

RANCANG BANGUN APLIKASI PEMBELAJARAN DASAR PEMROGRAMAN BERBASIS MOBILE PHONE

Ali Subhan Afrizal

NIDN. 0202048501

Program Studi Teknik Informatika Politeknik Sekayu

Email alsar@alsar.web.id

ABSTRAK

Pembelajaran pemrograman dasar merupakan suatu upaya untuk memfasilitasi, mendorong dan mendukung pengguna dalam belajar pemrograman. Namun banyak pengguna khususnya pemula tidak memiliki pengetahuan dasar tentang pemrograman sehingga mereka menganggap pemrograman merupakan pelajaran yang sulit. Anggapan ini membuat mereka malas untuk belajar pemrograman. Sehingga dibutuhkanlah aplikasi pembelajaran dasar pemrograman berbasis *mobile phone* yang menampilkan materi dan simulasi soal untuk dapat memudahkan dan meningkatkan minat pengguna dalam memahami pemrograman. Aplikasi ini akan menampilkan materi mengenai konsep dasar, percabangan, perulangan, *array* dan terdapat simulasi soal pada setiap materi serta terdapat evaluasi soal seluruh materi yang berjumlah 20 soal dengan waktu 15 menit. Dengan fitur ini diharapkan agar dapat membantu membantu orang - orang dalam pembelajaran mengenai pemrograman.

Kata Kunci : Aplikasi, Dasar Pemrograman, Mobile Phone

1. PENDAHULUAN

1.1 Latar Belakang

Berkembangnya suatu jaman, media dan teknologi memiliki pengaruh penting terhadap pendidikan. Contohnya, *Handphone*, *PC Tablet*, komputer dan internet telah mempengaruhi proses pembelajaran sampai saat ini. Aturan-aturan dari pendidik dan pebelajar telah berubah karena dipengaruhi media dan teknologi yang digunakan di dalam kelas. Banyak media pembelajaran yang dapat digunakan, salah satu media yang digunakan yaitu aplikasi pembelajaran pada *gadget*. Media pembelajaran mampu melatih pengetahuan, keterampilan dan ketepatan dalam sistem pembelajaran dengan cara yang lebih mudah dan cepat.

Aplikasi pembelajaran dasar pemrograman khususnya bahasa pemrograman *Java* merupakan salah satu pendekatan dalam sistem belajar. Saat ini *Java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak

aplikasi ataupun aplikasi berbasis *web* (*multiplatform*), oleh karena itu inovasi dalam pembelajaran dan strategi pembelajaran dasar *Java* perlu dilakukan. Sering kali dengan banyaknya materi yang ada membuat pelajar merasa jenuh dan bosan dalam belajar terutama bagi yang awam dan baru belajar pemrograman *Java*, ditambah lagi dengan penyampaian pengajar yang lebih mengacu pada teori-teori yang banyak dan kompleks.

Java merupakan bahasa yang murni *object oriented* yang merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Seluruh struktur program *java* berada di dalam *class*., tidak ada *function* atau *variable* yang dideklarasikan di luar *class*. *Java* adalah bahasa pemrograman yang mempunyai sintaks hampir sama dengan bahasa C/C++ yang mempelajari *variable*, tipe data, operator, kondisi, perulangan, dan *array* yang harus terlebih dahulu dipahami bagi pemula, meskipun dengan perkembangan teknologi sudah bisa memberi

dampak yang cukup berarti, namun masalah utama dalam proses pembelajaran seperti terbatasnya waktu bagi pelajar ataupun yang sudah bekerja, serta tempat yang menyediakan materi yang memadai menjadi masalah yang utama bagi pelajar dan masyarakat pada umumnya.

Berdasarkan permasalahan di perlukan suatu metode atau media pembelajaran yang penyampaian materi-materi serta contoh soal hendaknya dilakukan dalam situasi yang lebih menarik dan menyenangkan serta mudah diterima yang bersifat kapan saja dan dimana saja, salah satunya media pembelajaran yang bersifat *portable*, oleh karena itu penulis bermaksud untuk membuat **“Aplikasi Pembelajaran Dasar Pemrograman Berbasis *Mobile Phone*”**.

1.2 Rumusan Masalah

Bagaimana membuat suatu aplikasi pembelajaran pengenalan dasar pemrograman yang mampu melatih pengetahuan, keterampilan dan ketepatan dengan cara yang lebih menarik menggunakan bahasa pemrograman *Java* dan aplikasi *Android Studio* dengan sistem berbasis *Mobile Phone*?

1.3 Metodologi Penelitian

1.3.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian adalah sebagai berikut :

- 1) Studi Literatur
Penulis mengumpulkan data dari berbagai buku, jurnal, situs-situs di *internet* dan teori-teori yang diperlukan untuk pembuatan penelitian ini.
- 2) Observasi
Penulis mengumpulkan data melalui pengamatan terhadap berbagai sumber yang telah ada, yang berkaitan dengan

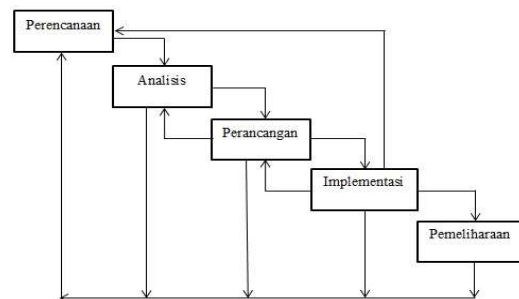
permasalahan yang dibahas dalam pembuatan penelitian ini.

3) Wawancara

Penulis melakukan wawancara kepada narasumber tentang permasalahan yang dibahas dalam pembuatan penelitian ini.

1.3.2 Metode Perancangan Sistem

Metode perancangan aplikasi yang digunakan yaitu *OOP (Object Oriented Programming)*.



