

## **BAB II**

### **LANDASAN TEORI**

#### **1.1. KONSEP PERANCANGAN SISTEM**

##### **1.1.1. Pengertian Perancangan Sistem**

Perancangan sistem adalah kegiatan merancang dan menentukan cara mengolah sistem informasi dari hasil analisa sistem sehingga sistem tersebut sesuai dengan requirement. Perancangan sistem merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang ada [3]. Perancangan sistem adalah sekumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan. Hal itu bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan user.

##### **1.1.2. Tahapan Perancangan Sistem**

Tahapan perancangan sistem adalah merancang sistem dengan terperinci berdasarkan hasil analisis sistem, sehingga menghasilkan model sistem baru [4].

Berikut tahapan-tahapan perancangan sistem :

1. Perancangan *Output*

Perancangan *output* tidak dapat diabaikan, karena laporan yang dihasilkan harus memudahkan bagi setiap unsur manusia yang membutuhkan.

2. Perancangan *Input*

Tujuan dari perancangan *input* yaitu dapat mengefektifkan biaya

pemasukan data, mencapai keakuratan yang tinggi, dan dapat menjamin pemasukan data yang akan diterima dan dimengerti oleh pemakai.

### 3. Perancangan Proses Sistem

Tujuan dari perancangan proses sistem adalah menjaga agar proses data lancar sehingga dapat menghasilkan informasi yang benar dan mengawasi proses dari sistem.

### 4. Perancangan Database

Database sistem adalah mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

### 5. Tahapan Perancangan Kontrol

Tujuan perancangan ini agar keberadaan sistem setelah diimplementasikan dapat memiliki kehandalan dalam mencegah kesalahan, kerusakan, serta kegagalan proses sistem.

#### **1.1.3. Tujuan Perancangan**

Tahap perancangan/ desain sistem mempunyai 2 tujuan utama yaitu [4]:

1. Untuk memenuhi kebutuhan pemakai sistem
2. Untuk memberikan gambaran yang jelas dari rancangan bangun yang lengkap pada pemrograman computer dan ahli-ahli teknik yang terlihat (lebih condong pada disain sistem terperinci)

## **1.2. KONSEP SISTEM INFORMASI MANAJEMEN**

### **1.2.1. SISTEM**

Sistem sudah sangat populer pada saat ini. Teknologi ini sudah banyak digunakan untuk mendeskripsikan banyak hal, khususnya bagi aktivitas-aktivitas yang diperlukan di pemrosesan data. Terdapat berbagai pendapat yang mendefinisikan definisi sistem, yaitu:

Sutabri [5] mendefinisikan “Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu”.

Fathansyah [6] menyatakan “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas jumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu”.

Menurut Japerson [7] “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu”.

Berdasarkan uraian di atas, dapat disimpulkan bahwa sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dari sistem tersebut.

### **1.2.2. INFORMASI**

Informasi adalah hasil dari pengolahan beberapa data yang diorganisasikan dengan cara tertentu sehingga menjadi sebuah bentuk yang bermakna dan

bermanfaat bagi penerima.

Menurut Japerson [7] “Informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan”.

Menurut Rochaety [8] “Informasi merupakan kumpulan data yang telah diolah baik bersifat kualitatif maupun kuantitatif dan memiliki arti lebih luas”.

Menurut Kristanto [9] “Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima”.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa informasi merupakan data yang telah diproses dan digunakan untuk pengambilan keputusan.

### **1.2.3. SISTEM INFORMASI**

Sistem informasi merupakan hal yang sangat penting dalam pengambilan keputusan. Tapi tidak semua data yang mengalir digunakan sebagai bahan pertimbangan pengambilan keputusan. Oleh karena itu dibutuhkan suatu sistem informasi yang dapat mengelola data. sistem itu harus dirancang sedemikian rupa agar dapat menentukan validitas data yang berasal dari berbagai sumber.

Kristanto [9] menyatakan bahwa “Sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak computer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut”.

