

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Yayasan

Tahap tinjauan perusahaan ini merupakan peninjauan terhadap tempat penelitian studi kasus yang dilakukan pada Donini Sentra Binatu Bandung.

2.1.1 Sejarah Yayasan

Pemuda hijrah (Shift) berdiri pada tahun 2015, shift adalah gerakan dakwah yang berada di kota Bandung dan konsentrasi dakwahnya adalah pada anak-anak muda di Bandung. Pemuda Hijrah sudah menjadi trendsetter dikalangan anak muda kota Bandung, ini terbukti Pemuda Hijrah memiliki followers Instagram yang menyentuh angka 1,1 juta.

2.1.2 Logo Yayasan

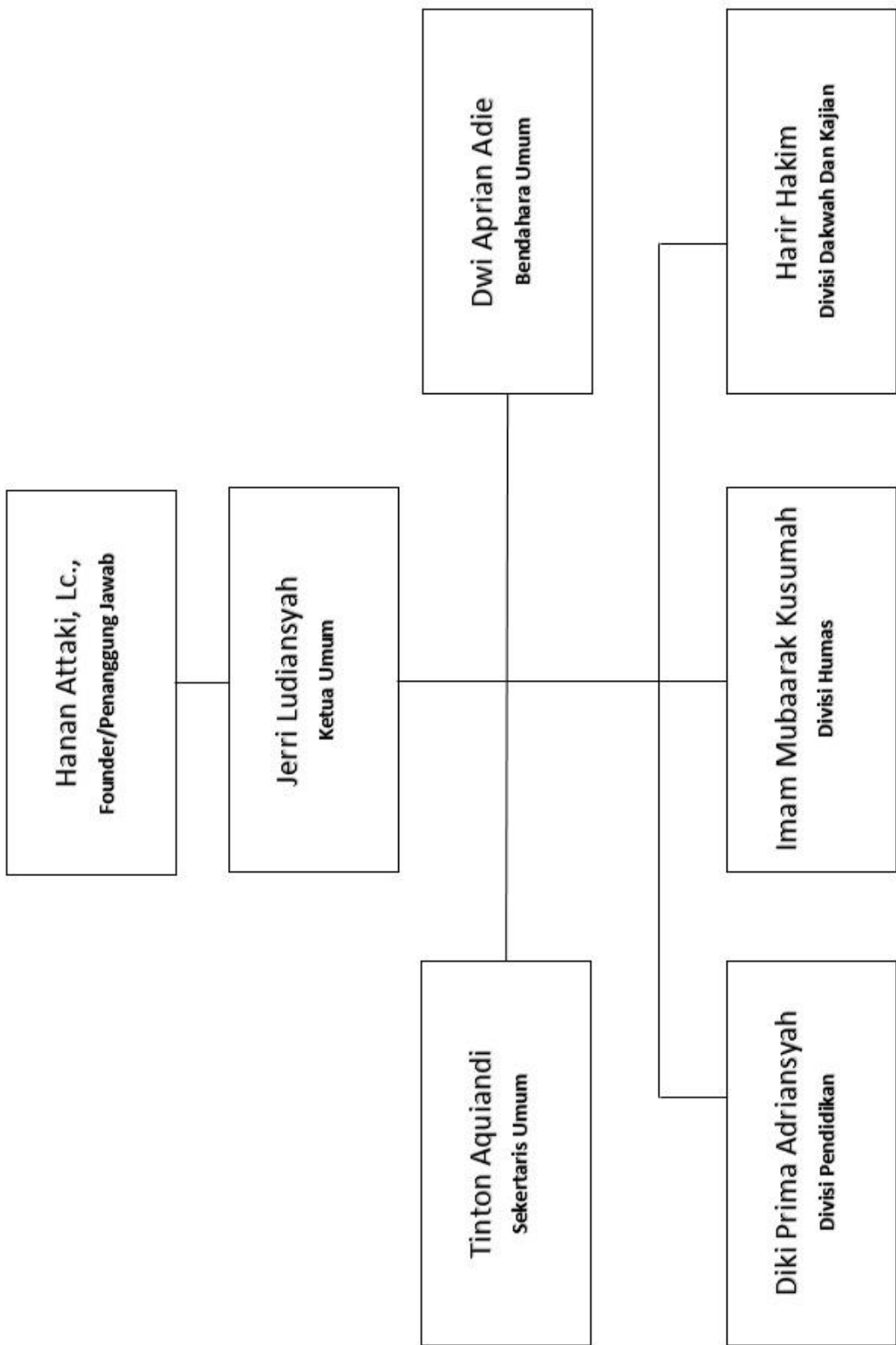
Berikut ini merupakan logo dari Shift Gerakan Pemuda Hijrah dilihat pada Gambar 2.1 :