Gambar 1.1 Kerangka Kerja Pengembangan Sistem Berorientasi Objek

2. LANDASAN TEORI

2.1 Aplikasi

Berikut merupakan beberapa pengertian mengenai aplikasi oleh para ahli antara lain :

- 1) Menurut Hendrayudi (2009:10) mengatakan bahwa aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus).
- 2) Menurut Rizky Dhanta (2009:16) aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*.

Jadi, dapat disimpulkan bahwa aplikasi adalah sebuah sekumpulan perintah program yang dibuat untuk tujuan tertentu demi menyelesaikan permasalahan tertentu.

2.2 Media Pembelajaran

Menurut Roymond dalam M Rahman (2014:24) mengatakan bahwa media pembelajaran adalah alat yang berfungsi untuk menyampaikan pesan pembelajaran.

Menurut Rosyada dalam Winda Fandella (2015, p.9) media pembelajaran adalah segala sesuatu yang dapat menyampaikan dan menyalurkan pesan dari sumber secara terencana sehingga tercipta lingkungan belajar yang kondusif dimana penerimanya dapat melakukan proses belajar secara efisien dan efektif.

Jadi, dapat disimpulkan bahwa media pembelajaran merupakan alat komunikasi yang digunakan untuk menyalurkan bahan ajar dari pendidik ke peserta didik dimana peserta didik dapat melakukan proses belajar secara efisien dan efektif.

2.3 Bahasa Pemrograman

Menurut Suniarti dalam M Rahman (2014:18) Bahasa pemrograman merupakan untai kata-kata berupa instruksi atau perintah-perintah yang biasanya terdiri dari banyak baris yang bisa dimengerti oleh komputer

Menurut Yulikuspartono dalam Fandella Winda (2015:29) mengemukakan bahwa, “pemrograman merupakan sederetan instruksi atau statement dalam bahasa yang dimengerti oleh komputer yang bersangkutan”.

Jadi berdasarkan pengertian diatas pemrograman merupakan suatu kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu. Perintah-perintah ini membutuhkan suatu bahasa tersendiri yang dapat dimengerti oleh komputer.

2.4 Java

2.4.1 Pengertian Java

Java adalah bahasa pemrograman yang *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan *Java*, anda dapat

menjalankannya hampir di semua komputer dan perangkat lain yang support *Java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya Menurut Heriyanto (2007:3) *Java* adalah bahasa yang dapat dijalankan disembarang *platform*, di beragam lingkungan : *internet*, *consumer electronic products*, dan *computer applications*.

Menurut Abdul Kadir (2011:3) *Java* merupakan pemrograman yang bersifat lintas-*platform*. Artinya, bahasa ini dapat dipakasi untuk menyusun program pada berbagai sistem operasi (misalnya Linux, Windows, dan UNIX).

Jadi berdasarkan pengertian diatas *Java* merupakan pemrograman yang dapat dijalankan *multi platform* (bisa berjalan di berbagai macam sistem operasi) yang mendukung *Java* dan menjalankan perangkat lunak ataupun jaringan.

2.4.2 Dasar-Dasar Pemrograman Java

1) Tipe Data

Tipe Data	Keterangan
Boolean	True atau False
Char	Karakter
Byte	-128 sampai 127
Short	-32768 sampai 32767
Int	-2147483648 sampai 2147483647
Long	-9223372036854775808 sampai 9223372036854775807
Double	4.9E-324 sampai 1.7976931348623157E308
Float	1.4E-45 sampai 3.4028235E38

String bukanlah merupakan tipe data di *Java*, *string* merupakan *Object*. Namun *string* memiliki keunikan yaitu *string* dapat langsung dibuat tanpa harus membuat *object*.

2) Variabel

Variabel merupakan sesuatu yang digunakan untuk menampung sebuah data. Sebuah variabel harus ada dalam sebuah kelas atau metode. Pembuatan sebuah variabel di *Java* terlihat pada kode dibawah ini.

Tipevariabel namavariabel;

Tipe variabel dapat berupa tipe data atau kelas, misal:

```
int nilai;    char indexNilai;
```

Atau dapat juga langsung saat pembuatan sebuah variabel.

Untuk menambahkan nilai ke sebuah variabel, maka dapat menggunakan tanda = (sama dengan), misal jika kita akan menambahkan nilai 100 pada variabel nilai dan A pada variabel indexNilai, maka dapat terlihat pada kode dibawah ini:

```
int nilai = 100;    char indexNilai = 'A';
```

Syarat-syarat penamaan variabel adalah :

- Harus diawali dengan huruf
- Tidak boleh terdapat karakter unik seperti @, #, % dan lain-lain
- Tidak boleh mengandung karakter putih (spasi, enter, tab)

3) Operator

Operator merupakan sebuah karakter khusus yang digunakan untuk menghasilkan suatu nilai.

a) Operator Aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus

Contoh :

```
int a = 10;
int b = 5;
int c = a/b;
System.out.println(c);
```

Hasil dari kode tersebut adalah 2, karena mencetak nilai c yaitu a/b (10/2).

b) Operator Penugasan

Operator	Keterangan
=	Pemberian Nilai
+=	Penjumlahan Bilangan
-=	Pengurangan Bilangan
*=	Perkalian Bilangan
/=	Pembagian Bilangan
%=	Perolehan Sisa Bagi

Contoh :

```
int a = 10;
a += 5;
System.out.println(a);
```

Hasilnya adalah 15. Hal ini dikarenakan a += 5 sama dengan a = a + 5, dikarenakan a sebelumnya adalah 10, maka itu berarti a = 10 + 5

c) Operator Pembandingan

Operator	Keterangan
==	Sama Dengan
!=	Tidak Sama Dengan
>=	Lebih Dari Sama Dengan
<=	Kurang Dari Sama Dengan
>	Lebih Dari
<	Kurang Dari

Hasil dari operasi pembandingan adalah boolean. *True* jika operasi pembandingan tersebut benar, dan *false* jika operasi pembandingan tersebut salah.