Menurut Ahmad dan Munawir [10] menyatakan bahwa “Sistem Informasi dalam suatu pemahaman yang sederhana dapat didefinisikan sebagai satu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan

kebutuhan serupa”.

Cegielski et.al [11] “Sistem informasi adalah proses mengumpulkan, memproses, menganalisis, dan menyebarkan informasi untuk tujuan”.

Dari pengertian diatas, penulis dapat menyimpulkan bahwa sistem informasi adalah suatu sistem yang dibuat untuk menghasilkan suatu informasi.

#### **1.2.4. MANAJEMEN**

Manajemen dapat dijabarkan secara luas dan para ahli memiliki sudut pandang tersendiri dalam mengartikan manajemen. Secara garis besar, manajemen merupakan aktivitas perencanaan dan koordinasi terhadap sumber daya yang tersedia.

Sutabri [5] mengungkapkan “Manajemen kita pahami sebagai proses atau kegiatan yang menjelaskan apa yang dilakukan oleh manajer pada operasional organisasi mereka untuk merencanakan, mengorganisasikann, memprakarsai dan mengendalikan operasi”.

Menurut Darmawan dan Fauzi [12] “Manajemen dapat dikatakan juga sebagai suatu kegiatan pengelolaan yang dimulai dari analisis kebutuhan, perencanaan, pengorganisasian, pengontrolan dan penilaian”.

Menurut Gesi [13] “Manajemen adalah sebuah proses untuk mengatur sesuatu yang dilakukan oleh sekelompok orang atau organisasi untuk mencapai tujuan organisasi tersebut dengan cara bekerja sama memanfaatkan sumber daya yang dimiliki”.

Dari pengertian diatas, penulis dapat menyimpulkan bahwa manajemen adalah manajemen dapat dikatakan sebuah proses untuk mengatur suatu kegiatan

yang dimulai dari analisis kebutuhan, perencanaan, pengorganisasian, pengontrolan dan penilaian”.

### 1.3. APLIKASI ANDROID

#### 1.3.1. Pengertian Android

*Android* merupakan sebuah sistem operasi (OS) yang bersifat *Open Source* (terbuka) yang dimiliki oleh Google.Inc. Pada awal peluncurannya, *Android* hanya digunakan untuk perangkat *Mobile*, yaitu telepon seluler. Namun seiring perkembangannya, sejak *Android* 3.0 (Honeycomb) diluncurkan, sistem operasi *Android* resmi digunakan dalam komputer tablet. Wahyutama dkk [14] menerangkan bahwa *Android* adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet yang bersifat *open source*. *Android* terus berkembang dan hal itu di tandai dengan pembaharuan versi-versinya.

#### 1.3.2. Versi Android

*Android* adalah OS atau sistem operasi smartphone yang paling banyak digunakan. Terdapat beberapa versi-versi *Android* yaitu :

**Tabel 2.1 Versi Android [15]**

No	Versi Kode	Versi Platform	Level Api
1	(no code name)	Android 1.0	Api Level 1
2	(no code name)	Android 1.1	Api Level 2
3	CUPCAKE	Android 1.5	Api Level 3, NDK 1
4	DONUT	Android 1.6	Api Level 4, NDK 2
5	ÉCLAIR	Android 2.0	Api Level 5

