

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Profile Masjid Ash-Shobirin**

Masjid Ash-shobirin beralamat digang Ash-shobirin, Babakan Jawa, Cisitu Indah 6 RT.01/04 Dago Cobleng. Penduduk yang berdomisili di sekitar masjid Ash-shobirin ada sekitar 600 jiwa dengan mayoritas yang beragama islam hampir 95%. Masjid ini berdiri sekitar 20 tahun yang lalu atau tepatnya tahun 1998 dengan bentuk yang belum permanen yaitu terbuat dari bambu dan berdiri diatas empang. Sebagai informasi masjid ini berdiri di tanah wakaf dan bangunan masjid ini menjadi bangunan permanen pada tahun 2006.

Masjid Ashshobirin terletak di daerah pemukiman yaitu tepatnya di Lembah Cisitu dan terletak digang sempit (Gg. Masjid Ashshobirin) perkampungan Babakan Jawa Cisitu Indah 6, Dago Cobleng. Daerah lokasi masjid Ashshobirin berbatasan langsung dengan sungai Cikapundung di sebelah timur, sebelah barat dengan jalan Cisitu Indah 6, sebelah utara dan selatan masing-masing berbatasan langsung dengan Lembah cisitu atau (Parahyangan Reksa Raga (PASAGA)) dan Perumahan Kampung Padi. Masjid ini sebagai sebagai salah satu pusat madrasah untuk anak-anak dalam bentuk pengajian (TPA) maupun kegiatan edukatif lainnya. Selain sebagai tempat ibadah, masjid ini sebagai tempat silaturahmi baik dalam acara di bidang keagamaan maupun kemasyarakatan yang ada di RT 01 pada khususnya dan RW 04 pada umumnya. Masjid ini menjadi strategis karena letaknya yang langsung berdampingan dengan rumah warga baik yang berdomisili di perkampungan Babakan Jawa, Lembah Cisitu atau PASAGA dan warga di perumahan Kampung Padi, Cisitu Cobleng Dago. Sehingga dapat disimpulkan Masjid Ashshobirin terletak di pemukiman padat penduduk yang mayoritas penduduknya muslim.

## **2.1.1 Visi Dan Misi Masjid Ash-shobirin**

Masjid Ashshobirin memiliki Visi dan Misi. Adapun Visi dan Misinya adalah sebagai berikut.

### **2.1.1.1 Visi Masjid Ash-shobirin**

Menjadikan Masjid Ashshobirin Pusat Kegiatan Dakwah dan Pmberdayaan Umat

### **2.1.1.2 Misi Masjid Ash-shobirin**

Adapaun Misi dari Masjid Ashshobirin adalah sebagai berikut :

1. Memfasilitasi sarana ibadah, menjadi wadah silaturahmi dan kegiatan-kegiatan bagi seluruh jamaah muslim Indonesia di RT.01 khususnya, dan di RW.04 Csitu Dago Coblom pada umumnya.
2. Menerima.mengelola, dan mendistribusikan dana untuk pendidikan islam.Menyediakan sarana dan prasarana ibadah yang sesuai standar.
3. Meningkatkan pemahaman dan tenggang rasa serta toleransi di antara komunitas-komunitas muslim dan non muslim
4. Mewujudkan lingkungan masyarakatIslami di lingkungan masjid dan masyarakat sekitarnya. Menjadi islam adalah agama yang cinta perdamaian, anti kejahatan, anti radikalisme dan anti terorisme.

### 2.1.1.3 Susunan Panitia Pembangunan Masjid Ash-shobirin

**Tabel 2. 1 Susunan Panitia Masjid Ash-Shobirin**

Pelindung	: Letnan Jenderal (Purnawirawan) H. Solihin G.P.
	: Ir.H.Rachmadi Iskandar Dinata (Putra Bp. Otto Iskandar Dinata)
	: Dr.H.Dedy Djamaludin Malik
	: Ustadz.Dr,H.Deden Suhendar,M.Si
Penasehat	: Camat Kecamatan Coblong
	: Lurah Kelurahan Dago
	: Ketua RW.04
	: H.Olih Solihin
	: Ir.H.Efendi Sutanto
	: H.Srihadi Sucipto
	: Ketua RT.01
	: Ustadz Dadang Ruhiyat
	: Suparman
	: Momod Sukandi
Ketua	: H.Herdar Rukman
Sekretaris	: Rusli
Bendahara	: Mamad Mahludin
Seksi Teknik dan Bangunan	: H.Olih Solihin (Koordinator)
	: Otom Sukandi
	: Usman

	: Jaka
	: Agus Suyatno
Seksi Dana	: Drs.Asep Burhanudin (Koodirnator)
	: dr. Reza Widiyanto Sudjud
	: Oman Rohman
	: Suropto
	: Ayi Sudari
	: Teddi Prasasti
	: Ninik Rohaeti
	: Mamat Ruhimat
Seksi Perlengkapan	: Wagiman (Koordinator)
	: Darmin
	: Armen
	: Didin
	: Tunjang
	: Nurohman
	: Wasio
Seksi HUMAS	: Tri Sunarto (Koordinator)
	: Eri Rahman
	: Enjang
	: Dodi Hardadi
	: Budi Sujana
	: Atep Suparman
Seksi Dokumentasi	: Kusnadi (Koordinator)
	: Mistanto
	: Iri Rustandi

	: Ponimin
	: Yayat
	: Darya
Seksi Konsumsi	: Subadi (Koordinator)
	: Tatti Suryati
	: Yayuk Hidayati,SE
	: Siti Khomsatun
	: Mimin
	: Heni Rohaeni
	: Temi Kartini
	: Wiwik