Gambar 2. 1 Logo Shift Gerakan Pemuda Hijrah

2.2 Struktur Organisasi Yayasan

Struktur organisasi merupakan gambaran secara grafik yang terbentuk struktur kerja dari suatu struktur organisasi. Struktur organisasi hanya dapat menunjukkan hubungan wewenang yang formal dan tidak dapat menggambarkan seberapa besar wewenang, tanggung jawab dan deskripsi pekerjaan yang terperinci. Untuk menjelaskan struktur organisasi yang menjadi sasaran kegiatan penelitian, maka struktur organisasi pada Donini Sentra Binatu dapat dilihat pada Gambar 2.2 :



2.3 Landasan Teori

Dalam penelitian skripsi ini digunakan landasan teori yang berisi teori – teori yang bisa dijadikan dasar dan acuan dalam pembangunan aplikasi pemuda hijrah. Android

Android adalah system operasi untuk handphone yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc. pendatang baru yang membuat peranti lunak untuk handphone. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi. [4]

Pada saat perilis perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat software dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mobile Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

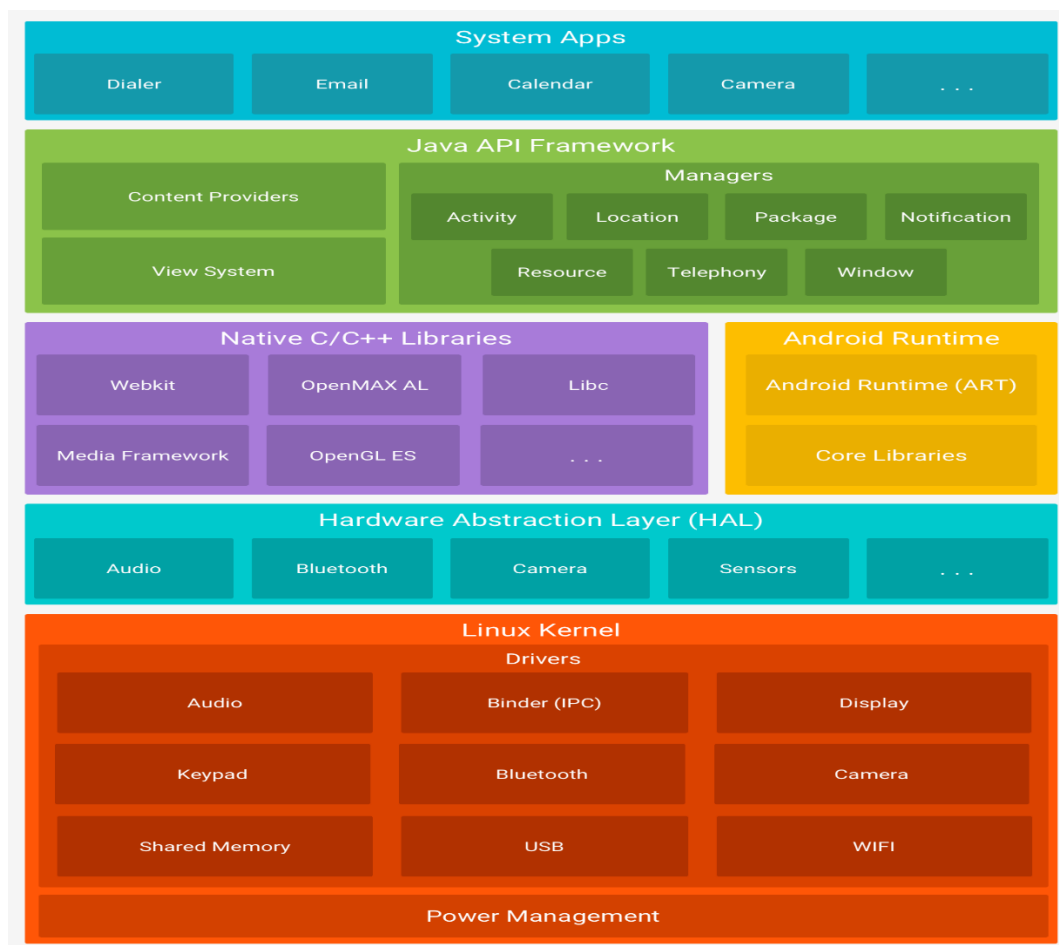
Android merupakan sistem operasi yang berkembang dengan pesat, namun tidak menjadikannya sistem operasi yang sempurna ada beberapa kekurangan dari sistem operasi Android diantaranya Android terkesan rumit, karena mempunyai banyak sekali widget maupun aplikasi dengan banyak pengaturan sehingga pengguna harus banyak belajar mengenai Android, selain itu Android yang merupakan sistem operasi terbuka sehingga pengguna dapat memasang aplikasi di luar toko aplikasi yang ditawarkan oleh perangkat Android tersebut sehingga sangat