Contoh:

```
Boolean a = 10 == 100;
System.out.println(a);
```

Hasil dari program diatas adalah *false*, karena memang 10 tidak sama dengan 100.

d) Operator Logika

Operator	Keterangan
&&	Dan
	Atau

Operator Logika digunakan untuk membentuk suatu keadaan dari dua atau lebih kondisi tertentu, operator logika biasanya digabungkan dengan operator pembandingan. Hasil dari operator logika

adalah boolean. Hasil operasi logika dengan menggunakan && adalah sebagai berikut:

Tabel 2.6 Logika *and*

Operasi 1	Operasi 2	Keterangan
True	True	True
True	False	False
False	True	False
False	False	True

Hasil operasi logika dengan menggunakan || adalah sebagai berikut:

Tabel 2.7 Logika *or*

Operasi 1	Operasi 2	Keterangan
True	True	False
True	False	True
False	True	True
False	False	False

Contoh:

```
Boolean a = 10 == 100 || 100 == 100
System.out.println(a);
```

Jika program di jalankan, maka hasil nya adalah *true*

4) Percabangan

a) Percabangan *If*

Pernyataan *if* merupakan salah satu bentuk pernyataan yang berguna untuk mengambil keputusan terhadap sebuah kemungkinan.

Bentuk pernyataan *if* berupa:

```
If (kondisi) {
    //statement;
}
```

Contoh:

```
Int a = 10;
If (a == 10) {
    System.out.println("Sepuluh");
}
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan "Sepuluh" karena kondisi pada *if* bernilai *true*, jika kondisi bernilai salah,

misal $a == 100$, maka program tidak akan menghasilkan tulisan apa-apa.

b) Percabangan *if-else*

Percabangan *if-else* merupakan percabangan yang sama dengan percabangan *if* namun memiliki kondisi *false*, artinya jika kondisi pada *if* tidak terpenuhi maka perintah pada *else* akan dijalankan.

Bentuk pernyataan *if-else*:

```
If (kondisi) {
    //Jalankan jika benar;
}else{
    //Jalankan jika salah;
}
```

Contoh:

```
Int a = 10;
If (a == 10) {
    System.out.println("Sepuluh");
}else{
    System.out.println("Bukan
    Sepuluh");
}
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan "Bukan Sepuluh", hal ini dikarenakan nilai bernilai 8.

c) *If* Bersarang

Percabangan *if* bersarang merupakan gabungan beberapa *if* dan dapat pula digabung dengan *if-else*. Bentuk pernyataan *if* bersarang adalah sebagai berikut:

```
If (kondisi) {
    //perintah kondisi1;
}else if (kondisi2){
    //perintah kondisi2;
}else{
    //perintah jika semua kondisi salah;
}
```

Contoh:

```
Int a = a;
Char nilai;
If (a >= 8) {
    nilai='A';
} else if (a >= 5) {
    nilai='B';
} else {
    nilai='C';
}
System.out.println(nilai);
//Outputnya adalah C
```

d) Percabangan *Switch*

Percabangan *switch-case* merupakan percabangan yang kondisinya hanya dapat menggunakan perbandingan `==` (sama dengan). Bentuk pernyataan percabangan *switch-case* adalah sebagai berikut:

```
switch(variable){
case nilai1:
    //jalankan intruksi;
    break; //hentikan intruksi
case nilai2:
    //jalankan intruksi;
    break; //hentikan intruksi
default:
    //jalankan intruksi;
    break; //hentikan intruksi
}
```

Contoh:

```
int a = 5;
switch(a){
case 5:
    System.out.println("Salah");
    break;
case 6:
    System.out.println("LIMA");
    break;
default:
    System.out.println("Tidak ada");
    break;
}
```

Jika program diatas dijalankan, maka program akan menghasilkan tulisan "LIMA". Pada percabangan *switch* pertama, diperlukan

sebuah variabel, setelah itu bada bagian *case* dibandingkan, jika sama, maka instruksi akan dijalankan sampai menemui tanda *break*.

e) Perulangan *while*

Pernyataan *while* berguna untuk melakukan proses perulangan untuk sebuah kondisi, selama kondisi tersebut bernilai benar (*true*), maka perulangan akan terus berjalan, dan terhenti ketika kondisi bernilai salah (*false*). Bentuk pernyataan *while* seperti berikut ini :

```
while(kondisi){
    //isi intruksi;
}
```

Contoh:

```
int a=1;
while(a<=10){
    System.out.println(a);
    a++;
    //menaikan nilai a;
}
```

Jika program tersebut dijalankan, maka hasilnya adalah tulisan dari no 1 sampai 10. Dan saat jumlah bernilai 11, maka perulangan akan terhenti dikarenakan kondisi bernilai *false* (11 <= 10).

f) Perulangan *do-while*

Perulangan *do-while* merupakan perulangan yang hampir mirip dengan perulangan *while* namun perbedaannya, pada perulangan *dowhile*, maka minimal instruksi akan dijalankan sekali. Bentuk pernyataan *do-while* sebagai berikut:

```
do{
    //isi intruksi;
} while(kondisi);
```

Contoh:

```
int a=100;
do{
    System.out.println(a);
    a++;
    //menaikkan nilai a;
}while(a<=10);
```

Jika program tersebut dijalankan, maka akan menghasilkan keluaran 100, artinya walaupun kondisi salah, namun minimal isi instruksi akan dijalankan sekali, hal ini dikarenakan proses *do-while* berbeda dengan *while*, dimana *do-while* pertama melakukan instruksi baru mengecek kondisi, sedangkan *while* pertama mengecek kondisi baru melakukan instruksi.

g) Perulangan *for*

Perulangan *for* merupakan perulangan yang memiliki variabel untuk melakukan pengkondisian, berbeda dengan *while* dan *dowhile* yang kita harus membuat sebuah variabel diluar untuk melakukan penkondisian, pada perulangan *for*, ditempatkan sebuah blok untuk membuat variabel dan melakukan proses pengkondisian.