#### 2.1.1.4 Nama Ustad Yang Telah Terdaftar

**Tabel 2. 2 Nama Ustad**

No	Nama	Umu r	Alamat	Keahlian	Madrasah
1	Dzul Fikram	25	Cisitu Indah No.43	Tilawah Qur'an(Tajwid,Makhroj)	Al Hidayah,As h-shobirin
2	Yana	20	Lembah Cisitu Dago No.31 Dago	Fikih(Hadist,Sholat,Ayat),Bera khlak yang baik	Ash- shobirin
3	Sarah	22	Jalan Kampu ng Padi No 12	Tata Cara Islam,Fikih,Tilawah Qur'an	Ash- Shobirin

			RT 3 RW 5		
4	Abu Hamzah Yusuf	26	Jalan Plesiran No 57A Dago Bandun g	Qur'an(Tajwid,Makhroj,Surat Pendek Iqra)	Ash- Shobirin, Al Mukhlis
5	Tsanin Hasanu din	28	Jalan Kampu ng Padi No.23 RT 2 RW 5	Mengkaji Qur'an(Tajwid,Makhroj,Hadist )	Ash- Shobirin

## 2.2 Landasan teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan melakukan penelitian. Landasan teori yang diuraikan merupakan hasil studi literatur, buku-buku, maupun situs internet.

### 2.2.1 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output. Menurut Kamus Kamus Besar Bahasa Indonesia, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk

dieksekusi oleh komputer. Program merupakan kumpulan instruction set yang akan dijalankan oleh pemroses, yaitu berupa software. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini.

Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada instructionset. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh aplikasi ialah program pemroses kata dan *Web Browser*. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app.

Secara historis, aplikasi adalah software yang dikembangkan oleh sebuah perusahaan. App adalah software yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang parallel dengan yang terjadi antar sistem operasi yang dimunculkan. [2]

### **2.2.2 Android**

Sejarah *Android* pada mulanya berasal dari perusahaan bernama Android, Inc. didirikan tempatnya di Palo Alto, California, pada Oktober tahun 2003 oleh Andy Rubin (pendiri Danger), Rich Miner seorang pendiri *Wildfire Communications, Inc.*, Nick Sears seorang mantan VP T-Mobile, dan Chris White seorang kepala desain dan pengembangan antarmuka Web TV untuk mengembangkan mengembangkan sebuah "perangkat seluler pintar yang lebih sadar tentang lokasi dan preferensi penggunanya [3]". Logo *Android* bisa di lihat pada gambar 2.1



**Gambar 2. 1 Logo  
Android**

*Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [3].

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Service (GMS) dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD). Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan *Android* sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM Holdings, *Atheros Communication*, diproduksi oleh Asustek Computer Inc, Garmin Ltd, *Softbank*, Sony Ericson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka *Android*, perangkat *mobile* yang merupakan modifikasi *kernel Linux 2.6*. Sejak *Android* dirilis telah dilakukan berbagai pembaharuan berupa perbaikan *bug* dan penambahan fitur baru [3].

*Android* adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Antarmuka pengguna *Android* umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan untuk digunakan oleh bermacam peranti bergerak (*mobile device*). Hal ini memungkinkan para pengembang menulis kode terkelola (*managed code*) dalam bahasa pemrograman Java, mengontrol peranti via perpustakaan Java yang dikembangkan Google. Telepon pertama yang menggunakan sistem operasi *Android* adalah HTC *Dream* yang dirilis pada 22 Oktober 2008. Adapun versi-versi *Android* yang pernah dirilis adalah sebagai berikut:

1. *Android* versi 1.1
2. *Android* versi 1.5 (*Cupcake*)
3. *Android* 1.6 (*Donut*)
4. *Android* versi 2.0/2.1 (*Eclair*)
5. *Android* Versi 2.2 (*Froyo*)
6. *Android* Versi 2.3 (*Gingerbread*)
7. *Android* Versi 3.0 (*Honeycomb*)
8. *Android* Versi 4.0 (*Ice Cream Sandwich*)
9. *Android* Versi 4.1 (*Jelly Bean*)
10. *Android* Versi 4.4 (*KitKat*)
11. *Android* Versi 5.0 (*Lollipop*)
12. *Android* Versi 6.0 (*Marshmallow*)
13. *Android* Versi 7.0 (*Nougat*)
14. *Android* Versi 8.0 (*Oreo*)

### 2.2.2.1 Arsitektur Android

Arsitektur *Android* Google menggambarkan *Android* seperti sebuah tumpukan *software*. Setiap lapisan dari tumpukan ini terdiri dari beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Dalam paket sistem operasi *Android* terdiri dari beberapa unsur seperti tampak pada gambar 2.6. Secara sederhana arsitektur *Android* merupakan sebuah *kernel Linux* dan sekumpulan pustaka C / C++ dalam suatu *Framework* yang menyediakan dan mengatur alur proses aplikasi [3].

### 2.2.2.2 Android Life Cycle

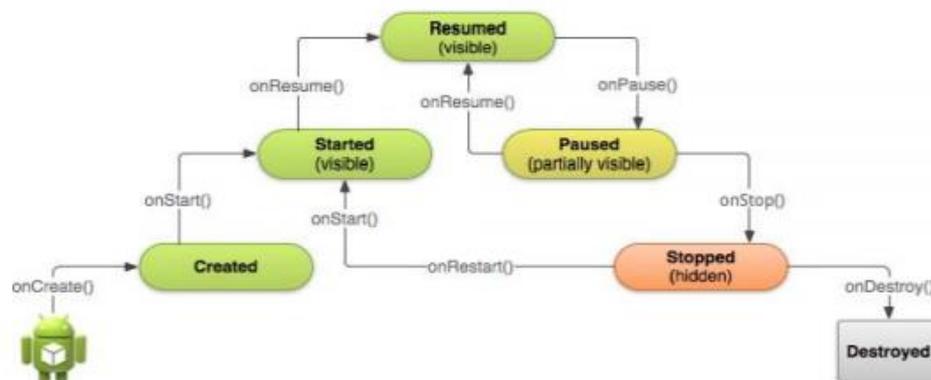
Aplikasi *Android* terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar *file* musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen *Android* yang berbeda seperti ditunjukkan pada, klasifikasi tersebut berdasarkan kelas-kelas dasar *java* yang digunakan bisa dilihat pada gambar 2.2.