rentan terkena ancaman malware atau virus. Tidak semua perangkat Android dapat langsung memperbarui sistem operasi terbaru, karena produsen smartphone lebih mementingkan produk baru untuk diberi sistem operasi yang terbaru, dibanding dengan memberi pemberitahuan tentang update sistem operasi terbaru sehingga membutuhkan waktu lama untuk memperbarui system operasi bagi beberapa perangkat.

2.3.1 Arsitektur Android

2.3.2.1 Arsitektur Platform

Android adalah tumpukan perangkat lunak berbasis Linux sumber terbuka yang dibuat untuk berbagai perangkat dan faktor bentuk. Diagram berikut menunjukkan komponen besar dari platform Android.[5]



Gambar 2. 2 Arsitektur Android

2.3.2.2 Linux Kernel

Fondasi platform Android adalah kernel Linux. Sebagai contoh, Android Runtime (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.

2.3.2.3 Hardware Abstraction Layer (HAL)

Hardware Abstraction Layer menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras tersebut.

2.3.2.4 Android Runtime

Untuk perangkat yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap Android Runtime (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format bytecode yang didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke bytecode DEX, yang dapat berjalan pada platform Android.

Beberapa fitur utama ART mencakup:

- Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT)
- Pengumpulan sampah (GC) yang dioptimalkan

- Dukungan debug yang lebih baik, mencakup profiler sampling terpisah, pengecualian diagnostik mendetail dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

2.3.2.5 Pustaka C/C++ Asli

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda.

Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka platform asli langsung dari kode asli.

2.3.2.6 Kerangka Kerja Java API

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

- Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan

- Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout
- Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status
- Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum
- Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri

Developer memiliki akses penuh ke API kerangka kerja yang sama dengan yang digunakan oleh aplikasi sistem Android.

2.3.2.7 Aplikasi Sistem

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama platform tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

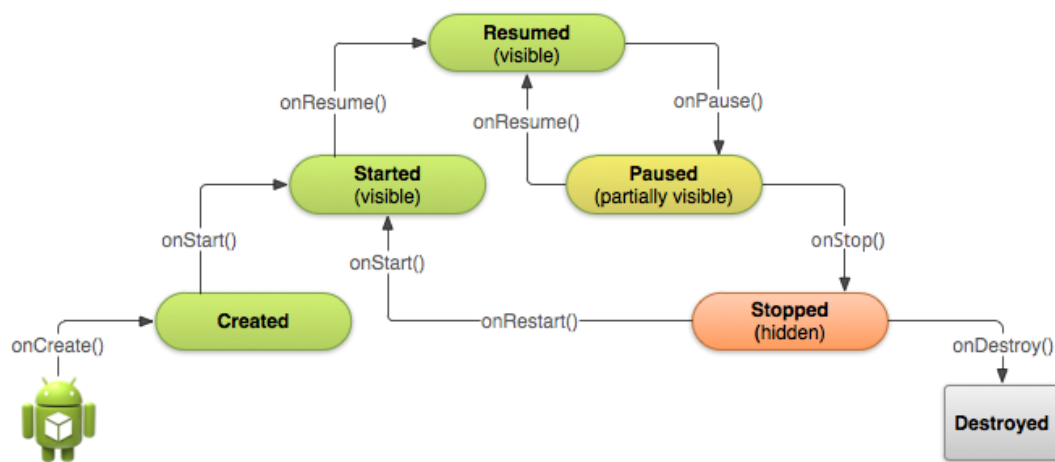
Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang Anda tetapkan.