Bentuk pernyataan *for* seperti berikut:

```
For (inisialisasi; kondisi;
penaikan/penurunan;){
    //instruksi;
}
```

Contoh menampilkan angka 1-100:

```
For (int i=1; i<=100; i++){
    System.out.println(i);
}
```

h) Perintah *break*

Perintah *break* merupakan perintah yang dapat digunakan untuk menghentikan proses

perulangan, misal jika kita membuat program seperti berikut:

```
For (int i=1; i<=100; i++){
    System.out.println(i);
    If (i==50){
        Break;
    }
```

Maka program hanya akan menampilkan angka dari 1 sampai 50, karena pada saat *i* mencapai 50, program dihentikan oleh perintah *break*.

i) Perintah *Continue*

Perintah *continue* dapat digunakan untuk meloncati sebuah perulangan, maksudnya adalah instruksi yang seharusnya dapat dilewat, hal ini berarti instruksi tidak akan dijalankan.

Misal:

```
For (int i=1; i<=100; i++){
    System.out.println(i);
    If (i%2==0){
        Break;
    }
}
```

Jika program diatas dijalankan, maka hasilnya akan menampilkan angka-angka ganjil saja, hal ini dikarenakan saat nilai *i* merupakan angka genap, maka perintah *continue* membuat program tidak menampilkan angka genap.

5) Array

Array merupakan objek yang dapat digunakan untuk menyimpan sejumlah data. Data yang dapat ditampung pada *array* dapat berupa tipe data ataupun kelas (objek).

a) Deklarasi *array*

```
TipeArray namaArray[];
```

Dimana tipe *array* dapat berupa tipe data biasa seperti *int*, *char*, *short* atau juga kelas seperti *String* dan yang lainnya.

b) Membuat *Array*

```
TipeArray namaArray[];
namaArray=newTipe Array[jumlah];
```

Dimana jumlah *array* merupakan jumlah data yang dapat ditampung oleh *array*.

c) Manipulasi Data dalam *Array*

Setelah membuat *array*, maka untuk melakukan proses manipulasi seperti menambahkan data ke *array*, mengubah data di *array* ataupun mengakses data dalam *array*, maka diperlukan sebuah indeks, dimana saat membuat sebuah *array* dengan jumlah data 5, maka hasilnya akan terlihat seperti ini.

Tabel 2.8 *Array* dengan jumlah data 5

No.	Indeks
1	0
2	1
3	2
4	3
5	4

Setelah membuat *array*, maka untuk melakukan proses manipulasi seperti menambahkan data ke *array*, mengubah data di *array* ataupun mengakses data dalam *array*, maka diperlukan sebuah indeks, dimana saat membuat sebuah *array* dengan jumlah data 5, maka hasilnya akan terlihat seperti ini.

```
int a[] = new int[3];
a[0] = 1;
a[1] = 45;
a[2] = 10;
System.out.println(a[0]);
System.out.println(a[1]);
System.out.println(a[2]);
```

d) *Array* Multidimensi

Java mendukung pembuatan *array* multidimensi maksudnya kita dapat menambahkan data *array* ke dalam sebuah

array, cara pembuatannya adalah sebagai berikut:

```
TipeArray namaArray[][]=new
TipeArray[jumlah] [jumlah]
```

2.6 Android

Menurut Sherief Salbino (2014:3) *Android* merupakan sistem operasi berbasis *Linux* yang bersifat terbuka (*open source*) dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet.

Menurut Murya dalam Ahmad Dudayef (2014:15) *android* adalah sistem operasi berbasis *linux* yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (*PDA*). *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sendiri yang digunakan oleh bermacam piranti bergerak. *Android* kini telah menjelma menjadi sistem operasi *mobile* terpopuler di dunia. Perkembangan *android* tidak lepas dari peran sang raksasa *Google*. *Android* pada mulanya didirikan oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White pada tahun 2003.

Berdasarkan uraian diatas penulis menyimpulkan bahwa *Android* adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat layar sentuh dan bersifat terbuka.

Berikut merupakan struktur anatomi *Android* :



Gambar 2.1 Struktur Anatomi Android

2.7 Android Studio

Android Studio adalah sebuah *IDE* (*Integrated Development Environment*) untuk mengembangkan aplikasi dengan *platform android*. *Android Studio* diperkenalkan Google pada tanggal 16 Mei 2013 pada sebuah pertemuan *Google I/O* yang disampaikan oleh *Google's Product Manager* (Pengelola Produk Google). *Android Studio* menjadi *tools* (alat) yang di support oleh Google untuk pengembangan aplikasi android menggantikan *Eclipse*.

Android Studio merupakan lingkungan pengembangan berdasarkan *IntelliJ IDEA* yang mirip dengan *Eclipse* dengan *ADT Plugin*. Berikut beberapa fitur yang dimiliki *Android Studio* :

1. Berbasis *Gradle*.
2. Android-spesifik *refactoring* dan perbaikan yang cepat.
3. Alat Lint untuk menangkap kinerja, kegunaan, versi kompatibilitas dan masalah lainnya.
4. ProGuard dan app-signature.
5. Wizard untuk *design* dan membuat komponen-komponen umum sebuah layout editor yang memungkinkan untuk *drag-and-drop* komponen *UI*, pratinjau *layout* pada beberapa konfigurasi layar, dan banyak lagi.
6. Built-in dukungan untuk *Google Cloud platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messagin* dan *App Engine* sebagai komponen *server-side*.

2.8 Android SDK

Menurut Sfaat dalam Ahmad Dudayef (2014:18) *Android SDK* adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *Java*.

Menurut Akbarul dalam Ahmad Dudayef (2014:18) *Android SDK* adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*. Saat ini disediakan *Android SDK* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*.

