari.

### **1.3.3 Batasan Objek Penelitian**

Agar penelitian tidak keluar dari masalah yang akan dibahas, maka batasan obek permasalahan yaitu:

1. Pembuatan aplikasi hanya digunakan untuk bahasa pemrograman java.
2. Aplikasi yang dihasilkan berbasis *mobile* dengan sistem operasi Android.
3. Aplikasi yang dihasilkan dapat menjalankan bahasa pemrograman java berbasis *command line interface*.

## **1.4 Tujuan Dan Manfaat**

### **1.4.1 Tujuan**

Tujuan dari penelitian ini antara lain:

1. Membuat aplikasi pengenalan pemrograman java berbasis android.
2. Membuat aplikasi dengan *compiler* berbasis android.
3. Diharapkan mahasiswa semakin tertarik dalam memahami bahasa pemrograman java sehingga hasil belajar lebih baik.

### **1.4.2 Manfaat**

Manfaat yang didapat dari penelitian ini antara lain:

1. Membantu mahasiswa untuk menerapkan source code yang dipelajari dengan menggunakan android.
2. Membantu mahasiswa dalam memahami bahasa pemrograman java.
3. Membantu pengguna umum yang ingin belajar bahasa pemrograman java untuk menerapkan dan memahami bahasa pemrograman java.

## 1.5 Sistematika Penulisan

Uraian singkat mengenai sistematika penulisan pada masing-masing bab adalah sebagai berikut :

### **BAB I : PENDAHULUAN**

Pada bab ini berisi tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penulisan.

### **BAB II : LANDASAN TEORI**

Pada bab ini berisi tentang teori-teori yang mendukung penelitian yang akan dilakukan oleh penulis/peneliti.

### **BAB III : METODOLOGI PENELITIAN**

Bab ini berisi metode – metode penelitian yang digunakan serta kebutuhan sistem aplikasi yang dibuat untuk menentukan konsentrasi skripsi dan rekomendasi bahasa pemrograman baik berupa analisis kebutuhan data, desain dan rancangan konteks diagram, rancangan *database*, dan rancangan tampilan.

### **BAB IV : HASIL PENELITIAN DAN PEMBAHASAN**

Bab ini menjelaskan tentang hasil analisis dan pembahasan yang diperoleh berkaitan dengan landasan teori yang relevan dan memberikan gambaran tentang desain aplikasi yang baru.

### **BAB V : SIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan sistem informasi geografis *pet shop* berbasis android selanjutnya.

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## BAB II

### TINJAUAN PUSTAKA

#### **2.1 *Object Oriented Programming***

*Object Oriented Programming* adalah paradigma pemrograman yang memandang perangkat lunak sebagai kumpulan objek yang saling berinteraksi di dalam suatu sistem. (Azis, 2005) Beberapa objek berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing objek harus berisikan informasi mengenai dirinya sendiri (*encapsulation*) dan objek yang dapat dikaitkan (*inheritance*). (Febrian, 2004)

Dalam OOP, *Class* merupakan sekumpulan objek yang memiliki atribut-atribut dan *method*. *Class* merupakan deskripsi dari satu atau lebih objek yang memiliki kesamaan atribut, layanan, metode, hubungan, dan semantik, termasuk deskripsi cara membuat objek baru dalam *class*. Ada juga yang disebut dengan *super class*, sebuah *class* induk yang nantinya mempunyai *class-class* yang terdiri dari *class* dan *subclass*.

Objek dalam OOP adalah sebuah benda atau unit atau sifat kerja yang memiliki atribut-atribut. Objek adalah sebuah abstraksi dari sesuatu pada domain masalah, menggambarkan kemampuan untuk menyimpan informasi mengenai hal tersebut, berinteraksi dengan hal tersebut atau keduanya.

Abstraksi prosedural dalam OOP disebut dengan operasi, yang menspesifikasi tipe dari perilaku dan terdiri dari fungsi-fungsi.

Istilah lain terdapat *encapsulation*/pengkapsulan, yang merupakan pembatasan ruang lingkup program terhadap data yang diproses supaya data terlindungi oleh prosedur atau objek lain, kecuali prosedur yang berada di objek itu sendiri.

*Polymorphism* adalah konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku yang berbeda, bahwa operasi yang sama mungkin memiliki perbedaan dalam *class* yang berbeda.

Pada OOP, terdapat juga yang disebut dengan *inheritance* (pewarisan), yaitu kepemilikan yang bersifat *implicit* dari fitur *subclass* yang didefinisikan dalam *superclass*. Fitur tersebut mencakup *variables* dan *method*.

## 2.2 Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun pada tahun 1995 diganti namanya menjadi “Java”.

### 2.2.1 Pengertian Java

Menurut (Rosa & Shalahuddin, 2010) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*standalone*) ataupun pada lingkungan jaringan.

Java berdiri di atas sebuah mesin penterjemah (*interpreter*) yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca kode bit (*bytecode*) dalam file *.class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (*platform independent*). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah “*write once, run everywhere*”, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (*platform*) manapun, tanpa harus melakukan perubahan kode program.

### 2.2.2 Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun sejak awal mula bahasa tersebut dirilis. *Compiler Java* (yang disebut dengan *javac* atau *Java Compiler*) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu kode bit. Dimana *bytecode* adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (*Java Virtual Machine*). JVM juga sering dinamakan sebagai *interpreter*, karena sifatnya yang selalu menerjemahkan

kode-kode yang tersimpan dalam kode bit dengan cara baris demi baris. Untuk menjalankan program Java, maka file dengan ekstensi *.java* harus dikompilasi menjadi file kode bit. Dimana untuk menjalankan kode bit tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan pustaka Java yang digunakan.

### 2.3 Kompilasi

Kompilasi merupakan teknik dalam melakukan pembacaan suatu program yang ditulis dalam bahasa sumber, kemudian diterjemahkan ke dalam suatu bahasa lain yang disebut bahasa sasaran. Dalam melakukan proses penerjemahan tersebut, kompilator akan melaporkan adanya keanehan-keanehan atau kesalahan yang mungkin ditemukannya. Proses penerjemahan yang dilakukan oleh kompilator ini disebut proses kompilasi (*compiling*).

Istilah kompilator biasa digunakan untuk program komputer yang menerjemahkan program yang ditulis dalam bahasa tingkat-tinggi (semacam bahasa Pascal, C++, BASIC, FORTRAN, Visual C#, Java) menjadi bahasa mesin.

### 2.4 Android

Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai 'jembatan' antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. (Sari & Komputer, 2016).

Android merupakan sistem operasi untuk perangkat *mobile* yang berbasis Linux dan bersifat terbuka atau *open source* dengan lisensi GNU yang dimiliki Google. (Wahana, 2013).

Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras,

perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Adapun versi – versi android yang pernah dirilis adalah sebagai berikut:

1. Android Versi 1.0 Alpha - Dirilis pada 23 September 2008.
2. Android Versi 1.1 Bender (Beta) - Dirilis pada 9 Februari 2009.
3. Android Versi 1.5 CupCake - Dirilis pada 27 April 2009.
4. Android Versi 1.6 Donut - Dirilis pada 15 September 2009.
5. Android Versi 2.0 - 2.1 Eclair - Dirilis pada 26 Oktober 2009.
6. Android Versi 2.2 Frozen Yoghurt - Dirilis pada 10 Mei 2010.
7. Android Versi 2.3 GingerBread - Dirilis pada 6 Desember 2010.
8. Android Versi 3.0-3.2 HoneyComb - Dirilis pada 22 Pebruari 2011.
9. Android Versi 4.0 Ice Cream Sandwich - Dirilis pada 19 Oktober 2011.
10. Android Versi 4.1 - 4.3 Jeally Bean - Dirilis pada 27 Juni 2012.
11. Android Versi 4.4 KitKat - Dirilis pada 31 Oktober 2013.
12. Android Versi 5.0 Lollipop - Dirilis pada 5 Juni 2014.
13. Android Versi 6.0 Marshmallow - Dirilis pada 17 Agustus 2015.
14. Android Versi 7.0 Nougat - Dirilis pada 18 Juli 2016.
15. Android Versi 8.0 Oreo - Dirilis pada 21 Agustus 2017.

## 2.5 Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang dikenalkan pihak Google pada acara Google I/O di tahun 2013. Android Studio merupakan suatu pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

Sebagai media yang menjadi pengembangan dari Eclipse, Android Studio dilengkapi dengan gaya baru serta mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai *build environment*. Fitur-fitur lainnya yang sudah disematkan pada Android Studio adalah sebagai berikut:

1. Menggunakan Gradle-based *build system* yang fleksibel.



2. *Template support* untuk Google Services dan berbagai macam tipe perangkat.
3. Layout editor yang lebih bagus.
4. *Built-in support* untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
5. Import library langsung dari Maven Repository.

## 2.6 Elemen Android

### 1. Dalvik Virtual Machine (DVM)

Salah satu element kunci dari Android adalah Dalvik Virtual Machine (DVM). Android berjalan di Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi Android menggunakan virtual machine sendiri yang dirancang untuk memastikan beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

### 2. Android Runtime (ART)

ART adalah singkatan untuk Android Runtime. Mesin virtual baru Android yang menggantikan Dalvik. ART memungkinkan aplikasi yang sama untuk dijalankan pada hardware yang sangat berbeda dengan bertindak sebagai perantara.

ART ini memiliki keunggulan dengan menggunakan metode "*Ahead Of Time*" (AOT) yang melakukan kompilasi data aplikasi telah dilakukan sebelum aplikasi dijalankan, dan mengubahnya menjadi aplikasi native di Android. Berkat adanya teknologi ART ini akan memungkinkan Android tidak memerlukan bantuan dari sebuah mesin virtual atau juga menerjemahkan ulang setiap kode pemrograman sehingga mampu mempersingkat proses startup jauh lebih cepat dimana.

Sedangkan Android yang menggunakan Dalvik terkadang masih terasa kurang performa dan responnya berat karena memang menggunakan metode "*Just In Time*" (JIT) yang akan mengkompilasi data aplikasi

setiap kali aplikasi dijalankan. Tentunya ini akan cukup memakan waktu untuk aplikasi itu sendiri.

### 3. Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi middleware dan aplikasi kunci yang di-release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

### 4. ADT (Android Development Tools)

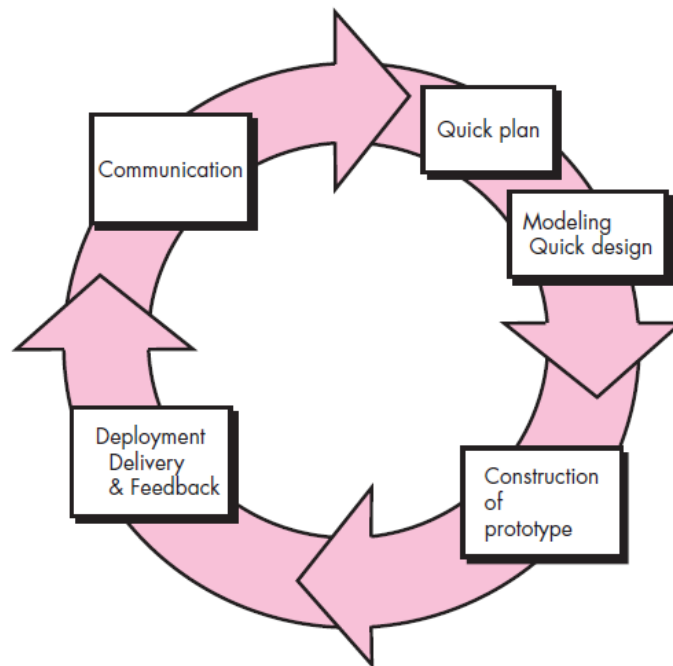
Android development tools adalah plugin yang didesain untuk IDE Android Studio yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan dalam membuat aplikasi project Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

## 2.7 Metode Prototype

(Pressman, 2012) menguraikan bahwa dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar. *Prototype* bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik.

Berikut adalah tahapan dalam metode *prototype* :

1. Komunikasi(*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan user untuk menentukan kebutuhan.
2. Perencanaan cepat(*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat(*Modeling Quick Design*), yaitu membuat rancangan desain program.
4. Pembentukan prototype(*Construction of prototype*), yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. Penyerahan sistem dan umpan balik(*Development Delivery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Metode *Prototype*

## 2.8 Unified Modelling Language (UML)

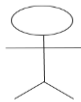
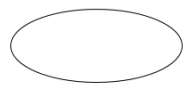

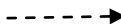
*Unified Modeling Language (UML)* adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik

pemrograman berorientasi objek (Winda & Umi, 2016). Pemrodelan adalah proses merancang peranti lunak sebelum melakukan pengodean, model peranti lunak dapat dianalogikan sebagai *blueprint* pada pembangunan gedung. Diagram - diagram yang digunakan pada UML antara lain adalah *class diagram*, *object diagram*, *use case diagram*, *activity diagram*, dan *sequence diagram*.

### 2.8.1 Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, sebuah use case mepresentasikan sebuah interaksi antara aktor dan sistem yang akan dibuat. Dalam *use case diagram* terdapat istilah seperti aktor, *use case* dan *case relationship*.




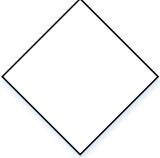
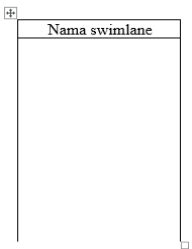
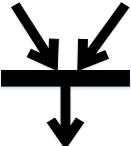
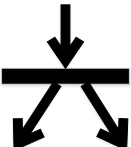
**Tabel 2.1 Simbol Use Case**

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<i>Association</i> : adalah relasi antara aktor dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.

### 2.8.2 Activity Diagram

*Activity diagram* menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. Activity Diagram berupa *flowchart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.2 Simbol Activity Diagram

Simbol	Keterangan
	Activity : Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	Initial Node : Bagaimana objek dibentuk atau diawali
	Activity Final Node : Bagaimana objek dibentuk dan diakhiri.
	Decision : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	Swimlane : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	Join : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	Fork : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

### 2.8.3 Sequence Diagram

*Sequence diagram* menggambarkan interaksi objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Dalam *sequence diagram* terdapat 2 simbol yaitu :

1. Actor, untuk menggambarkan pengguna sistem.
2. Lifeline, untuk menggambarkan kelas dan objek.

### 2.8.4 Class Diagram

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan di buat untuk membangun sistem. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Class memiliki apa yang disebut atribut dan metode atau operasi.