**Gambar 2. 2 Komponen Android**

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut akan, tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service*, *Broadcast Receiver*, *Content Provider* [4].

*Android* memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi *main()*, sistem *Android* menjalankan kode dalam *method Activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi *Android* memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web. Hal ini dikarenakan aplikasi *mobile* memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna [4].



**Gambar 2. 3 Siklus Hidup Android**

Terdapat beberapa state dalam siklus hidup *Android* yang terjadi. Siklus Hidup *Android* akan tetapi hanya beberapa dari state tersebut yang menjadi Statis diantaranya :

1. *Resumed*

*Resumed* terjadi ketika aplikasi berjalan setelah *state paused* . State ini akan menjalankan perintah program yang ditulis pada *method onResume()* [4].

2. *Paused*

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apa pun ataupun menampilkan apa pun dalam step ini [4].

3. *Stopped*

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan *service* di *background* [4].

State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cycle callback* berikutnya. Artinya, setelah sistem *OnCreate()* dipanggil, dengan cepat sistem akan memanggil *method onStart()*, kemudian diikuti oleh *onResume()* [4].

Pada penelitian ini versi *Android* yang digunakan adalah *Android* versi 5.0 (nougat) karena mempunyai material merupakan hasil desain terbaru sebagai spesifikasi minimal cocok untuk pengembangan aplikasi karena sudah mempunyai hasil desain *Android* terbaru selain itu. Versi mempunyai kelebihan dalam Peningkatan pada pencarian kontekstual sehingga bisa menentukan lokasi dengan tepat ke tempat aplikasi tertentu dan akan mulai secara tepat. Secara garis besar arsitektur *Android* dapat dijalankan dan digambarkan sebagai berikut : *Application* dan *Widgets* [4].

## 1. *Framework*

*Android* adalah “Open Development Platform” yaitu *Android* menawarkan kepada pengembang atau member kemampuan untuk membangun aplikasi yang inovatif. Pegembang bebas untu mengakses perangkat keras, akses informasi *resource*, menjalankan *service background*, mengatur alarm, dan menambahkan status notifikasi. Komponen-komponen yang termasuk di dalam *applications frameworks*

adalah sebagai berikut:

- a. *Views*
- b. *Content provider*
- c. *Resource manager*
- d. *Notification manager*
- e. *Activity manager*

## 2. *Libraries*

*Libraries* adalah layer dimana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

## 3. *Android Runtime*

Layer yang membuat aplikasi *Android* dapat djalankan dimana dalam prosesnya menggunakan Implementasi *Linux. Dalvik virtual machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*.

## 4. *Linux Kernel*

*Linux Kernel* adalah layer di mana inti dari *operating system* dari *Android* itu berada. Berisi *file-file system* yang mengatur sistem *processing*, *memory*, *recource*, *drivers*, dan sistem-sistem operasi *Android* lainnya [4].

### 2.2.3 Hybrid

*Hybrid apps* merupakan sebutan untuk aplikasi *smartphone* yang dibuat dengan teknologi web. Dikatakan *hybrid apps* karena aplikasi tersebut merupakan kombinasi antara aplikasi web dan aplikasi *native*. Aplikasi web merupakan aplikasi yang berada di server web dan diakses menggunakan web browser, sedangkan aplikasi *native* merupakan aplikasi yang dibuat dengan bahasa asli sistem operasi. *Hybrid apps* juga menggabungkan keuntungan yang ada pada aplikasi web yang multi *platform* dan keuntungan *native apps* yang mampu mengakses fitur-fitur *native* pada *smartphone* [5].

Dibanding dengan aplikasi *native*, aplikasi *hybrid* memiliki banyak keunggulan. Keunggulan aplikasi *hybrid* dibandingkan dengan aplikasi *native* yaitu:

1. Cukup membuat aplikasi satu kali untuk digunakan oleh berbagai sistem operasi *smartphone*.
2. Menggunakan bahasa yang mudah dipahami yaitu CSS3, HTML5 dan javascript yang sudah sangat dikenal bagi web *programmer*.
3. Biaya pengembangan lebih rendah.
4. Waktu yang diperlukan untuk pengembangan aplikasi lebih sedikit.

Adapun bila dibandingkan dengan aplikasi web, aplikasi *hybrid* memiliki banyak keunggulan sebagai berikut:

1. Proses loading dan kinerja lebih cepat.
2. Dapat dijalankan secara online maupun *offline*.
3. Dapat mengakses fitur-fitur perangkat keras *smartphone*, seperti kamera, galeri, kontak, dan sebagainya.
4. Dapat di *upload* ke appstore.

#### 2.2.3.1 Node.js

**Node.js** merupakan perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintak bahasa pemrograman JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi *client* / browser saja, maka Node.js ada untuk melengkapi peran JavaScript

sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server web seperti *Apache* atau *Nginx* [5].