Dari uraian diatas penulis menyimpulkan bahwa *Android SDK* adalah *tools API* wajib yang diperlukan untuk mengembangkan aplikasi *platform android*.

2.9 Unified Modeling Language (UML)

Menurut Brigida dalam Winda Fandella (2015:14) "*Unified Modeling Language* (UML) merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem/perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, Microsoft, dan sebagainya.

Menurut Nugroho dalam Winda Fandella (2015:14) UML (*Unified Modeling Language*) yang merupakan metodologi kolaborasi antara metoda-metoda *Booch*, OMT (*Object Modeling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metode yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa "pemrograman berorientasi objek" (OOP).

Dari definisi di atas dapat disimpulkan bahwa, *Unified Modeling Language* (UML) adalah bahasa pemodelan standar yang digunakan sebagai alat bantu dalam untuk merancang/ mengembangkan sistem perangkat lunak.

2.10 SQLite Database

Menurut Safaat dalam Ahmad Dudayef (2014:20) SQLite adalah salah satu *software* yang *embedded* yang sangat populer, kombinasi *SQL interface* dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. Dengan adanya *SQLite database* ini, banyak membantu dalam pembuatan *database* dari setiap versi *android* karena *SQLite database* ini termasuk dalam *Android runtime*. Untuk pengembangannya, dalam membuat dan membuka *database* dipergunakan sebuah *libraries* yang harus kita import, yaitu *android.database.sqlite*. *SQLiteOpenHelper* yang menyediakan tiga metode, adapun tiga metode tersebut adalah sebagai berikut:

1) *Constructor*

Menyediakan representasi versi dari *database* dan skema yang kita gunakan.

2) *onCreate()*

Menyediakan *SQLiteDatabase* object yang kita gunakan dalam definisi tabel dan inisialisasi data.

3) *onUpgrade()*

Menyediakan fasilitas konversi *database* dari *database* versi yang lama ke *database* versi yang baru atau sebaliknya

3 ANALISA DAN PERANCANGAN

3.1 Deskripsi Sistem

Berdasarkan latar belakang dan perumusan masalah yang terdapat dalam bab sebelumnya dapat diketahui perancangan yang dilakukan adalah membuat aplikasi pembelajaran dasar pemrograman berbasis *mobile phone*.

Aplikasi ini dibangun menggunakan *Android Studio* yang terdiri dari beberapa materi dan sub-materi untuk tingkat dasar pemrograman *Java*.

Materi - materi yang akan disampaikan dalam aplikasi:

1) Konsep Dasar

- a) Variabel
- b) Operator
- c) *Java Comments*
- d) Inkremen Dekremen
- e) *String*
- f) Inputan *User*

2) Percabangan dan Perulangan

- a) Percabangan *if*
- b) *If Bersarang*
- c) *Else if*
- d) *Switch-case*

3) Perulangan

- a) Perulangan *while*
- b) Perulangan *for*
- c) Perulangan *do-while*
- d) Perintah *continue* dan *break*

4) *Array*

- a) Pengenalan *Array*
- b) Menambah Elemen *Array*
- c) Peningkatan perulangan *for*
- d) *Array* Multidimensi

3.2 Perancangan

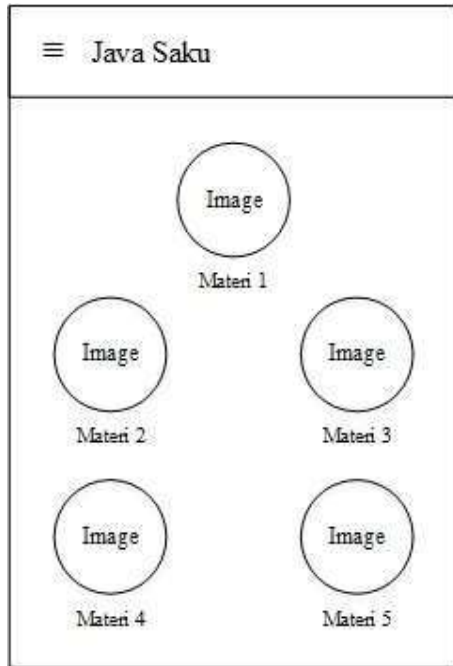
3.2.1 Perancangan *Use Case Diagram*

Perancangan antarmuka pemakai (*User Interface*) pada aplikasi ini akan ditampilkan sesuai dengan kebutuhan pengguna yang akan melihat informasi yang dibutuhkan.

Pada aplikasi ini terdapat beberapa tampilan menu yang akan memuat tampilan – tampilan informasi berdasarkan perancangan menu yang ada.

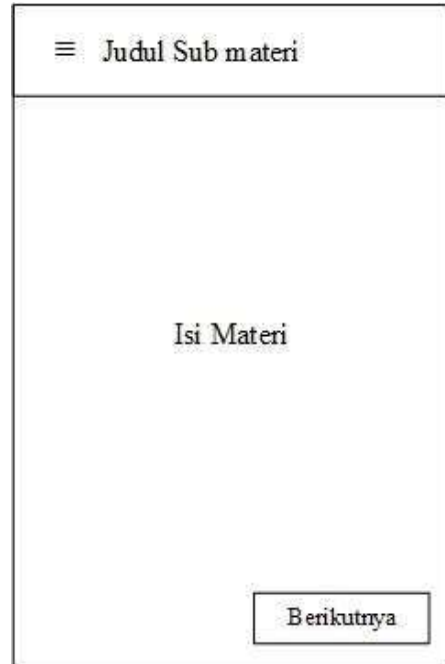
Rancangan antarmuka aplikasi ini terdiri atas :