## 2.9 Database

Kadir (Kadir, 2009) Menguraikan Sebuah Basis Data adalah sebuah struktur yang umumnya dikategorikan dalam hal Sebuah Basis Data flat dan sebuah Basis Data relasional. Basis data relasional lebih disukai karena lebih masuk akal dibandingkan Basis data flat Ada tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan.

### 2.9.1 Sqlite

(Safaat, 2012) SQLite adalah salah satu software yang embedded yang sangat populer, kombinasi SQL interface dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di android termasuk dalam Android runtime, sehingga setiap versi dari android dapat membuat database dengan SQLite.

Dalam sistem android terdapat beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

1. Shared prefences yaitu menyimpan data beberapa nilai (value) dalam bentuk groups key yang dikenal dengan prefences.
2. Files yaitu menyimpan data dalam file, dapat berupa menulis ke file atau membaca dari file.

3. SQLite Database, yaitu menyimpan data dalam bentuk database.
4. Content Providers, yaitu menyimpan data dalam bentuk content providers service.

### 2.10 Black Box Testing

(Pressman, 2012) Menguraikan Black Box Testing atau Pengujian Kotak Hitam atau juga disebut Behavioral Testing, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik Black-Box Testing memungkinkan untuk mendapatkan set kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program.

Black-Box Testing bukan merupakan alternatif dari pengujian White Box Testing. Sebaliknya, Black-Box Testing adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode White Box Testing.

Black Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut.

1. Fungsi tidak benar atau hilang.
2. Kesalahan interface atau antarmuka.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja atau perilaku.
5. Kesalahan inisialisasi dan terminasi.

### 2.11 Penelitian Terkait

Berikut adalah penelitian terkait terdahulu yang berkaitan dengan penelitian yang akan dilakukan.

Tabel 2.3 Penelitian Terkait

No	Judul	Nama	Tahun	Keterangan
1	Rancang Bangun Aplikasi Pembelajaran Pemrograman Berorientasi Objek Dengan Bahasa Java	Bella Hardiyana dan Yayang Nopandi	2014	Penelitian ini membahas mengenai aplikasi pembelajaran bahasa pemrograman java untuk membantu mahasiswa dalam meningkatkan kualitas belajar dan memahami bahasa pemrograman java.

Tabel 2.3 Penelitian Terkait (Lanjutan)

No	Judul	Nama	Tahun	Keterangan
2	Membangun Kamus Obstetri Berbasis Android (Studi Kasus : Kampus Kebidanan Adila Bandar Lampung)	Yuni Puspita Sari	2016	Penelitian ini membahas mengenai pemanfaatan teknologi smartphone yang digunakan untuk membangun kamus obstetri.
3	Implementasi Media Pembelajaran Teknik Pengkodean Barcode Berbasis Multimedia Dalam Meningkatkan Kualitas Kegiatan Belajar Mengajar	Septilia Arfida, Rahman E. Harahap	2014	Penelitian ini membahas mengenai aplikasi media pembelajaran teknik pengkodean barcode berbasis multimedia untuk membantu guru dalam meningkatkan kualitas kegiatan belajar mengajar sehingga menjadi lebih menarik bagi para pelajar.



## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

Dalam menyusun penelitian ini, diperlukan data – data yang lengkap sebagai bahan yang mendukung kebenaran materi pembahasan sehingga dilakukan pengumpulan data untuk mendapatkan informasi atau materi yang diperlukan. Adapun metode pengumpulan data pada penelitian ini dapat dijabarkan sebagai berikut :

##### **3.1.1 Observasi**

Metode pengumpulan data ini dilakukan dengan melakukan pengamatan langsung pada objek penelitian. Objek penelitian yang dipilih oleh peneliti adalah jurusan Teknik Informatika pada Institut Informasi dan Bisnis Darmajaya yang berada di Lampung.

##### **3.1.2 Wawancara**

Metode ini dilakukan dengan bertemu langsung dan melakukan tanya jawab atau wawancara dengan pihak yang berkaitan, yaitu mahasiswa.

##### **3.1.3 Studi Literatur**

Metode ini digunakan untuk mencari sumber – sumber atau data yang mendukung dan diperlukan dalam penelitian yang diperoleh dari buku – buku referensi, jurnal penelitian terkait. Penelitian terkait yang pernah dilakukan sebelumnya terdapat pada tabel 2.3.

#### **3.2 Metode Pengembangan Perangkat Lunak**

Pada tahapan pengumpulan perangkat lunak, penelitian ini dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu metode pengembangan sistem *Prototype*.

### 3.2.1 Komunikasi

Komunikasi harus dilakukan dengan tepat. Data yang objektif serta relevan dengan pokok pembahasan, menjadi indikator keberhasilan suatu penelitian. Komunikasi dilakukan dengan cara mengadakan interaksi dengan klien dan user sehingga kebutuhan perangkat lunak dapat terpenuhi.

### 3.2.2 Perencanaan Cepat

*Quick Plan* (rencana cepat) merupakan tahapan dengan melakukan analisis dan perencanaan setelah kita mendapatkan data-data dari tahapan komunikasi.

#### 3.2.2.1 Analisa Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional adalah sebuah langkah dimana seseorang pembangun perangkat lunak menganalisis sumber daya yang dibutuhkan untuk membangun perangkat lunak yang akan dibuat. Tidak hanya menganalisis untuk siapa saja aplikasi ini akan digunakan, analisa kebutuhan non fungsional juga dilakukan untuk menganalisis perangkat keras dan perangkat lunak yang dibutuhkan, agar aplikasi dapat berjalan dengan baik. Analisis non fungsional yang dilakukan dibagi dalam tiga tahapan, yaitu :

##### a. Analisa Kebutuhan Perangkat Lunak (*User*)

Analisa kebutuhan pengguna yang digunakan untuk memasang aplikasi bahasa pemrograman java dan compiler dengan ketentuan sebagai berikut:

1. Menggunakan Sistem Operasi berbasis Android.
2. Menggunakan Sistem Operasi Android minimum versi 5.1 Lollipop.

##### b. Analisa Kebutuhan Perangkat Lunak (*Software*)

Analisa kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi bahasa pemrograman java disertai compiler dengan ketentuan sebagai berikut:

1. Perangkat lunak sistem operasi pada PC adalah Microsoft Windows 8.1 64 bit.

2. Perangkat lunak sistem operasi pada android minimum adalah Android versi 5.1 (Lollipop).
3. Perangkat lunak yang dibutuhkan untuk pembuatan program adalah Android Studio.
4. Perangkat lunak yang dibutuhkan untuk pembuatan database adalah DB Browser for SQLite.

**c. Analisa Kebutuhan Perangkat Keras (*Hardware*)**

Analisa kebutuhan perangkat keras yang digunakan untuk membangun aplikasi bahasa pemrograman java disertai compiler adalah sebagai berikut :

1. Spesifikasi minimum untuk PC :
  - Processor Intel Core i5-4210U.
  - Ram 4 Gb.
  - Harddisk 500 Gb.
2. Spesifikasi minimum untuk android :
  - Processor Qualcomm Snapdragon 615.
  - Ram 1 Gb.
  - Storage 500Mb.