Kelebihan Memakai Node.js :

1. Node.js menggunakan bahasa pemrograman JavaScript yang merupakan bahasa pemrograman yang paling populer dan banyak dikenal oleh masyarakat luas
2. Node.js mampu menangani ribuan koneksi bersamaan dengan penggunaan resource minimum untuk setiap prosesnya
3. Node.js sangat diandalkan terutama untuk membuat aplikasi real-time
4. Node.js adalah *project open source*, sehingga siapa pun dapat melihat struktur kode dan juga dapat berkontribusi untuk pengembangannya
5. Penggunaan JavaScript di sisi server dan juga client meminimalisir ketidakcocokan antar dua sisi lingkungan pemrograman, seperti terkait komunikasi data yang mana menggunakan struktur JSON yang sama di kedua sisi, validasi form yang sama yang dapat dijalankan di sisi server dan client, dan sebagainya.
6. Database NoSQL seperti MongoDB dan CouchDB mendukung langsung Javascript sehingga *interfacing* dengan database ini akan jauh lebih mudah.
7. Node.js memakai V8 yang selalu mengikuti perkembangan standar ECMAScript (nama standar resmi dari JavaScript, Namun JavaScript yang lebih dikenal dalam implementasinya), sehingga tidak perlu ada kekhawatiran bahwa browser tidak mendukung fitur-fitur di Node.js.

### 2.2.3.2 NPM

Salah satu tool yang akan sering digunakan dalam Nodejs adalah NPM (*Node Package Manager*) [5]. NPM sudah otomatis terinstal saat menginstal Nodejs.

NPM berfungsi untuk:

1. Membuat *Project* Baru;
2. Menginstal modul atau *library*;
3. Menjalankan skrip *command line*.

### 2.2.3.3 Ionic Framework dan Apache Cordova

*Framerwork* merupakan kumpulan kode program yang siap pakai sehingga ketika membuat aplikasi tidak perlu dimulai dari nol. Jika dalam aplikasi web dikenal dengan Bootstrap sebagai framework populer yang menyediakan berbagai desain siap pakai dan terintegrasi dengan jQuery. Namun tidak cocok ketika digunakan untuk aplikasi hybrid [5].

Ionic merupakan kerangka ponsel HTML5 dengan fokus pada kinerja yang memanfaatkan akselerasi hardware dan tidak memerlukan pihak ketiga seperti JS library. Ionic bekerja berbarengan dengan Angular.js untuk membangun sebuah aplikasi interaktif. Aplikasi hybrid pada dasarnya ialah website kecil yang berjalan di shell browser, sebuah aplikasi yang mempunyai akses ke lapisan platform asli dari sebuah *device*. Aplikasi hybrid mempunyai sangat banyak manfaat jika dibandingkan dengan aplikasi native, khususnya dalam hal mendukung platform dan kecepatan pengembangan, Ionic dilengkapi dengan elemen UI mobile dan layout yang mirip dengan SDK asli pada ios atau android. Ionic memerlukan Apache Cordova untuk menjalankan aplikasi [6].

Dalam pengembangan aplikasi hybrid, juga banyak pihak yang menawarkan framework dengan segala kemampuannya yang akan sangat membantu pembuatan aplikasi menjadi lebih mudah. Salah satu yang paling populer dan banyak digunakan oleh para programmer di dunia yaitu framework Ionic. Ionic terintegrasi dengan Angular sehingga sangat cocok digunakan untuk membuat aplikasi yang berbasis Angular.

Sejak versi 2, Angular sudah banyak berbeda dibanding dengan versi pertamanya. Untungnya, Ionic merupakan framework yang selalu update dan mengikuti perkembangan Angular.

Beberapa keunggulan yang dimiliki Ionic dibanding dengan para pesaingnya diantaranya sebagai berikut:

1. Ionic berbasis *open source* sehingga bebas mengembangkan aplikasi baik untuk keperluan sendiri maupun aplikasi komersial menggunakan Ionic.
2. Ionic terintegrasi dengan Angular, bahkan gaya pengkodeannya juga persis mengikuti gaya pengkodean Angular.
3. Ionic memiliki UI default yang sangat cantik dan mudah untuk di *customize*.
4. Ionic dapat diinstall melalui GUI maupun CLI, serta menyediakan *Tool* dan *Service* yang memudahkan pengguna Ionic bagi programmer.
5. Ionic dapat diintegrasikan dengan mudah dengan fitur-fitur smartphone seperti kamera, galeri, kontak, geolocation, dan sebagainya.
6. Tim pengembang Ionic sangat aktif di media sosial.
7. Ionic juga memiliki komunitas yang sangat besar aktif di media sosial.

Apache Cordova adalah framework untuk membangun sebuah aplikasi pada berbagai macam *platform* seperti Android, Blackberry, Iphone ataupun Windows Phone menggunakan HTML5, JQuery, JQuery Mobile dan CSS3. Membangun aplikasi untuk device yang berbeda seperti Android, iPhone, Windows Mobile dan lainnya dibutuhkan framework dan bahasa pemrograman yang berbeda, seperti pada Android menggunakan bahasa pemrograman Java, blackberry dengan bahasa pemrograman Java, iPhone dengan Basic C dan Windows Phone dengan C# [6].

#### **2.2.4 PHP**

PHP dikenal secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side* [7]. PHP diciptakan oleh Rasmus Lerdorf pertama

kali tahun 1994. Pada awalnya PHP adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, nama bahasa ini diubah menjadi "*PHP: Hypertext Preprocessor*" dengan singkatannya "PHP".

Beberapa kelebihan PHP sebagai bahasa pemrograman web antara lain sebagai berikut.

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan di mana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem.

Berikut adalah contoh sintaks bahasa pemrograman PHP dapat dilihat digambar 2.4 :

```
<!DOCTYPE html>
<html>
  <head>
    <title><?php echo "Belajar PHP" ?></title>
  </head>
  <body>
    <?php
      echo "saya sedang belajar PHP<br>";
      echo "<p>Belajar PHP hingga jadi master</p>";
    ?>
  </body>
</html>
```

**Gambar 2. 4 Bahasa Pemrograman PHP**

### 2.2.5 Database

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari field atau kolom. Struktur file yang menyusun sebuah database adalah data Record dan Field [8].