		Android 2.0.1 Android 2.1	Api Level 6 Api Level 7, NDK 3
6	FROYO	Android 2.2.X	Api Level 8, NDK 4
7	GINGERBREAD	Android 2.3 Android 2.3.1 Android 2.3.2	Api Level 9, NDK 5
8	GINGERBREAD	Android 2.3.3 Android 2.3.7	Api Level 10
9	HONEY COMB	Android 3.0	Api Level 11
10	HONEY COMB	Android 3.1	Api Level 12, NDK 6
11	HONEY COMB	Android 3.2.X	Api Level 13
12	ICE CREAM SANDWICH	Android 4.0, 4.0.1, 4.0.2	Api Level 14, NDK 7
13	ICE CREAM SANDWICH	Android 4.0.3, 4.0.4	Api Level 15, NDK 8
14	JELLYBEAN	Android 4.1.X	Api Level 16
15	JELLYBEAN	Android 4.2.X	Api Level 17
16	JELLYBEAN	Android 4.3.X	Api Level 18
17	KIT KAT	Android 4.4 – Android 4.4.4	Api Level 19
18	LOLLIPOP	Android 5.0	Api Level 20
19	LOLLIPOP	Android 5.1	Api Level 21
20	MARSHMALLOW	Android 6.0	Api Level 23
21	NOUGAT	Android 7.0	Api Level 24
22	NOUGAT	Android 7.1	Api Level 25
23	OREO	Android 8.0	Api Level 26
24	PIE	Android 9.0	Api Level 28

### 1.3.3. Aplikasi Berbasis Android

Aplikasi Android menggunakan bahasa java atau kotlin, hal ini dapat mengontrol perangkat mobile melalui google-enabled java. Aplikasi Android dikembangkan dalam Bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK) [16]. Pengembangan aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka dibawah skema lisensi apapun yang mereka inginkan. Pengembang memiliki beberapa pilihan dalam membuat aplikasi yang berbasis Android. Arsitektur yang tersedia pada sistem operasi Android adalah sebagai berikut [17]

- a. *Linux Kernel* Digunakan Google untuk membangun sistem android yang mencakup *memory management, security setting, power management* dan beberapa *driver hardware*.
- b. *Android Runtime* Terdiri dari *Core Libraries* dan *Dalvik Virtual Machine*.
- c. *Libraries Android* menyertakan satu *set library-library* dalam bahasa C/C++ yang digunakan pada komponen-komponen Android.
- d. *Application Framework Programmer* memiliki akses penuh untuk memanfaatkan *API (Android Protocol Interface)* dala mengatur fungsi-fungsi dasar smartphone.
- e. *Application* Merupakan fungsi teratas Android dan fungsi yang paling sering di akses bagi para pengguna

Web service ReST API berfungsi untuk menyediakan data yang diminta oleh aplikasi. Sebelum menyediakan data yang diminta oleh aplikasi client, ReST



API terlebih dahulu menghubungi server dalam hal ini yaitu MySQL database. Maka dari itu web service ReST API di sini bertindak sebagai jembatan antara peminta layanan dengan penyedia layanan (*database*) [18]

#### **1.4. ALAT BANTU PEMODELAN SISTEM**

Adapun alat bantu pemodelan sistem yang digunakan penulis yaitu UML (*Unified Modeling Language*), *Use case Diagram*, *Activity diagram*, *Class Diagram*. Alat bantu ini digunakan untuk perancangan sistem yang akan dibangun.

##### **1.4.1. UML (Unified Modeling Language)**

Untuk menganalisis kebutuhan sistem penulis menggunakan alat bantu yaitu dengan *use case* diagram. Tujuan pembuatan *use case* adalah untuk mendapatkan dan menganalisis informasi persyaratan yang cukup untuk mempersiapkan model yang mengkomunikasikan apa yang diperlukan dari perspektif pengguna, tetapi bebas dari detail fisik tentang bagaimana sistem akan dibangun dan diimplementasikan. Dan ada pula beberapa pengertian menurut para ahli antara lain.

Sukamto dan Shalahuddin [19] mengungkapkan “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Munawar [20] “UML(*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunai pengembangan sistem yang berorientasi objek”.

Sari dan Utami [21] “UML adalah metode terbuka untuk menspesifikasikan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak yang sedang dikembangkan”.

Berdasarkan definisi atas dapat disimpulkan bahwa “*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)”.