**3.2.3 Permodelan Desain**

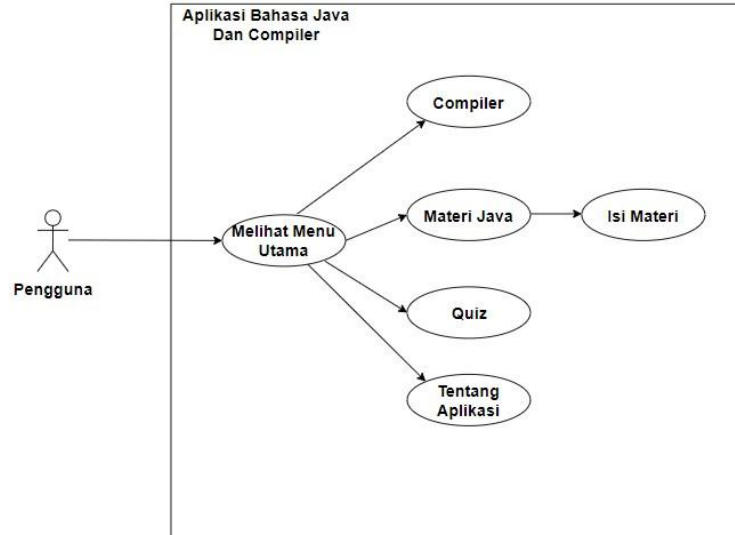
Analisis kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini untuk proses desain sistem yang akan dibuat. Desain Sistem membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

**3.2.3.1 Desain Uml**

UML (*Unified Modelling Language*) yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga untuk dokumentasi sistem software. Pada Penelitian kali ini UML yang dipakai adalah *Use Case*, *Activity Diagram* dan *Class Diagram*.

a. Rancangan *Use Case Diagram*

Rancangan *use case diagram* terdapat pada gambar 3.1.



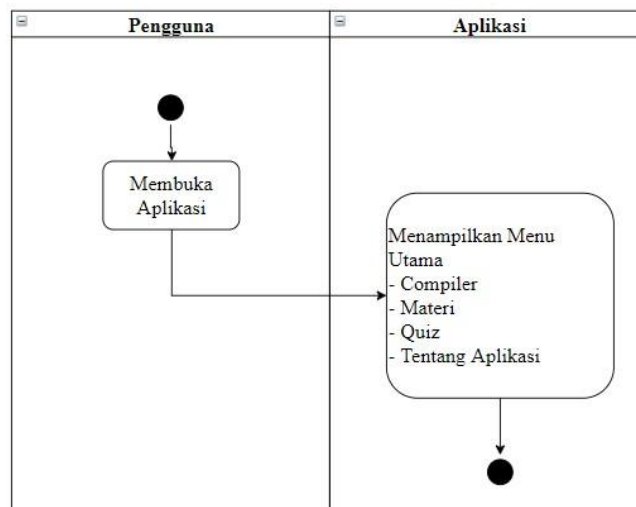
Gambar 3.1 *Use Case Diagram* Sistem

Berdasarkan *use case diagram* pada gambar 3.2 dapat dijelaskan secara singkat masing – masing fungsi dari use case sebagai berikut :

1. *Use case* menu utama, *use case* yang berisi tentang halaman daftar materi.
2. *Use case* menu navigasi, *use case* yang berisi tentang halaman *compiler*, kuis, tentang aplikasi.

b. Rancangan *Activity Diagram* Menu Utama

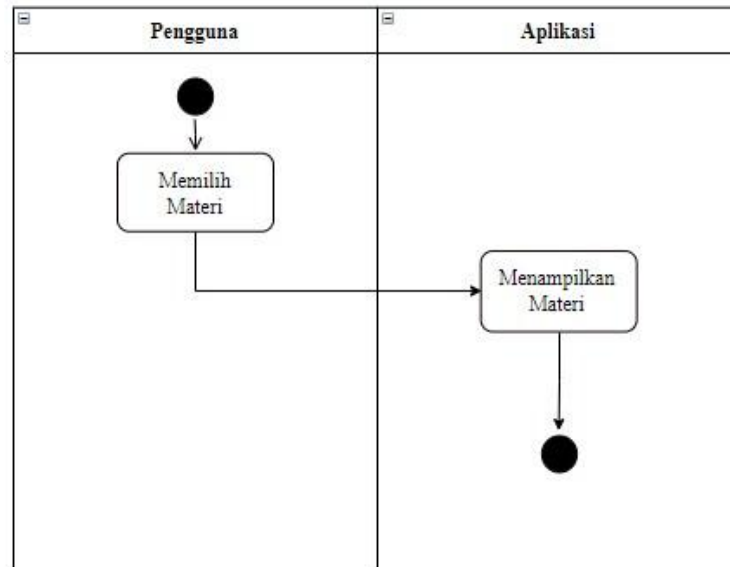
*Activity diagram* menu utama terdapat pada gambar 3.2.



Gambar 3.2 Activity Diagram Menu Utama

## c. Rancangan Activity Diagram Materi

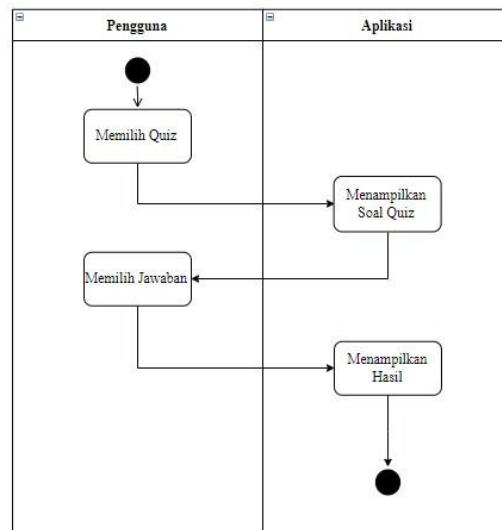
Rancangan activity diagram materi terdapat pada gambar 3.3.



Gambar 3.3 Activity Diagram Materi

## d. Rancangan Activity Diagram Quiz

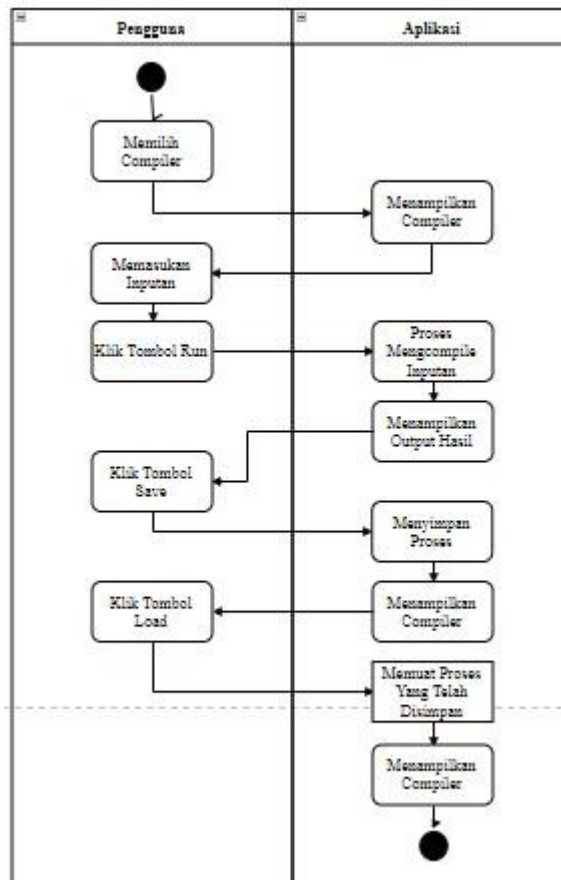
Rancangan activity diagram Quiz terdapat pada gambar 3.4.