1. Data adalah satu kesatuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan di dalam suatu file database.
2. Record adalah data yang isinya merupakan satu kesatuan seperti NamaUser dan Password dinamakan satu record. Setiap record diberi nomor urut yang disebut nomor record (Record Number).
3. Field adalah sub bagian dari Record. Dari contoh isi record di atas, maka terdiri dari 2 field, yaitu: field NamaUser dan Password.

### 2.2.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*), database ini *multithread, multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus yang bersifat khusus.

Kekuatan MySQL tidak ditopang oleh sebuah komunitas, seperti Apache, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing-masing, tetapi MySQL didukung penuh oleh sebuah perusahaan profesional dan komersial, yakni MySQL AB dari Swedia.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan mudah secara otomatis [9].

Berikut adalah contoh sintaks sederhana pada MySQL:

1. Membuat database

#### **Tabel 2. 3 Membuat Database**

```
CREATE DATABASE rental;
```

2. Melihat database

#### **Tabel 2. 4 Melihat Database**

```
SHOW DATABASES;
```

3. Membuat tabel

#### **Tabel 2. 5 Membuat Tabel Database**

```
CREATE TABLE customer (  
    id_customer VARCHAR (8) NOT NULL,  
    nama_customer VARCHAR (20) NOT NULL,  
    tgl_rental DATETIME NOT NULL,  
    tgl_kembali DATETIME NOT NULL,  
    PRIMARY KEY (id)  
);
```

4. Melihat tabel

#### **Tabel 2. 6 Melihat Tabel Database**

```
SHOW TABLES;
```

5. Menghapus tabel

#### **Tabel 2. 7 Menghapus Tabel Database**

```
DROP TABLE customer;
```

6. Menghapus database

#### **Tabel 2. 8 Menghapus Database**

```
DROP DATABASE rental;
```

Dalam menggunakan *database* MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL dapat dikategorikan menjadi 3 sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*). Berikut adalah pemaparan dari setiap kategori:

1. DDL (*Data Definition Language*)

Data Definition Language yang kalau di singkat DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini database dan table. Beberapa perintah dasar yang termasuk DDL ini antara lain :

- a. CREATE berfungsi untuk membuat database baru, tabel baru, view baru dan kolom.
- b. ALTER berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
- c. DROP berfungsi untuk menghapus database dan tabel.
- d. TRUNCATE berfungsi untuk menghapus semua catatan dari tabel.
- e. COMMENT berfungsi untuk menambahkan komentar pada data.
- f. RENAME berfungsi untuk mengubah nama objek.

## 2. DML (*Data Manipulation Language*)

Kategori selanjutnya adalah Data Manipulation Language (DML). DML ialah perintah yang digunakan untuk mengelola/memanipulasi data dalam database. Terdapat beberapa perintah DML pada MySQL sebagai berikut :

- a. SELECT berfungsi untuk mengambil/menampilkan data dari database.
- b. INSERT berfungsi untuk memasukkan data ke dalam tabel.
- c. UPDATE berfungsi untuk memperbaiki data dalam tabel.
- d. DELETE berfungsi untuk menghapus data dari tabel.
- e. CALL berfungsi untuk memanggil subprogram PL / SQL atau Java.

- f. EXPLAIN PLAN berfungsi untuk menjelaskan jalur akses ke data.
- g. LOCK TABLE berfungsi untuk mengunci tabel.

### 3. DCL (*Data Control Language*)

Kategori terakhir adalah Data Control Language (DCL). DCL ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah DCL pada MySQL sebagai berikut :

- a. GRANT berfungsi untuk memberikan hak akses pengguna ke database.
- b. REVOKE berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah GRANT.

## 2.2.7 Web Service

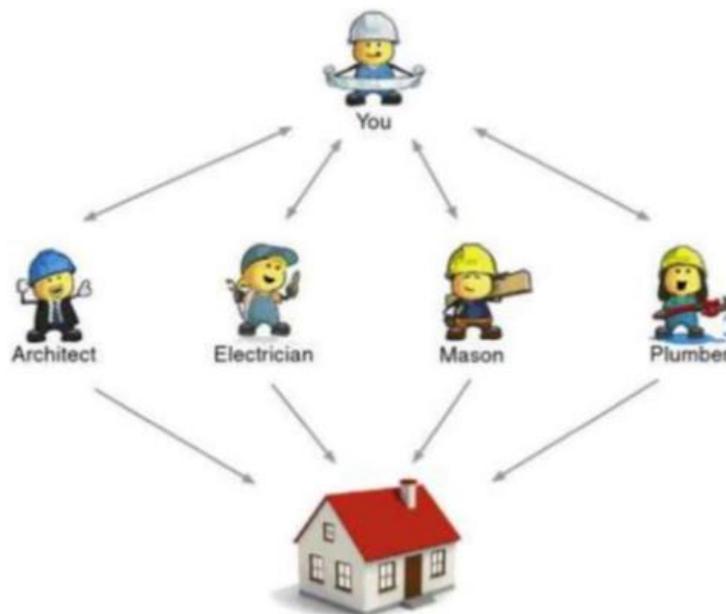
W3C mendefinisikan web service sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan i aplikasi yang lain dengan menggunakan message berbasis XML atau JSON melalui protokol internet. Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, menyediakan method yang dapat diakses oleh network. Web Service juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda. Beberapa karakteristik dari web service adalah:

1. Message-Based
2. Standards-Based
3. Programming Language Independent
4. Platform-Neutral

Beberapa key standard didalam web service adalah: JSON, XML, SOAP, WSDL and UDDI.