2.3.2 Siklus Hidup Android

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 [4].

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service*, *BroadcastReceiver*, *ContentProvider*.

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 ataupun web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang cukup 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.



Gambar 2. 3 Sirklus Hidup Android

diilustrasikan pada gambar 2.3 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()`.

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 apapun ataupun menampilkan apapun dalam state ini .

3. Stopped

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground. State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *life cyclecallback* berikutnya. Artinya, setelah sistem *OnCreate()* dipanggil, dengan cepat sistem akan memanggil method *OnStart()*, kemudian diikuti oleh *onResume()*.

2.3.3 Fitur Android

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah [6]:

- *Multi-proses* dan *App Widgets*

Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara *multitasking*. Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan *game*, proses lain dapat berjalan di belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan.

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widgets dapat menjalankan proses request seperti musik streaming atau mendeteksi suhu ruangan secara *background*.

Multi-proses dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika pengguna tidak benar.

- *Touch Gestures* dan *Multi-touch*

Touchscreen adalah *user interface* intuitif yang digunakan banyak *smartphone* di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena

cukup dengan menggunakan jari tangan. *ulti-touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu. Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu, pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut.

- *Hard dan Soft Keyboard*

Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi.

2.3.4 Android SDK

Android SDK adalah tools *API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone atau Smartphone. Beberapa fitur-fitur android yang paling penting adalah [6] :

- *Framework* : Aplikasi yang mendukung pengganti komponen dan reusable.
- *Dalvik Virtual Machine* dioptimalkan untuk perangkat mobile.
- *Integrated Browser* berdasarkan engine open source WebKit.
- Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi hardware)
- SQLite untuk penyimpanan data.
- Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung hardware)
- Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
- Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware)

- Lingkungan *Development* yang lengkap dan termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin.

2.3.5 JAVA

Inovasi bahasa komputer dimotivasi oleh dua faktor: perbaikan dalam seni pemrograman dan perubahan dalam lingkungan komputasi, tidak terkecuali Java. Dibangun di atas warisan yang kaya dari C dan C++, Java menambahkan perbaikan dan fitur yang mencerminkan keadaan seni dalam pemrograman saat ini. Menanggapi munculnya lingkungan online, Java menawarkan fitur yang merampingkan pemrograman untuk arsitektur yang sangat terdistribusi.

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Versi awal Java ditahun 1996 sudah merupakan versi rilis sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

- `java.lang`: Peruntukan kelas elemen-elemen dasar
- `java.io`: Peruntukan kelas input dan output, termasuk penggunaan berkas.
- `java.util`: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
- `java.net`: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
- `java.awt`: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
- `java.applet`: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Seperti telah dibahas sebelumnya, banyak jenis komputer dan sistem operasi yang terhubung ke Internet. Untuk program-program untuk secara dinamis didownload

ke semua berbagai jenis platform, beberapa sarana untuk menghasilkan kode dieksekusi portabel diperlukan. Mekanisme yang sama yang membantu menjamin keamanan juga membantu menciptakan portabilitas karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Java memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lain, diantaranya:

- a. Multiplatform
- b. OOP (*Object Oriented Programming*)
- c. Library Class yang lengkap.
- d. Mewarisi Kekayaan C/C++
- e. Pengumpulan Sampah Otomatis

1. Mudah didekompilasi

Dekompilasi adalah proses membalikkan hasil kompilasi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/direverse-engineer.

2. Pengguna memori yang banyak

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal.

2.3.6 Web Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service

menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler. Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya [7].

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

- Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
- Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
- Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

2.3.7 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [8].