Gambar 3.4 Activity Diagram Quiz

## e. Rancangan Activity Diagram Compiler

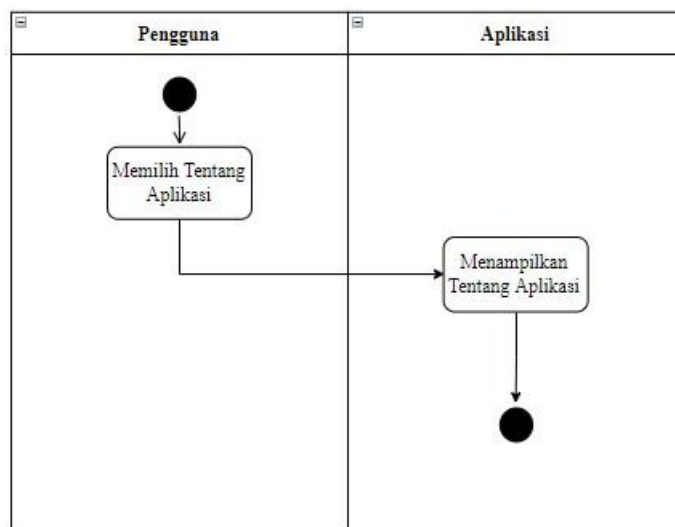
Rancangan activity diagram quiz terdapat pada gambar 3.5.



Gambar 3.5 Activity Diagram Compiler

f. Rancangan Activity Diagram Tentang Aplikasi

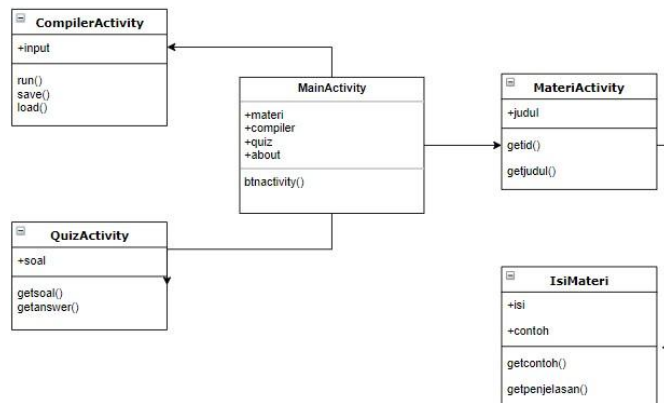
Rancangan activity diagram tentang aplikasi terdapat pada gambar 3.6.



Gambar 3.6 Activity Diagram Tentang Aplikasi

3.2.3.2 Class Diagram

Class diagram menggambarkan struktur dan deskripsi class, package serta objek yang berhubungan satu sama lain. Class diagram aplikasi Bahasa Pemrograman Java Disertai Compiler yaitu sebagai berikut. Tampilan rancangan class diagram terdapat pada gambar 3.7.



Gambar 3.7 Class Diagram Aplikasi Bahasa Pemrograman Java

### 3.2.3.3 Desain Antar Muka Aplikasi

Desain antar muka aplikasi merupakan rancangan bagaimana bentuk dan rupa tampilan aplikasi yang akan dibuat.

#### a. Rancangan Tampilan *Splash Screen*

Tampilan ini adalah tampilan awal yang akan muncul ketika program dibuka. Tampilan rancangan *splash screen* terdapat pada gambar 3.8.



Gambar 3.8 Tampilan Rancangan *Splash Screen*

#### b. Rancangan Tampilan Menu Utama

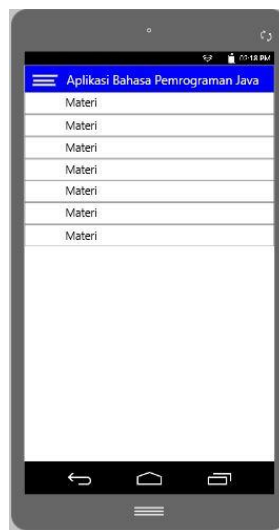
Pada rancangan tampilan menu utama terdapat beberapa *button* yang dapat dipilih, yaitu compiler, materi, quiz dan tentang aplikasi. Tampilan rancangan menu utama terdapat pada gambar 3.9.



Gambar 3.9 Tampilan Rancangan Menu Utama

c. Rancangan Tampilan Halaman Materi

Pada rancangan tampilan halaman materi, terdapat daftar materi yang dapat dipilih. Tampilan rancangan halaman terdapat dilihat pada gambar 3.10.



Gambar 3.10 Tampilan Rancangan Halaman Materi

d. Rancangan Tampilan Halaman Isi Materi



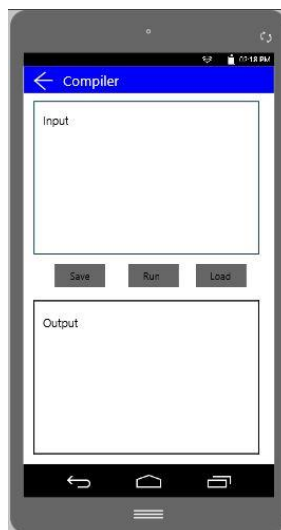
Pada rancangan tampilan halaman isi materi, terdapat penjelasan dan contoh mengenai materi yang dipilih. Tampilan rancangan halaman isi materi terdapat pada gambar 3.11.



Gambar 3.11 Tampilan Rancangan Halaman Isi Materi

e. Rancangan Tampilan Halaman *Compiler*

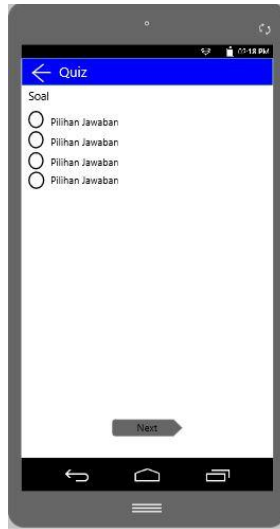
Pada rancangan tampilan halaman *compiler*, terdapat 2 *text area* berupa *input*, dan *output*, serta 3 tombol yaitu, tombol *run*, *save* dan *load*. Tampilan rancangan halaman compiler dapat dilihat pada gambar 3.12.



Gambar 3.12 Tampilan Rancangan Halaman *Compiler*

f. Rancangan Tampilan Halaman *Quiz*

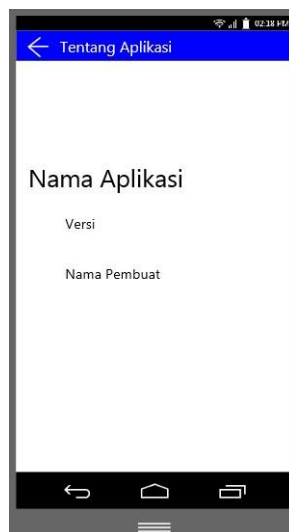
Pada rancangan tampilan halaman *quiz*, terdapat soal dan beberapa pilihan jawaban yang akan dijawab oleh pengguna. Tampilan rancangan halaman *quiz* terdapat pada gambar 3.13.