### 2.2.8 API (Application Programming Interface)

*Application Programming Interface* (API) merupakan *software interface* yang terdiri dari kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu bangunan atau rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan *software* yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari *software* tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut [10]. Analogi API dapat dilihat pada gambar 2.5 berikut:



**Gambar 2. 5 Analogi API**

*Sumber Gambar : Jurnal Teknologi Terpadu [10].*

### 2.2.9 Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut. Bahasa pemrograman dapat dikelompokkan dalam berbagai macam sudut pandang [11]. Salah satu pengelompokan bahasa pemrograman adalah pendekatan dari notasi bahasa pemrograman tersebut, apakah lebih dekat ke bahasa mesin atau bahasa manusia. Dengan cara ini, bahasa pemrograman dapat dikelompokkan menjadi tiga yakni :

1. *Low Level Language*

Yaitu bahasa pemrograman tingkat rendah, merupakan bahasa pemrograman generasi pertama. Bahasa pemrograman generasi pertama ini merupakan bahasa pemrograman yang sangat sulit dimengerti karena instruksinya menggunakan bahasa mesin, dan hanya dimengerti oleh pembuatnya saja karena programnya berupa kode-kode mesin. Bahasa pemrograman *level* rendah ini pertama kali muncul atau digunakan mulai sekitar tahun 1945. Ketika itu untuk membuat dan menjalankan suatu program dibutuhkan waktu yang lama dan itu pun sering dijumpai kesalahan-kesalahan. Program sangat sulit dibaca dan ditulis, sehingga pada saat itu sangat sedikit orang yang tertarik untuk menjadi *programmer*.

2. *Middle Level Language*

Yaitu bahasa pemrograman tingkat menengah, yang merupakan bahasa pemrograman generasi ke dua. Dalam bahasa ini seorang *programmer* sudah mulai bisa menggunakan bahasa sehari-hari, walaupun masih banyak susah dimengerti juga. Banyak perintah-perintah yang menggunakan inisial atau singkatan-singkatan seperti “MOV” yang berarti “MOVE” (pindah), “STO” yang berarti (STORE) dan lain-lain. Bahasa pemrograman yang tergolong dalam *middle* level ini adalah *assembler*.

3. *High Level Language*

Yaitu bahasa pemrograman tingkat tinggi, yang merupakan bahasa pemrograman generasi ke tiga dan selanjutnya. Ciri-cirinya yaitu bahasa pemrograman ini sudah terstruktur dengan baik, mudah dimengerti karena sudah menggunakan bahasa sehari-hari (bahasa Inggris tapi ya), Bahasa pemrograman inilah bahasa pemrograman yang sekarang ini kita kenal, seperti C, C++, JAVA, PHP, Visual Basic, Pascal, ORACLE, MS-SQL, Python, XML dan lain-lain sebagainya. Sudah begitu beragam dan bermacam-macam jenis sesuai karakter struktur dan kegunaannya.

#### **2.2.10 SuperSaas**

Perangkat Lunak sebagai Layanan atau SaaS di mana aplikasi berbasis web yang berfungsi penuh dapat digunakan seperti Google drive, Zoho.com, Kizoa, Wordle, Prezi dan banyak lagi; Platform sebagai Layanan atau PaaS yang merujuk pada penggunaan kerangka kerja berbasis web untuk membangun perangkat lunak seperti penemu Aplikasi MIT dan Supersaas.com untuk membuat modul reservasi, Infrastruktur sebagai Layanan atau IaaS yang kadang-kadang digambarkan sebagai ruang perangkat keras untuk disewakan. Hosting situs web gratis online seperti Wordpress.com, Weebly.com, dan Wix.com mungkin termasuk dalam kategori ini [12].

#### **2.2.11 Profile Matching**

Dalam proses profile matching secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk karyawan menempati posisi tersebut. Adapun sistem program yang dibuat adalah software profile matching yang berfungsi sebagai alat bantu untuk mempercepat proses matching antara profil

jabatan (soft kompetensi jabatan) dengan profil karyawan (soft kompetensi karyawan) sehingga dapat memperoleh informasi lebih cepat, baik untuk mengetahui gap kompetensi antara jabatan dengan pemegang jabatan maupun dalam pemilihan kandidat yang paling sesuai untuk suatu jabatan (ranking kandidat) [13]

Langkah-langkah pada metode *profil matching* yaitu

Menentukan variable-variabel pemetaan Gap kompetensi Menentukan aspek-aspek yang akan digunakan dalam memproses nilai karyawan.

2. Menghitung hasil pemetaan Gap kompetensi Yang dimaksud dengan *Gap* disini adalah beda antara profil karyawan dengan profil standar yang diharapkan atau dapat ditunjukkan pada rumus di bawah ini :

$$Gap = Profil\ karyawan - Profil\ standar...(1)$$

Profil karyawan yaitu nilai-nilai yang diperoleh dari karyawan sedangkan profil standar yaitu nilai standar yang ditentukan terlebih dahulu. Setelah diperoleh *gap* pada masing-masing karyawan, setiap profil karyawan diberi bobot nilai dengan patokan

Kemudian setiap aspek dikelompokkan menjadi 2 kelompok, yaitu kelompok *Core Factor* dan *Secondary Factor*. Perhitungan *core factor* ditunjukkan menggunakan rumus diwah ini

$$NCF = \frac{\sum NC}{IC} \dots\dots\dots(2)$$

NCF = nilai rata-rata *core factor*

NC = Jumlah total nilai *core factor*

IC = Jumlah item *core fator*

Sementara untuk perhitungan *secondary factor* bisa ditunjukkan dengan rumus berikut :

$$NSF = \frac{\sum}{IS} \dots\dots\dots (3)$$

NSF = nilai rata-rata *secondary factor*

NS = Jumlah total nilai *Secondary factor*

IS = Jumlah item *Secondary fator*

Setelah perhitungan *Core factor* dan *Secondary factor*, kemudian menghitung Nilai total berdasarkan dari persentase dari *core* dan *secondary* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Contoh perhitungan bisa dilihat pada rumus dibawah ini.