JSON terbuat dari dua struktur:

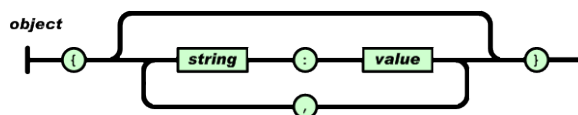
Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

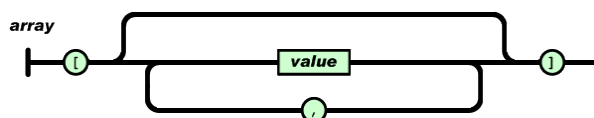
JSON menggunakan bentuk sebagai berikut:

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



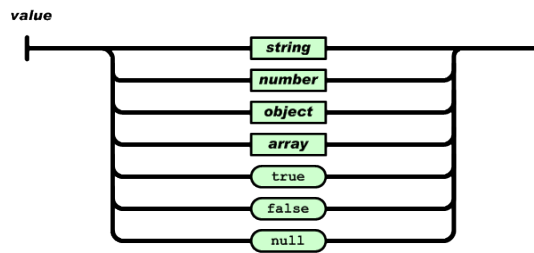
Gambar 2. 4 Object JSON

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



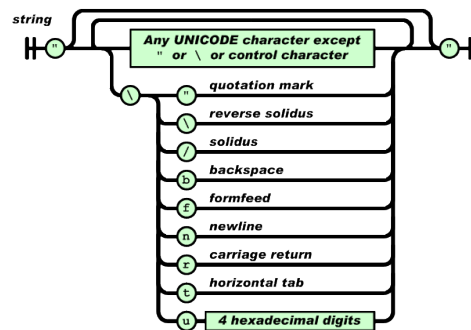
Gambar 2. 5 Array JSON

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau *angka*, atau *true* atau *false* atau *null*, atau sebuah *objek* atau sebuah *larik*. Struktur-struktur tersebut dapat disusun bertingkat.



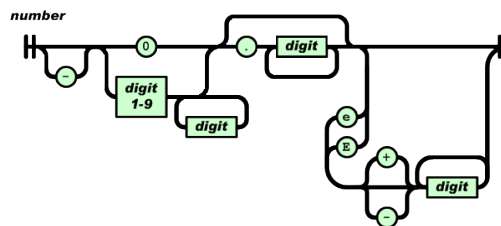
Gambar 2. 6 Value JSON

String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar 2. 7 String JSON

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2. 8 Number JSON

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.

2.3.8 API

API adalah sekumpulan perintah, fungsi, serta protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

API dapat menjelaskan cara sebuah tugas (*task*) tertentu dilakukan. Dalam pemrograman prosedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakan [9].

2.3.9 API Clarifai

Clarifai adalah alat pengenalan gambar dan video yang secara otomatis memberikan tag ke obyek dan kategori mengambil hanya piksel sebagai input, menggunakan brary semantik dan visual untuk kecerdasan buatan. Sistem ini didasarkan pada jaringan saraf, teknik pembelajaran mesin scalable yang dapat menangani skala besar konten visual yang mengalir melalui API. Clarifai juga menggunakan kesamaan emantik dan visual untuk menyeberangi membandingkan gambar yang diunggah dengan gambar lainnya di library mereka untuk menampilkan kesamaan.