Gambar 3.13 Tampilan Rancangan Halaman *Quiz*

g. Rancangan Tampilan Halaman Tentang Aplikasi

Pada rancangan tampilan halaman tentang aplikasi, terdapat informasi tentang aplikasi bahasa pemrograman java disertai *compiler*. Tampilan rancangan halaman tentang aplikasi terdapat pada gambar 3.14.



Gambar 3.14. Tampilan Rancangan Halaman Tentang Aplikasi

### 3.2.4 Pembentukan Prototype

Pada tahap inilah aplikasi dibuat menggunakan bahasa pemrograman tertentu dan diuji bagaimana program berjalan.

#### 1) Pembuatan Aplikasi

Dalam penelitian ini aplikasi dibuat pada dasarnya dengan menggunakan Android Studio dan juga perangkat lunak penunjang yang telah dijelaskan pada sub bab analisis kebutuhan perangkat lunak. Sedangkan untuk bahasa pemrograman yang digunakan adalah Java.

#### 2) Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan menggunakan metode black-box testing. Pengujian yang akan dilakukan yaitu dengan menguji fungsi dari masing – masing halaman yang terdapat pada aplikasi. Pengujian dilakukan dengan menggunakan tiga buah perangkat yang berbeda yang memiliki spesifikasi dengan kriteria tinggi, sedang, dan rendah dilihat dari segi perangkat kerasnya.

Berikut adalah ketiga perangkat tersebut dan spesifikasinya :

#### 1. Perangkat Kategori Rendah

- a. Nama Perangkat : Alcatel
- b. Versi Sistem Operasi : Android 5.1 (Lollipop)
- c. Prosesor : *Mediatek 1,2 GHz*
- d. Kapasitas RAM : 2 Gb

#### 2. Perangkat Kategori Sedang

- a. Nama Perangkat : Xiaomi Redmi 3S
- b. Versi Sistem Operasi : Android 6.0 (Marshmallow)
- c. Prosesor : *Octa-core 1,4 GHz*
- d. Kapasitas RAM : 2 Gb

#### 3. Perangkat Kategori Tinggi

- a. Nama Perangkat : Xiaomi Redmi 4 Prime
- b. Versi Sistem Operasi : Android 6.0 (Marshmallow)
- c. Prosesor : *Octa-core 2,0 GHz Cortex-A53*
- d. Kapasitas RAM : 3 Gb

### **3.2.5 Penyerahan Sistem Dan Umpan Balik**

Penyerahan sistem dan umpan balik merupakan tahapan ketika aplikasi telah selesai dibuat. Aplikasi yang telah dibuat akan distribusikan melalui *Google Play Store* sehingga semua orang bisa mengunduhnya.

## BAB IV

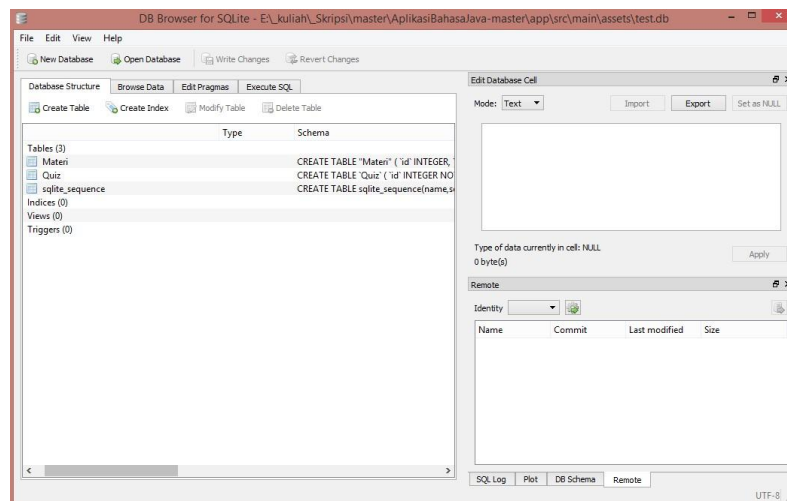
### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Hasil Penelitian

Hasil penelitian lanjutan dari tahap perancangan. Pada tahap ini aplikasi yang telah dirancang akan diimplementasikan ke dalam bentuk *prototype*. Tampilan Aplikasi Bahasa Pemrograman Java Disertai Compiler adalah sebagai berikut.

##### 4.1.1 Implementasi Basis Data

Implementasi basis data yang dibuat menggunakan database sqlite yang merupakan bagian dari android studio. Basis data dibuat berdasarkan Class Diagram yang telah dirancang sebelumnya. Data – data yang dimuat dalam basis data ini berupa materi bahasa pemrograman java, dan contoh soal. Pembuatan basis data ini dilakukan dengan menggunakan tools DB Browser for SQLite yang dijalankan melalui desktop. Implementasi basis data terdapat pada gambar 4.1



Gambar 4.1 Implementasi Basis Data

## 4.1.2 Implementasi Tampilan Aplikasi

### 4.1.2.1 Tampilan Halaman *Splash Screen*

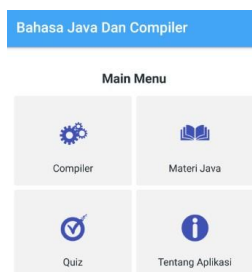
Halaman ini adalah halaman yang akan tampil pertama ketika aplikasi dibuka atau dijalankan. Tampilan halaman *splash screen* terdapat pada gambar 4.2.



Gambar 4.2 Tampilan Halaman *Splash Screen*

### 4.1.2.2 Tampilan Menu Utama

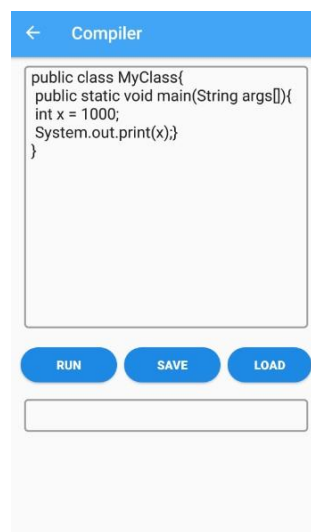
Halaman ini adalah halaman yang akan muncul setelah halaman *splash screen* dimuat. Pada halaman ini terdapat beberapa pilihan menu yang dapat dipilih oleh pengguna diantaranya adalah, Compiler, Materi Java, Quiz, Tentang Aplikasi. Tampilan Menu Utama terdapat pada gambar 4.3.



Gambar 4.3 Tampilan Menu Utama

### 4.1.2.3 Tampilan Halaman *Compiler*

Halaman *compiler* digunakan untuk melakukan uji bahasa pemrograman java dengan cara memasukkan *inputan* lalu melakukan *running*, sehingga akan memberikan hasil berupa output. Pada halaman compiler ini terdapat 3 tombol, yaitu run, save dan load. Tombol *run* berfungsi untuk menjalankan *source code* java yang telah dimasukkan, tombol save berfungsi untuk menyimpan *source code*, tombol load berfungsi untuk memuat *source code* yang telah disimpan. Dan juga terdapat 2 *text area*, yang pertama berupa tempat penulisan *source code* yang disebut *text area input*, dan yang kedua adalah *text area output* yang menampilkan hasil dari proses *compile* dari input. Tampilan Halaman *Compiler* terdapat pada gambar 4.4.