$$(x)\%NCF(\text{Nilai Rata-rata core factor}) + (x)\%NSF(\text{Nilai Rata-rata secondary factor})=N(\text{Total dari aspek})\dots\dots\dots(4)$$

Keterangan :

(x)% : Nilai Persen yang Diinputkan

Terakhir perhitungan Ranking, perhitungan tersebut bisa ditunjukkan dengan rumus dibawah ini

**Tabel 2. 9 Hasil Dan Pembahasan**

<b>Ranking = (x)%N1 + (x)%N2+</b> <b>(x)%N3.....(5)</b>			
Keterangan :			

$N1, N2, N3$	: Nilai aspek yang sudah dihitung total			
$(x)\%$	: Nilai persen yang diinputkan.			
<b>• Hasil dan Pembahasan</b>				
<b>Nama</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Nurul	3,5	1.000.000	6	4
Reza	3,7	1.500.000	4	4
Riskiyati	3,7	1.000.000	5	6

Dari data diatas dibuatlah GAP seperti berikut:

**Tabel 2. 10 Gap**

No MHS	1	2	3	4
M1	3	4	4	2
M2	4	3	3	2
M3	4	4	4	3
Gap	3	3	3	2
M1	0	1	1	0
M2	1	0	0	0
M3	1	1	1	1

- Pembobotan**

Setelah diperoleh gap pada masing-masing. Setiap profil mahasiswa diberi bobot nilai dengan patokan tabel bobot nilai gap seperti yang terlihat dalam tabel berikut:

**Tabel 2. 11 Pembobotan**

No	Selisi h	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih(kompetensi sesuai dengan yang dibutuhkan)
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level
4	2	3,5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kekurangan 2 tingkat/level
6	3	2,5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kekurangan 3 tingkat/level
8	4	1,5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kekurangan 4 tingkat/level

Dengan demikian, setiap mahasiswa akan memiliki tabel bobot seperti:

**Tabel 2. 12 Tabel Bobot**

No MHS	1	2	3	4
M1	5	4,5	4,5	5

M2	4,5	5	5	5
M3	4,5	4,5	4,5	4,5

- **Perhitungan Core Dan Secondary Factor**

Yang menjadi *core factor* adalah IPK, dan Penghasilan orang tua, maka yang secondary factor tanggungan orang tua dan semester.

**Untuk M1**

$$NC = \text{---} = 4,75$$

$$NS = \text{---} = 4,75$$

**Untuk M2**

$$NC = \text{---} = 4,75$$

$$NS = \text{---} = 5$$

**Untuk M3**

$$NCF = \text{---} = 4,5$$

$$NSF = \text{---} = 4,5$$

- **Pehitungan Nilai Total**

Dari hasil perhitungan setiap aspek, berikutnya dihitung nilai total berdasarkan presentase dari core dan secondary yang diperkirakan berpengaruh.

**Untuk M1**

$$N = 60 \% (NCF) + 40 \% (NSF)$$

$$N = (0,6 \times 4,75) + (0,4 \times 4,75)$$

$$N = 2,85 + 1,9$$

$$N=4,75$$

**Untuk M2**

$$N=60 \% (NCF)+40 \% (NSF)$$

$$N= (0,6 \times 4,75) + (0,4 \times 5)$$

$$N=2,85 + 2$$

$$N=4,85$$

**Untuk M3**

$$N=60 \% (NCF)+40 \% (NSF)$$

$$N= (0,6 \times 4,5) + (0,4 \times 4,5)$$

$$N=2,7+1,8$$

$$N=4,5$$

Dengan demikian urutan yang berhak mendapatkan beasiswa adalah:

**Tabel 2. 13 Perhitungan Nilai Total**

NO MHS	Ranking
M2	1
M1	2
M3	3

- **KESIMPULAN**

Beasiswa sangat diharapkan oleh mahasiswa, karena beasiswa bertujuan untuk meringankan beban biaya kuliah.

Dalam pemilihan beasiswa dapat digunakan Metode Profile Matching.

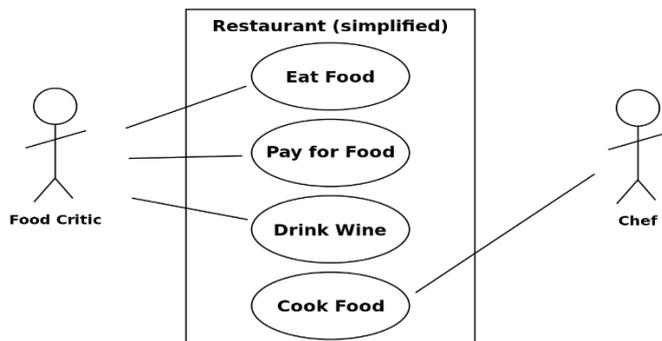
Metode Profile Matching merupakan metode mencocokkan profile mahasiswa dengan profile yang diharapkan untuk penerima beasiswa. Metode ini menggunakan perankingan untuk merekomendasikan sebuah keputusan.

### **2.2.13 Unified Modeling Language(UML)**

Diagram *use case* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar, dan menjelaskan sistem secara fungsional yang terlihat *user*. Biasanya dibuat pada awal pengembangan. *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua feature yang ada pada sistem. Sebuah *use case* dapat *use case* memanggil (*include*) fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang memanggil (*include*) dieksekusi secara normal. Sebuah *use case* dapat dipanggil (*include*) oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng-extend *use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain[14].