#### **1.4.2. Use case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* adalah pola atau gambaran berbentuk diagram yang menggambarkan hubungan suatu sistem yang tengah di buat. Dalam penggambarannya, sistem yang dibuat harus berada didalam kotak sistem dan memiliki minimal satu aktor yang berada di luar sistem yang didefinisikan oleh para ahli dibawah ini :

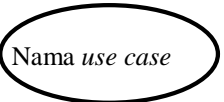


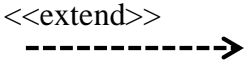

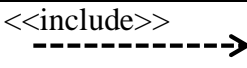
Sukamto dan Shalahuddin [19] menyatakan bahwa “*Use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Munawar [20] “*Use case Diagram* yaitu deskripsi fungsi dari sebuah sistem dari persepektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistem sendiri melalui sebuah cerita bagai mana sistem dipakai”.

Sari dan Utami [21] “*Use Case* adalah deskripsi fungsi dari sebuah system dari perspektif/sudut pandang para pengguna sistem”.

Dari beberapa pendapat tersebut dapat disimpulkan bahwa *use case* diagram merupakan permodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

**Tabel 2.2 Simbol *Use case Diagram* [19]**

<b>Nama Simbol</b>	<b>Simbol</b>	<b>Deskripsi</b>
<i>Use case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
<i>Aktor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<i>Association</i> atau asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<i>Extend</i> atau ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
<i>Generalisasi</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang

		ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>
--	--	--

### 1.4.3. Activity diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut tiga definisi *activity diagram* menurut para ahli :

Sukanto dan Shalahuddin [19] mengatakan bahwa : “Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.


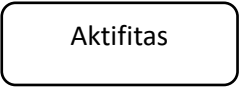



Munawar [20] mengungkapkan “*Activity diagram* adalah untuk menangkap tingkah laku dinami dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya”.

Muslihudin dan Oktafianto [22] “Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem”.

Jadi kesimpulan dari pengertian *Activity diagram* adalah sebuah model yang menyerupai *flowchart* horizontal yang menunjukkan urutan tindakan yang merekam sebuah alur proses dan hasil dari tindakan tersebut.

**Tabel 2.3 Simbol *Activity diagram* [19]**

Nama Simbol	Simbol	Deskripsi
-------------	--------	-----------

Status Awal		Status awal aktifitas sistem, sebuah diagram memiliki sebuah status awal
Aktifitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas atau lebih dari satu
Penggabungan / <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status Akhir

#### 1.4.4. Class Diagram

*Class Diagram* dapat merupakan implementasi dari sebuah *interface*, yaitu class abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Berikut ini merupakan penjelasan *Class Diagram* dari tiga ahli sebagai berikut :

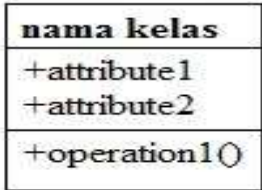



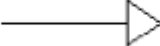

Sukamto dan Shalahuddin [19] mengungkapkan bahwa “Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Munawar [20] mengungkapkan “*Class Diagram* menggambarkan atribut, operation dan juga constraint yang terjadi pada sistem”.

Muslihudin dan Oktafianto [22] menyatakan “*Class Diagram* memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Umum dijumpai pada pemodelan sistem berorientasi objek”.

Jadi kesimpulan dari pengertian *Class Diagram* adalah adalah kumpulan *object* yang menggambarkan struktur statis dari sebuah sistem yang menunjukkan *object class* dan hubungannya serta untuk pengembangan sistem tersebut.

**Tabel 2.4 Simbol *Class Diagram* [19]**

Simbol	Nama	Keterangan
	<i>Kelas</i>	Kelas pada struktur sistem.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Asosiasi</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Asosiasi berarah</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Generalisasi</i>	Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).
	<i>Agregasi</i>	Relasi antar kelas dengan makna semua bagian ( <i>wholepart</i> )

## **1.5. ALAT BANTU PERANCANGAN PROGRAM**

### **1.5.1. Android Studio**

*Android Studio* adalah sebuah IDE untuk *Android Development* yang diperkenalkan Google pada acara Google I/O 2013. *Android studio* merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. *Android Studio* merupakan IDE resmi pengembangan aplikasi *Android*.