Gambar 4.4 Tampilan Halaman *Compiler*

### 4.1.2.4 Tampilan Halaman Materi

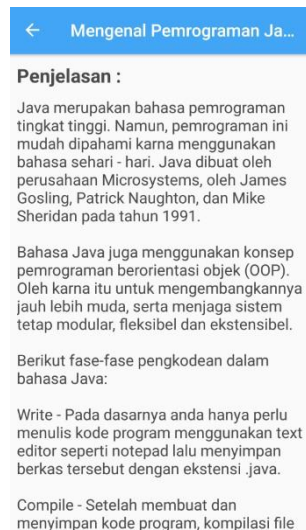
Halaman ini berfungsi untuk menampilkan materi – materi yang terdapat pada Aplikasi Bahasa Pemrograman Java Disertai Compiler. Tampilan halaman materi terdapat pada gambar 4.5.



Gambar 4.5 Tampilan Halaman Materi

#### 4.1.2.5 Tampilan Halaman Isi Materi

Halaman ini berfungsi untuk menampilkan penjelasan dan contoh dari materi yang telah dipilih sebelumnya pada halaman materi. Tampilan halaman isi materi terdapat pada gambar 4.6.



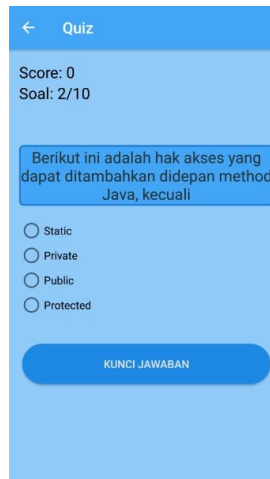
Gambar 4.6 Tampilan Halaman Isi Materi

#### 4.1.2.6 Tampilan Halaman Quiz

Halaman ini digunakan untuk mengevaluasi pemahaman pengguna terhadap bahasa pemrograman java. Proses evaluasi dilakukan dengan cara memilih jawaban dari soal yang telah disediakan, setelah jawaban terkunci akan muncul jawaban yang benar, jika pengguna memilih



jawaban yang benar atau salah, maka papan *score* akan langsung menghitungnya. Tampilan halaman quiz terdapat pada gambar 4.7.



Gambar 4.7 Tampilan Halaman Quiz

#### 4.1.2.7 Tampilan Halaman Tentang Aplikasi

Halaman ini menampilkan informasi tentang pembuat aplikasi, versi aplikasi, serta informasi mengenai aplikasi Bahasa Pemrograman Java Disertai Compiler. Tampilan halaman tentang aplikasi terdapat pada gambar 4.8.



Gambar 4.8 Tampilan Halaman Tentang Aplikasi

## 4.2 Pengujian Software

Pengujian dilakukan untuk mengetahui kualitas dari perangkat lunak yang dirancang ataupun yang sedang dikembangkan, apakah sesuai dengan

perancangan yang sudah dibuat sebelumnya ataupun tidak. Pengujian program merupakan pengujian terhadap program yang dijalankan, sehingga dapat diketahui kesalahan – kesalahan maupun kekurangan dari program itu sendiri. Jika dalam proses pengujian mengalami masalah, error ataupun tidak sesuai dengan proses yang sudah ditentukan, maka dapat dilakukan proses perbaikan rancangan aplikasi. Pengujian dilakukan dengan 3 perangkat dengan spesifikasi yang berbeda, perangkat nomor 1 dengan kriteria rendah, nomor 2 kriteria sedang dan perangkat nomor 3 kriteria tinggi. Berikut adalah hasil pengujian pada ketiga perangkat.

#### 4.2.1 Hasil Pengujian Menu Utama

Hasil pengujian menu utama ditunjukkan pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Menu Utama

<b>No Perangkat</b>	<b>Bagian Yang Diuji</b>	<b>Butir Uji</b>	<b>Hasil Uji</b>
1	Halaman Menu Utama	Pengguna Membuka Aplikasi	Halaman Menu Utama Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
2	Halaman Menu Utama	Pengguna Membuka Aplikasi	Halaman Menu Utama Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Halaman Menu Utama	Pengguna Membuka Aplikasi	Halaman Menu Utama Terbuka, Tampilan Sesuai Seperti Yang Diinginkan

#### 4.2.2 Hasil Pengujian Halaman Compiler

Hasil pengujian halaman compiler ditunjukkan oleh tabel 4.2.

Tabel 4.2 Hasil Pengujian Halaman Compiler

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
1	Halaman Compiler	Pengguna Membuka Halaman Compiler	Halaman Compiler Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
2	Halaman Compiler	Pengguna Membuka Halaman Compiler	Halaman Compiler Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Halaman Compiler	Pengguna Membuka Halaman Compiler	Halaman Compiler Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
1	Tombol Run, Save dan Load	Pengguna Memilih Pilihan Tombol	Tombol Run Compiler Berjalan Dan Menampilkan Output, Tombol Save Dapat Menyimpan Text Input, Tombol Load Dapat Memuat Input Text Yang Telah Disimpan

Tabel 4.2 Hasil Pengujian Halaman Compiler (Lanjutan)

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
2	Halaman Compiler	Pengguna Memilih Pilihan Tombol	Tombol Run Compiler Berjalan Dan Menampilkan Output, Tombol Save Dapat Menyimpan Text Input, Tombol Load Dapat Memuat Input Text Yang Telah Disimpan
3	Halaman Compiler	Pengguna Memilih Pilihan Tombol	Tombol Run Compiler Berjalan Dan Menampilkan Output, Tombol Save Dapat Menyimpan Text Input, Tombol Load Dapat Memuat Input Text Yang Telah Disimpan

#### 4.2.3 Hasil Pengujian Halaman Materi

Hasil pengujian halaman materi ditunjukkan oleh tabel 4.3.

Tabel 4.3 Hasil Pengujian Halaman Materi

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
1	Halaman Materi	Pengguna Membuka Halaman Materi	Halaman Materi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan

Tabel 4.3 Hasil Pengujian Halaman Materi (Lanjutan)

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
2	Halaman Materi	Pengguna Membuka Halaman Materi	Halaman Materi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Halaman Materi	Pengguna Membuka Halaman Materi	Halaman Materi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
1	Daftar Materi	Pengguna Memilih Materi	Materi Yang Dipilih Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
2	Daftar Materi	Pengguna Memilih Materi	Materi Yang Dipilih Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Daftar Materi	Pengguna Memilih Materi	Materi Yang Dipilih Terbuka, Tampilan Sesuai Seperti Yang Diinginkan

#### 4.2.4 Hasil Pengujian Halaman Quiz

Hasil pengujian halaman quiz ditunjukkan pada tabel 4.4.

Tabel 4.4 Hasil Pengujian Halaman Quiz

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
1	Halaman Quiz	Pengguna Membuka Halaman Quiz	Halaman Quiz Terbuka, Tampilan Sesuai Seperti Yang Diinginkan

Tabel 4.4 Hasil Pengujian Halaman Quiz (Lanjutan)

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
2	Halaman Quiz	Pengguna Membuka Halaman Quiz	Halaman Quiz Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Halaman Quiz	Pengguna Membuka Halaman Quiz	Halaman Quiz Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
1	Tombol Pilihan Jawaban, dan Selanjutnya	Pengguna Memilih Jawaban dan Tombol Selanjutnya	Jawaban Terpilih, Tombol Selanjutnya Berfungsi Sesuai Dengan Fungsinya
2	Tombol Pilihan Jawaban, dan Selanjutnya	Pengguna Memilih Jawaban dan Tombol Selanjutnya	Jawaban Terpilih, Tombol Selanjutnya Berfungsi Sesuai Dengan Fungsinya
3	Tombol Pilihan Jawaban, dan Selanjutnya	Pengguna Memilih Jawaban dan Tombol Selanjutnya	Jawaban Terpilih, Tombol Selanjutnya Berfungsi Sesuai Dengan Fungsinya

#### 4.2.5 Hasil Pengujian Halaman Tentang Aplikasi

Hasil pengujian halaman tentang aplikasi ditunjukkan pada tabel 4.5.