Aplikasi renovin menggunakan UML dikarenakan Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Perancangan komponen antarmuka terintegrasi dengan perancangan PL dan struktur data

### 2.2.13.1 Diagram Use case (Use case diagram)



**Gambar 2. 6 Contoh Use Case**

Diagram *use case* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. *Use case* adalah interaksi antara *actor eksternal* dan sistem digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Diagram *use case* menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Elemen-elemen yang ada pada diagram *use case* adalah aktor, *use case*, dan hubungan ketergantungan, generalisasi dan asosiasi[14].

a. Aktor

Aktor adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu siapa atau apa yang menggunakan sistem. Aktor berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan. *Use case* selalu diawali oleh aktor yang mengirim pesan[14].

b. *Use case*

*Use case* adalah cara spesifikasi penggunaan sistem oleh aktor. *Use case* menspesifikasikan perilaku sistem atau bagian sistem dan merupakan deskripsi sekumpulan sekuen aksi termasuk varian-varian yang dilakukan sistem untuk memproduksi hasil atau nilai ke actor[14].

c. Hubungan Antar *Use case*

Keterhubungan antar *use case* dengan *use case* lain berupa generalisasi antara *use case*, yaitu include dan extend. *Include* adalah perilaku *use case*

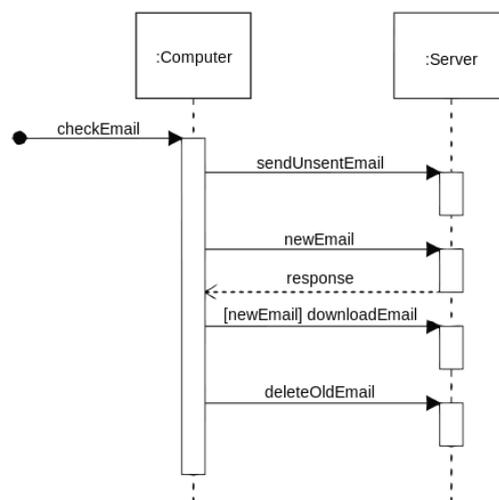
merupakan bagian dari *use case* yang lain. Sedangkan *extend* adalah perilaku *use case* merupakan perilaku *use case* yang lain.

Adapun level dalam usecase diantara lain sebagai berikut :

Sea level khususnya mewakili sebuah interaksi diskrit antara aktor utama dan sistem.

Fish level *use case* yang ada disana hanya karena telah masuk *use case sea level*[14].

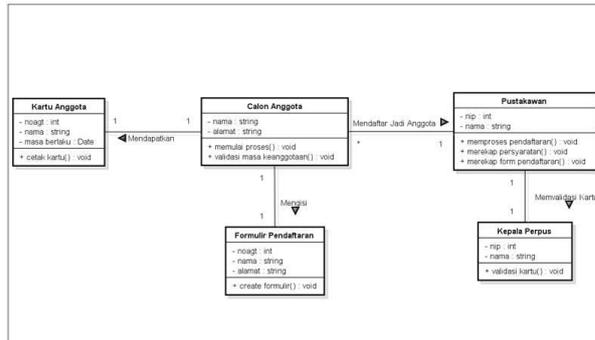
### 2.2.13.2 Diagram Sekuensial (Sequence Diagram)



**Gambar 2. 7 Sequence Diagram**

Pada penelitian ini sekuensial diagram digunakan untuk memodelkan skenario penggunaan. Penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Diagram sekuean menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan . Diagram sekuensial ini juga akan manggambarkan *behavior* yang ada pada aplikasi. Oleh sebab itu diagram sekuensial ini digunakan untuk menggambarkan bagaimana antiitas dalam sistem berinteraksi[14].

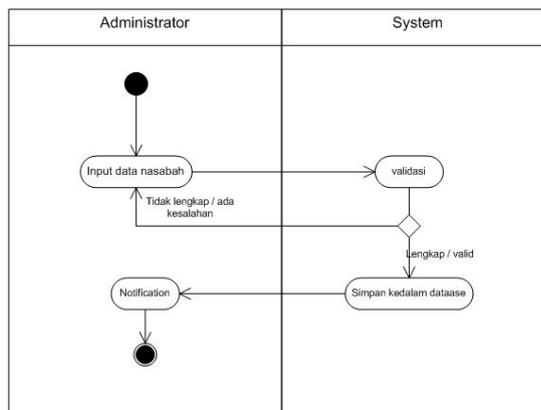
### 2.2.13.3 Diagram Kelas (Class Diagram)



**Gambar 2. 8 Class Diagram**

*Class diagram* adalah model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara *class*. *Class diagram* mirip ER-Diagram pada perancangan *database*, bedanya pada ER-diagram tidak terdapat operasi/*methode* tapi hanya atribut. *Class* terdiri dari nama kelas, atribut dan operasi/*methode*[14].

### 2.2.13.4 Digram Aktivitas (Activity diagram)



**Gambar 2. 9 Activity Diagram**

Diagram aktivitas pada umumnya adalah diagram *flowchart* yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain. Diagram aktivitas berfokus pada aktivitas-aktivitas, potongan-potongan dari proses yang boleh jadi berkorespondensi dengan metode – metode atau fungsi-fungsi anggota dan pengurutan dari aktivitas-aktivitas ini. Elemen-elemen yang ada pada diagram aktivitas adalah *activity state* dan *action state*, transisi,

dan objek. Pada penelitian ini diagram aktifitas digunakan untuk menggambarkan alur sistem dari awal hingga proses berakhir[14].