1. Face Detection
Menghitung jumlah wajah dalam gambar



FACE DETECTION MODEL

27 faces detected

Gambar 2. 9 Face Detection Model

2. Color Model
Mendeteksi warna apa saja yang ada dalam gambar



COLOR MODEL



Gambar 2. 10 Color Model

3. Apparel Model

Mengidentifikasi item apa yang ada dalam gambar



APPAREL MODEL	
PREDICTED CONCEPT	PROBABILITY
Sunglasses	1.000

Gambar 2. 11 Apparel Model

4. Food Model

Mengidentifikasi bahan apa saja yang ada dalam makanan

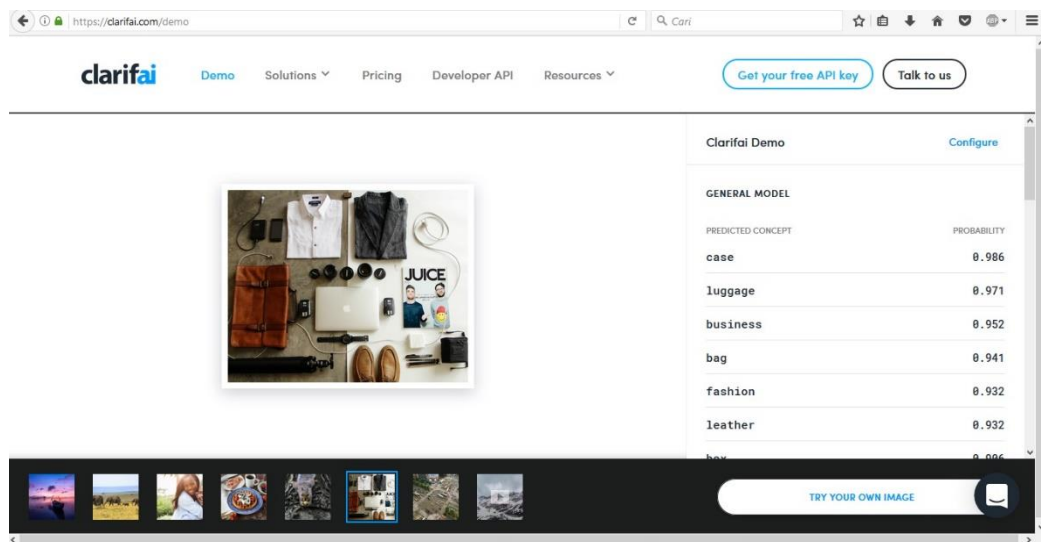


FOOD MODEL	
PREDICTED CONCEPT	PROBABILITY
pizza	1.000
crust	1.000
dough	0.999
pepperoni	0.999
sauce	0.999
mozzarella	0.998
cheese	0.998
tomato	0.995
salami	0.990
pie	0.961
ham	0.879

Gambar 2. 12 Food Model

2.3.10 Cara Kerja Clarifai

Teknologi Clarifai bergantung pada penggunaan Convolutional Neural Networks CNN untuk memproses gambar, dan kemudian menghasilkan daftar tag gambar. CNNs secara de didefinisikan sebagai model pembelajaran mesin hirarkis, yang belajar gambar yang kompleks representasi dari volume data yang besar dijelaskan. Mereka menggunakan beberapa lapisan transformasi dasar yang akhirnya menghasilkan representasi yang sangat canggih []



Gambar 2. 13 Clarifai Demo

Clarifai bekerja melalui analisis gambar untuk menghasilkan daftar deskripsi tag dari gambar yang diberikan. Untuk setiap tag dalam daftar ini, sistem juga menyediakan nilai probabilitas. probabilitas ini merupakan kemungkinan menggambarkan gambar menggunakan tag spesifik. The Clarifai API dapat diakses sebagai layanan web jarak jauh. Skema kerja teknologi Clarifai ditampilkan pada Gambar. 2.15. Dalam skema ini, Clarifai menggunakan gambar ViDRILO sebagai masukan dan menggunakan CNNs untuk menganalisis dan menghasilkan daftar label dan probabilitas.

2.3.11 Object Oriented Analisis Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

2.3.12 Unified Modeling Language

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, et. al. 2004).

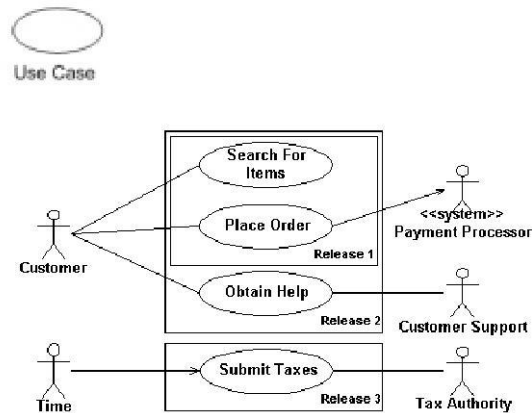
1. Use Case Diagram

Diagram yang menggambarkan actor, use case dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case.

Use Case memiliki dua istilah :

- System use case; interaksi dengan sistem.

- Business use case; interaksi bisnis dengan konsumen atau kejadian nyata



Gambar 2. 14 Use Case Diagram

1) Actor

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu. Bisa merupakan manusia, sistem, atau *device* yang memiliki peranan dalam keberhasilan operasi dari sistem.