Tabel 4.5 Hasil Pengujian Halaman Tentang Aplikasi

No Perangkat	Bagian Yang Diuji	Butir Uji	Hasil Uji
1	Halaman Tentang Aplikasi	Pengguna Membuka Halaman Tentang Aplikasi	Halaman Tentang Aplikasi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
2	Halaman Tentang Aplikasi	Pengguna Membuka Halaman Tentang Aplikasi	Halaman Tentang Aplikasi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan
3	Halaman Tentang Aplikasi	Pengguna Membuka Halaman Tentang Aplikasi	Halaman Tentang Aplikasi Terbuka, Tampilan Sesuai Seperti Yang Diinginkan

### 4.3 Kelebihan Dan Kelemahan Aplikasi

#### 4.3.1 Kelebihan Aplikasi

- a. Terdapat materi – materi dasar mengenai pemahaman bahasa pemrograman java.
- b. Aplikasi memiliki fitur compiler untuk mencoba *source code* yang telah dibuat.
- c. Aplikasi ini mudah dipahami.

#### 4.3.2 Kekurangan Aplikasi

- a. *Source code* java yang dapat dicompile hanya berupa CLI.

- b. Database aplikasi belum bersifat online sehingga ketika menambah materi baru harus diperbarui dan diupload kembali melalui google playstore.
- c. Aplikasi membutuhkan internet saat menjalankan fitur compiler.
- d. Materi aplikasi belum mencakup seluruh bahasa pemrograman java.



## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melakukan semua analisa, perancangan, implementasi, dan evaluasi system maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Dengan adanya aplikasi Bahasa Pemrograman Java Disertai Compiler Berbasis Android dapat membantu pengguna atau mahasiswa untuk memahami dasar – dasar bahasa pemrograman java.
2. Aplikasi ini dapat dijalankan pada android versi 5.1 (*Lollipop*) keatas.
3. Aplikasi ini dapat mencoba *source code* dengan compiler bahasa pemrograman java berbasis android.

#### 5.2 Saran

Peneliti memberikan saran bagi yang akan mengembangkan penelitian ini, yaitu sebagai berikut :

1. Untuk penelitian selanjutnya diharapkan aplikasi dapat diperbaharui secara berkala sehingga menyediakan materi yang baru.
2. Untuk penelitian lebih lanjut diharapkan aplikasi dapat menjalankan fitur *compiler* tanpa membutuhkan internet.

## DAFTAR PUSTAKA

- Arfida, S., & Harahap, R. E. (2014). Implementasi Media Pembelajaran Teknik Pengkodean Barcode Berbasis Multimedia. *Proseding Seminar Bisnis & Teknologi IIB Darmajaya*, 409. Retrieved from <https://jurnal.darmajaya.ac.id/index.php/sembistek/article/view/226/112>.
- Azis, F. (2005). *Object Oriented Programming dengan PHP*. Jakarta: Elex Media. Komputindo.
- Febrian, J. (2004). *Kamus Komputer dan Istilah Teknologi Informasi*. Bandung: Informatika.
- Hardiyana, B., & Nopandi, Y. (2014). Rancang Bangun Aplikasi Pembelajaran Pemrograman Berorientasi Objek Dengan Bahasa Java. *Prodi Sistem Informasi Universitas Komputer Indonesia*.
- Kadir, A. (2009). *Mastering AJAX dan PHP*. Yogyakarta: C.V Andi Offset.
- Pressman, R. S. (2012). *Rekayasa Perangkat Lunak (7th ed.)*. Yogyakarta: C.V Andi Offset.
- Rosa, A. ., & Shalahuddin, M. (2010). *Modul Pembelajaran Pemrograman Berorientasi Objek – Dengan Bahasa Pemrograman C++, PHP, dan Java*. Bandung: Modula.
- Safaat, N. (2012). *Android Pemrograman Aplikasi Mobile Smartphone Dan Tablet Pc Berbasis Android Revisi Kedua*. Bandung: Informatika.
- Sari, Y. P., & Komputer, F. I. (2016). MEMBANGUN KAMUS OBSTETRI BERBASIS ANDROID (Studi Kasus : Kampus Kebidanan Adila Bandar Lampung), *16*(1), 65–73.
- Wahana, K. (2013). *Android Programming*. Semarang: C.V Andi Offset.
- Winda, A., & Umi, M. (2016). Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati. *Jurnal Sains Dan Informatika*, *2*(2013), 21–28. Retrieved from <http://jurnal.politala.ac.id/index.php/JSI/article/view/97>

# **LAMPIRAN**

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

    tools:context=".DashboardAct">

    <EditText

        android:id="@+id/input"

        android:layout_width="match_parent"

        android:layout_height="300dp"

        android:layout_marginStart="18dp"

        android:layout_marginTop="10dp"

        android:layout_marginEnd="18dp"

        android:layout_marginBottom="8dp"

        android:background="@drawable/btn_rect_grey_outline"

        android:ems="10"

        android:focusableInTouchMode="true"

        android:gravity="top|left"

        android:inputType="textMultiLine"
```

```
android:paddingLeft="8dp"  
android:paddingRight="8dp"  
android:scrollbars="vertical"  
android:scrollHorizontally="true"></EditText>
```

<LinearLayout

```
android:id="@+id/ln"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_margin="10dp"  
android:gravity="center"  
android:orientation="horizontal">
```

<Button

```
android:id="@+id/run"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:background="@drawable/btn_rounded_accent"  
android:text="Run"  
android:textColor="@android:color/white"  
android:textStyle="bold" />
```

<Button

```
    android:id="@+id/save"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:background="@drawable/btn_rounded_accent"

    android:text="Save"

    android:textColor="@android:color/white"

    android:textStyle="bold" />
```

<Button

```
    android:id="@+id/load"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:background="@drawable/btn_rounded_accent"

    android:text="Load"

    android:textColor="@android:color/white"

    android:textStyle="bold" />
```

</LinearLayout>

```
<TextView  
  
    android:id="@+id/output"  
  
    android:layout_width="match_parent"  
  
    android:layout_height="wrap_content"  
  
    android:layout_marginStart="18dp"  
  
    android:layout_marginTop="5dp"  
  
    android:layout_marginEnd="18dp"  
  
    android:layout_marginBottom="18dp"  
  
    android:background="@drawable/btn_rect_grey_outline"  
  
    android:ems="10"  
  
    android:gravity="top|left"  
  
    android:padding="8dp"  
  
    android:textStyle="bold"  
  
    android:scrollbars="vertical"  
  
    android:scrollHorizontally="true"  
  
    android:textSize="16sp"></TextView>
```

```
</LinearLayout>
```

```
package id.wiyenda.aplikasibahasajava;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;

public class AboutAct extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_about);

        getSupportActionBar().setTitle("About");

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    }

}
```