*Android Studio* memiliki fitur *Flexible Gradle*, dapat menggunakan beberapa generasi APK, *Template* menarik yang cocok untuk *Google Services* dan berbagai jenis *Device*, kaya akan *Layout* dan *Lint Tools* yang berguna untuk mengetahui performa, pemakaian, kecocokan versi dan masalah lainnya.

Jadi, *Android Studio* merupakan aplikasi resmi untuk pengembangan aplikasi *Android* yang disediakan oleh Google dimana *Android Studio* ini dikembangkan di IntelliJ IDEA Community Edition oleh JetBrains.

### **1.5.2. Basis Data (*Database*)**

Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh sistem informasi. Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel- tabel yang saling berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada [23]

Suatu bangunan basis data memiliki jenjang sebagai berikut :

1. Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau karakter khusus yang membentuk sebuah item data atau *field*.
2. *Field/item*, merupakan representasi suatu atribut dan *record* (rekaman/tupel) yang sejenis yang menunjukkan suatu item dari data.
3. *Record*/rekaman/tupel: Kumpulan dari *field* membentuk suatu *record* atau rekaman. *Record* menggambarkan suatu unit data individu yang tertentu.
4. *File*, merupakan kumpulan dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Mewakili tiap-tiap data.
5. *Database*, merupakan kumpulan dari file atau tabel yang membentuk suatu *database*.

### 1.5.3. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL. *MySQL* merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi *General Public License* (GPL). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

Menurut Enterprise [24] “MySQL adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan”.

Menurut Hidayatullah dan Kawistara [25] “MySQL adalah salah satu aplikasi DBMS yang sudah banyak digunakan oleh para pemrograman aplikasi web”.



Menurut Raharjo [26] “MySQL merupakan software RDBMS (atau *server database*) yang dapat mengelola dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user(*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)”.

Berdasarkan beberapa definisi di atas dapat disimpulkan bahwa *MySQL* adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL.

#### 1.5.4. LARAVEL

Laravel adalah kerangka kerja aplikasi web berbasis PHP yang *open source*, menggunakan konsep *model-view-controller*. Laravel berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.

Menurut Yudhanto dan Prasetyo [27] “Laravel adalah sebuah *Framework* PHP yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*) .

Laravel adalah sebuah MVC *web development framework* yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi [28].

Laravel memberikan keterbaruan alat untuk berinteraksi dengan database disebut dengan migration [28]. Dengan migration, pengembang dapat dengan

mudah untuk melakukan modifikasi sebuah *database* pada sebuah platform secara independen karena implementasi skema database direpresentasikan dalam sebuah class. Migration dapat berjalan pada beberapa basis data yang telah didukung Laravel (MySQL, PostgreSQL, MSSQL, dan SQLITE) dan untuk implementasi *Active Record* pada Laravel disebut *Eloquent* yang menggunakan standard modern OOP. Laravel juga memberikan sebuah *Command Line Interface* disebut dengan artisan dengan artisan, pengembang dapat berinteraksi dengan aplikasi untuk sebuah aksi seperti *migrations*, *testing*, atau membuat *controller* dan *model*. Selain itu, laravel juga memiliki Blade template engine yang memberikan estetika dan kebersihan kode pada *view* secara parsial.

#### **1.5.5. BALSAMIQ**

Balsamiq Mockup merupakan salah satu *software* yang di gunakan dalam pembuatan desain atau *prototype* dalam pembuatan tampilan *user interface* sebuah aplikasi.

Menurut Nurseptaji [29] “Balsamiq Mockups adalah aplikasi yang sangat berguna untuk mendesain halaman web”.

Menurut Khasanah [30] “Balsamiq mockups 3 merupakan *software* yang digunakan untuk pembuatan tampilan antarmuka pengguna atau *user interface* sebuah aplikasi”.

Menurut Bambang dan Ridlo [31] “Mockup antar muka pengguna (user interface) merupakan salah satu jenis model yang lincah yang dapat digunakan untuk mempresentasikan persyaratan dalam bahasa yang dapat dimengerti oleh kedua belah pihak yaitu pengguna dan pengembang aplikasi”.