1) Antarmuka Menu Utama



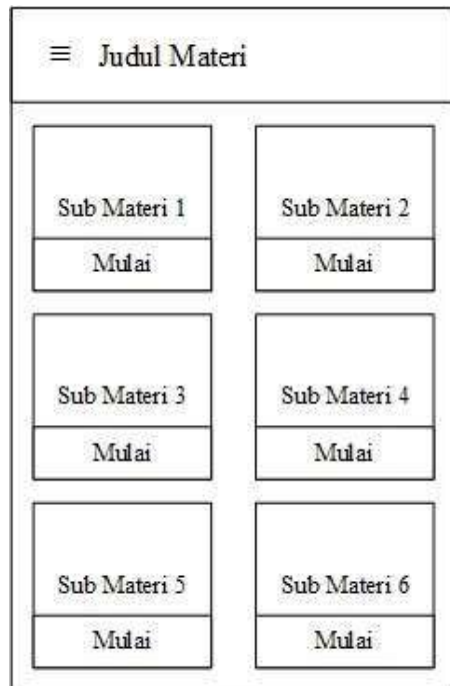
Gambar 3.1 Antarmuka Menu Utama

3) Antarmuka Pemaparan Materi



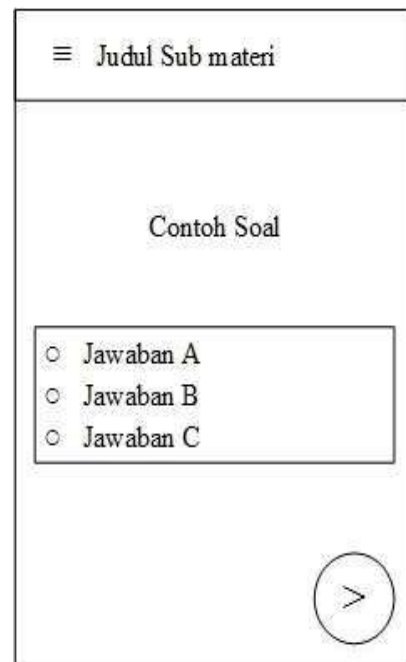
Gambar 3.3 Antarmuka Pemaparan Materi

2) Antarmuka Sub-Materi



Gambar 3.2 Antarmuka Sub-Materi

4) Antarmuka Soal Latihan Pilihan Ganda



Gambar 3.4 Antarmuka Soal Latihan Pilihan Ganda

5) Antarmuka Soal Latihan Essay

Gambar 3.5 Antarmuka Soal Latihan Essay

7) Antarmuka Jawaban Soal Latihan Salah

Gambar 3.7 Antarmuka Jawaban Salah

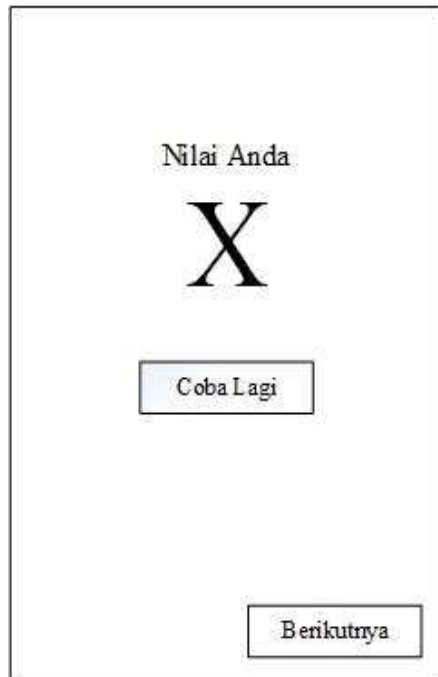
6) Antarmuka Soal Latihan Materi Melengkapi Program

Gambar 3.6 Antarmuka Soal Latihan Materi Melengkapi Program

8) Antarmuka Evaluasi

Gambar 3.8 Antarmuka Evaluasi

9) Antarmuka Hasil Evaluasi



Gambar 3.9 Antarmuka Hasil Evaluasi

11) Antarmuka Tentang Aplikasi



Gambar 3.11 Antarmuka Tentang Aplikasi

10) Menu Navigasi



Gambar 3.10 Antarmuka Menu Navigasi

12) Konfirmasi Keluar



Gambar 3.12 Antarmuka Konfirmasi Keluar Aplikasi

4 HASIL DAN PEMBAHASAN

4.1 Tampilan Halaman Pengguna

Tampilan halaman pengguna merupakan tampilan halaman aplikasi antara sistem dengan dengan pengguna. Berikut merupakan tampilan halaman hasil dari aplikasi pembelajaran dasar pemrograman berbasis *mobile phone*.

4.1.1 Tampilan Halaman Utama

Halaman utama pengguna aplikasi tanpa splash atau animasi sehingga ketika di jalankan akan langsung menampilkan pilihan ke menu pembelajaran. Di dalam halaman ini berisi tombol Konsep Dasar, Percabangan, Perulangan, Array, dan Evaluasi. Pada bagian atas kiri aplikasi terdapat tombol jika di sentuh akan menampilkan menu navigasi aplikasi.



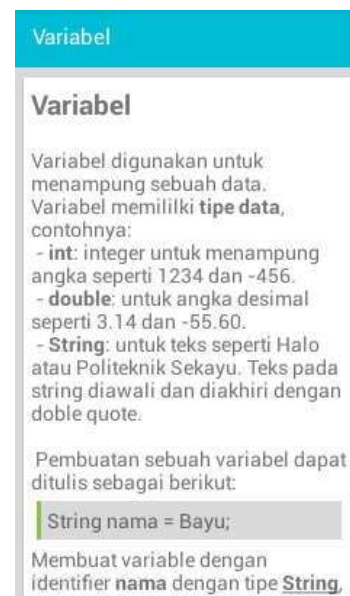
Gambar 4.1 Tampilan Halaman Utama

4.1.2 Tampilan Konsep Dasar

Halaman konsep dasar memuat tentang dasar-dasar pemrograman *Java* seperti Variabel, Operator, *Java Comments*, Inkremen dan Dekremen, String dan Inputan *User*. Ketika di tekan akan menampilkan materi yang dipilih.



Gambar 4.2 Tampilan Halaman Konsep Dasar



Gambar 4.3 Tampilan Halaman Materi Konsep Dasar

4.1.3 Tampilan Halaman Percabangan

Halaman Percabangan memuat tentang percabangan *Java* seperti Percabangan *if*, *if Bersarang*, *else if*, *switch case*. Ketika di tekan akan menampilkan materi yang dipilih.



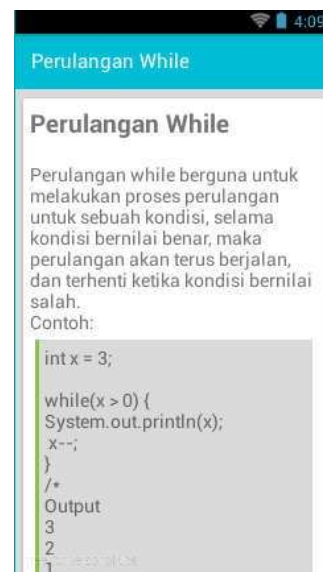
Gambar 4.4 Tampilan Halaman Percabangan



Gambar 4.6 Tampilan Halaman Perulangan



Gambar 4.5 Tampilan Halaman Materi Percabangan



Gambar 4.7 Tampilan Halaman Materi Perulangan

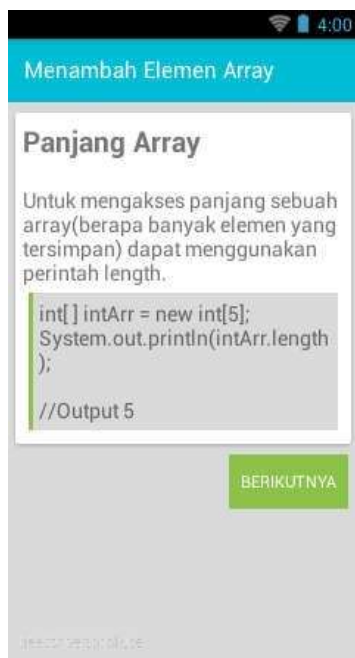
4.1.4 Tampilan Halaman Perulangan

Halaman Perulangan memuat tentang perulangan *Java* seperti Perulangan *while*, Perulangan *for*, Perulangan *do-while* dan Perintah *continue* dan *break*.

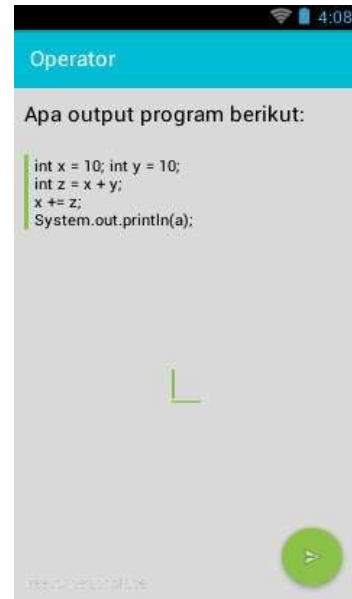
Ketika di tekan akan menampilkan materi yang dipilih.

4.1.5 Tampilan Halaman Array

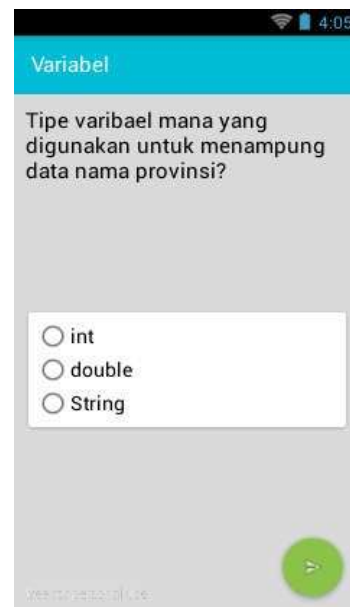
Halaman Array memuat tentang array *Java* seperti Pengenalan *Array*, Menambah Elemen *Array*, Peningkatan Perulangan *for* dan *Array* MultiDimensi. Ketika di tekan akan menampilkan materi yang dipilih.

Gambar 4.8 Tampilan Halaman *Array*Gambar 4.9 Tampilan Halaman Materi *Array***4.1.6 Tampilan Halaman Soal Materi Essay**

Pada halaman ini pengguna menginput sendiri jawaban yang ditanyakan oleh aplikasi.

Gambar 4.10 Tampilan Halaman Soal Materi *Essay***4.1.7 Tampilan Halaman Soal Materi Pilihan Ganda**

Pada halaman ini pengguna memilih salah satu jawaban yang ditanyakan oleh aplikasi.



Gambar 4.11 Tampilan Halaman Soal Materi Pilihan Ganda

4.1.8 Tampilan Halaman Soal Materi Melengkapi Program

Pada halaman ini pengguna menginput form kosong untuk melengkapi program yang tidak lengkap



Gambar 4.12 Tampilan Halaman Soal Materi Melengkapi Program

4.1.9 Tampilan Halaman Jawaban Soal Materi Salah

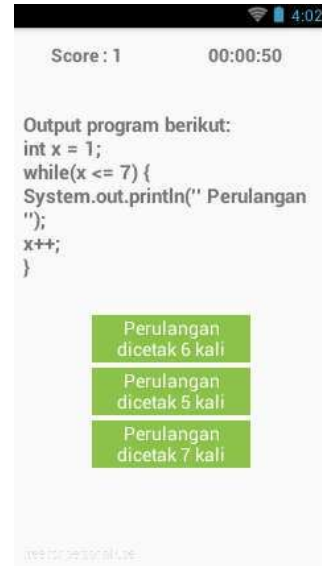
Pada halaman ini jika pengguna salah menjawab soal akan diberi pilihan coba lagi atau kembali ke materi sebelumnya.