Berdasarkan pendapat tersebut dapat kita simpulkan bahwa Balsamiq mockups merupakan *software* yang dapat digunakan untuk mendesain tampilan antarmuka pengguna atau *user interface* sebuah aplikasi.

## 1.6. PENGUJIAN APLIKASI

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Langkah-langkah dalam pengujian *software* antara lain :

1. *Unit testing*, merupakan suatu aktifitas mencoba alur proses aplikasi pada struktur kontrol untuk memastikan fungsionalitas secara lengkap dan memaksimalkan pendeteksian kesalahan.
2. *Integration testing*, merupakan sekumpulan dari permasalahan yang ditemukan pada verifikasi dan konstruksi program.
3. *High-order test* yaitu aplikasi yang telah selesai akan diintegrasikan atau dirancang menjadi bagian yang lengkap dan menjadi satu kesatuan.
4. *Validation test*, yaitu memberikan jaminan akhir bahwa aplikasi yang dirancang telah memenuhi semua kriteria yang diperlukan.

Salah satu metode pengujian yang berfokus pada spesifikasi fungsionalitas dari perangkat lunak disebut *Black Box Testing*. Ningrum [32] “Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah software tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing masing”.

*Black Box Testing* digunakan untuk mendeteksi permasalahan berikut:

- a. Fungsi yang salah atau hilang.
- b. Kesalahan pada interface.
- c. Kesalahan struktur data dan basis data.
- d. Kesalahan fungsi.
- e. Kesalahan deklarasi dan terminasi.

### 1.7. PENELITIAN SEJENIS

Studi penelitian sejenis adalah kajian dari pendapat orang lain atau peneliti yang dapat dijadikan sebagai sumber teori serta acuan bagi peneliti dalam melaksanakan penelitian. Tinjauan penelitian sejenis dapat dilakukan untuk menemukan metode yang relevan dengan permasalahan penelitian. Hasil tinjauan penelitian sejenis dapat dijadikan sebagai dasar bagi peneliti untuk memilih metode dan melakukan pengembangan terhadap solusi dari permasalahan yang ada. Tabel 2.5 menyajikan penelitian-penelitian yang relevan dengan topik yang peneliti angkat.

**Tabel 2.5 Kajian Penelitian Sejenis**

<b>No.</b>	<b>Judul</b>	<b>Masalah</b>	<b>Metode</b>	<b>Kesimpulan</b>
1.	Penerapan Sistem Informasi Manajemen Kegiatan Masjid Berbasis Web Pada Masjid Budi Luhur	Masalah administrasi juga menjadi salah satu penyebab kurang efektifnya kegiatan-kegiatan. Sebagai contoh, di dalam pengelolaan pemasukan dan pengeluaran	Wawancara, Observasi Langsung, Analisa Proses Bisnis, Identifikasi Masalah, Analisa Kebutuhan,	Dengan adanya fasilitas pendaftaran acara yang diadakan oleh pengurus DKM Budi Luhur secara online diharapkan dapat meningkatkan minat jemaah dalam mengikuti seluruh acara syi'ar islam.