Gambar 4.13 Tampilan Halaman Jawaban Soal Materi Salah

4.1.10 Tampilan Halaman Evaluasi

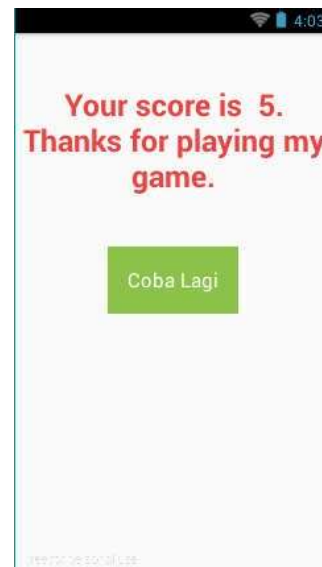
Halaman evaluasi memuat soal-soal kuis materi aplikasi pembelajaran dasar *Java*. Pengguna diberikan waktu 15 menit untuk menjawab 20 soal.



Gambar 4.14 Tampilan Halaman Evaluasi

4.1.11 Tampilan Halaman Hasil Evaluasi

Halaman hasil evaluasi memuat nilai total yang telah dijawab oleh pengguna.



Gambar 4.15 Tampilan Halaman Hasil Evaluasi

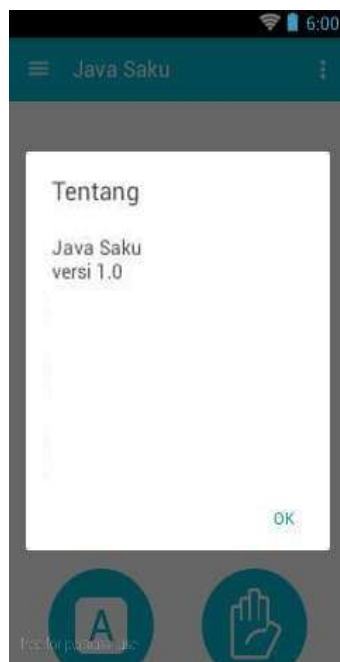
4.1.12 Tampilan Menu Navigasi



Gambar 4.16 Tampilan Menu Navigasi

4.1.13 Tampilan Halaman Tentang

Halaman tentang terdapat pada menu navigasi memuat tentang versi aplikasi.



Gambar 4.17 Tampilan Tentang Aplikasi

4.1.14 Tampilan Konfirmasi Keluar

Halaman keluar jika aplikasi menekan tombol *back* pada menu utama atau memilih tombol Keluar pada menu navigasi



Gambar 4.18 Tampilan Konfirmasi Keluar

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun kesimpulan dari pembuatan penelitian ini dapat diuraikan sebagai berikut:

- 1) Aplikasi pembelajaran dasar pemrograman ini dibuat dalam bentuk *android* yang *portable*.
- 2) Aplikasi pembelajaran dasar pemrograman ini tidak hanya berisikan materi namun juga disertai soal setelah penjelasan materi dan evaluasi soal.

5.2 Saran

Dari hasil penelitian yang dilakukan selanjutnya terdapat beberapa saran yang nantinya dapat dikembangkan untuk aplikasi ini diantaranya:

- 1) Pengayaan materi lanjutan seperti materi dasar-dasar Objek dan Kelas, serta pemahaman tentang pengenalan OOP (*Object Oriented Programming*).

- 2) Penambahan fitur pencarian dan *code playground* agar pengguna dapat menjalankan kode langsung dari aplikasi.
- 3) Ditambahkan animasi, efek suara dan video tutorial pembelajaran..

DAFTAR PUSTAKA

- Dudayef, Ahmad, 2015, Aplikasi Tuntunan Sholat Dan Pemetaan Masjid Berbasis Android , *Skripsi*, Tidak dipublikasikan, Politeknik Sekayu, Sekayu.
- Fandella, Winda, 2015, Aplikasi Pembelajaran Matematika Interaktif Berbasis Multimedia Tingkat Pendidikan Sekolah Dasar Kelas I (Satu), *Skripsi*, Tidak dipublikasikan, Politeknik Sekayu, Sekayu.
- Hendrayudi, 2009, *VB 2008 Untuk Berbagi Keperluan Programming*, Elex Media Komputindo, Jakarta.
- Heryanto, I, Raharjo, B & Haryono, A 2010, *Mudah belajar java*, Informatika, Bandung.
- Hidayat, Arif, 2014, *Tutorial Dasar java*, Blue Publisher, Jawa Timur.
- Kadir, A. 2011. *Mudah Menjadi Programmer*. Yogyakarta : Yescom.
- Kurniawan,, Eko, 2011, *Belajar Java Dasar*, Strip Bandunk, Bandung.
- Jakuben, Ben, 2012, *Java Basics for Android Development*, [Online] Tersedia:
<http://blog.teamtreehouse.com/java-basics-for-android-development-part-1/> [22 Juni 2016].
- Kendall, 2010, *Pemodelan Visual dengan UML*, Graha Ilmu, Yogyakarta.
- Rahman, M. 2014. Aplikasi Parawisata Kabupaten Musi Banyuasin Berbasis Android. Program Diploma Teknik Informatika Politeknik Sekayu : Tugas Akhir Tidak Diterbitkan
- Rosyada, Dede, 2008, *Media Pembelajaran*, Gaung Persada Press, Jakarta.
- Salbino, Sherief, 2014, *Buku Pintar Gadget Android Untuk Pemula*, Kunci Komunikasi, Jakarta.
- Subakti, Irfan, 2006, *Interaksi Manusia dan Komputer*, Insitut Teknologi Sepuluh November, Surabaya
- Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta : Erlangga
- Waloeoyo, Yohan, 2010, *Google Android – Sistem Operasi Ponsel Masa Depan*, CV. ANDI OFFSET, Yogyakarta.
- Zapata, Belen, 2013, *Android Studio Application Development*, Pact Publishing, United Kingdom.