	Khaerul Anam, Hendri Irawan [33]	keuangan Masjid Budi Luhur yang saat ini masih menggunakan perekapan manual menggunakan buku dan disimpan di lemari saja	Merancang Sistem Usulan	
2.	Digitalisasi sistem informasi manajemen masjid modern  Sutono, Ai Musrifah, Repi Maulana Risyan [34]	Selama ini pengelolaan kegiatan manajemen masjid kurang optimal karena pimpinan sulit memonitoring kegiatan dan Jemaah pun kurang puas dalam penerimaan informasi terhadap kegiatan yang dilakukan oleh pengelola masjid sehingga harus ada solusi dari permasalahan tersebut.	<i>Waterfall</i>	Mampu memberikan kepuasan pada pengguna system dan menganggap bahwa keberadaan system pada pengelolaan kegiatan operasional masjid dianggap penting
3.	Sistem Informasi Manajemen Pengelolaan Masjid Raya Sabilal Muhtadin Banjarmasin  Wagino [35]	Melakukan pengelolaan masjid seperti mengarsipkan data pengurus masjid, khas masuk, khas keluar, data penceramah, data imam sholat wajib, imam sholat jumat dan khatib serta pengelolaan	<i>Waterfall</i>	Dengan Adanya Sistem Informasi yang dibuat ini, pengolahan data dan pencarian data serta pembuatan laporan sudah dilakukan secara terkomputerisasi sehingga informasi yang dihasilkan menjadi lebih cepat, tepat dan akurat.

		<p>penggunaan aula serbaguna masih menggunakan buku besar. Sedangkan cara yang digunakan itu membuka peluang kekeliruan data, kesalahan, kurang cepat dan tidak efisien.</p>		
4.	<p>Implementasi Sistem Informasi Manajemen Masjid Berbasis Web</p> <p>Herfandi Herfandi, Fahri Hamdani [36]</p>	<p>Pada skala lingkungan masjid selama ini biasanya laporan masih dibuat menggunakan cara manual dan dihitung menggunakan mesin penghitung dan dilaporkan melalui corong mikrofon yang di-update di setiap Jum'at, di setiap bulan, di setiap akhir tahun atau di setiap musyawarah pertanggung jawaban kepengurusan (takmir) masjid</p>	<i>Waterfall</i>	<p>Dengan adanya sistem informasi ini dapat dipergunakan semaksimal mungkin oleh masyarakat serta petugas masjid. Sistem informasi manajemen masjid ini diharapkan akan terus dikembangkan sesuai kebutuhan yang ada dan sesuai perkembangan teknologi.</p>
5	<p>Perancangan Sistem Informasi Pengelolaan Masjid Berbasis Java Pada Masjid Al-Ikhlas Pondok Aren</p>	<p>Sistem pengelolaan keuangannya yang masih manual sehingga banyak menimbulkan kesulitan dalam dewan kemakmuran masjid</p>	<i>Metode Grounded</i>	<p>Rancangan aplikasi sistem informasi pengelolaan masjid pada Masjid Al-Ikhlas Pondok Aren dapat lebih cepat, efektif, efisien, aman, terkonsep dan up to date dalam pengolahan datanya, dapat mengurangi terjadinya</p>

	Erwin Haryono, Ika Mei Lina, Endaryono [37]	melaksanakan tugas pengelolaan data dan keuangan masjid.		kesalahan dalam perhitungan keuangan masjid, sistem informasi yang terintegrasi, mudah diakses dan terkomputerisasi serta dapat mempermudah pembuatan laporan secara terperinci dan akurat
--	---	--	--	--

Penelitian sejenis yang telah diuraikan pada Tabel 2.5, dapat ditarik kesimpulan bahwa permasalahan utama yang melatar belakangi adalah Sistem pengelolaan keuangannya yang masih manual sehingga banyak menimbulkan kesulitan dalam dewan kemakmuran masjid melaksanakan tugas pengelolaan data dan keuangan masjid. Dengan dilakukan perancangan sistem informasi manajemen masjid yang pada objek-objek penelitian dari penelitian sejenis Tabel 2.5. Kesimpulan dari hasil penelitian sejenis adalah dapat mengurangi terjadinya kesalahan dalam perhitungan keuangan masjid, sistem informasi yang terintegrasi, mudah diakses dan terkomputerisasi serta dapat mempermudah pembuatan laporan secara terperinci dan akurat.

Berdasarkan kelima penelitian sejenis pada Tabel 2.5 serta melihat kelemahan dari penelitian-penelitian tersebut, penulis akan merancang sistem informasi manajemen masjid berbasis mobile yang memiliki perbedaan yaitu aplikasi yang dibangun dapat dijalankan dengan mobile android.