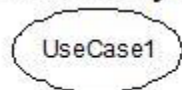


Gambar 2. 15 Actor

2) Use Case

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

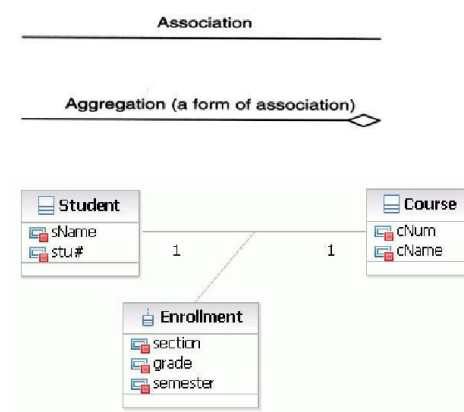
Use-case symbol



Gambar 2. 16 Use Case

3) Association

Mengidentifikasi interaksi antara setiap actor tertentu dengan setiap use case tertentu. Digambarkan sebagai garis antara actor terhadap use case yang bersangkutan. Asosiasi bisa berarah (garis dengan anak panah) jika komunikasi satu arah, namun umumnya terjadi kedua arah (tanpa anak panah) karena selalu diperlukan demikian.



Gambar 2. 17 Association

5) Dependency

Dependensi <<include>>

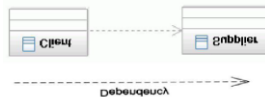
- Mengidentifikasi hubungan antar dua use case dimana yang satu memanggil yang lain.
- Jika pada beberapa use case terdapat bagian yang memiliki aktivitas yang sama maka bagian aktivitas tersebut biasanya dijadikan use case tersendiri dengan relasi dependensi setiap
- use case semula ke use case yang baru ini sehingga memudahkan pemeliharaan.

Digambarkan dengan garis putus-putus bermata panah dengan notasi <<include>> pada garis.

- Arah mata panah sesuai dengan arah pemanggilan

Dependensi <<extend>>

- Jika pemanggilan memerlukan adanya kondisi tertentu maka berlaku dependensi <<extend>>.
- Digambarkan serupa dengan dependensi <<include>> kecuali arah panah berlawanan.
- Note: konsep “extend” ini berbeda dengan “extend” dalam Java.



Gambar 2. 18 Dependency

6) Generalization

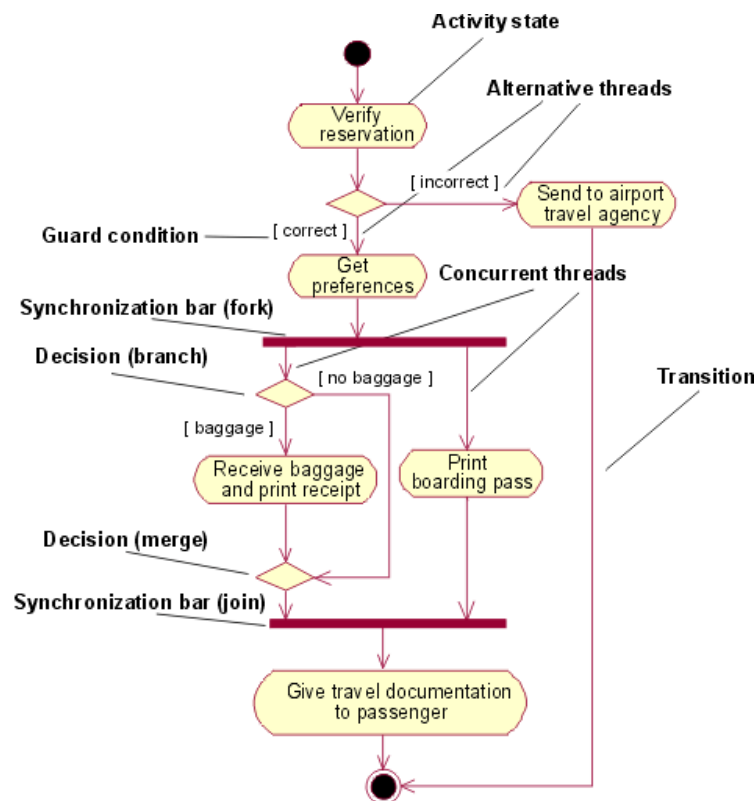
Mendefinisikan relasi antara dua actor atau dua use case yang mana salah satunya meng-inherit dan menambahkan atau override sifat dari yang lainnya. Penggambaran menggunakan garis bermata panah kosong dari yang meng-inherit mengarah ke yang di-inherit.



Gambar 2. 19 Generalization

2. Activity Diagram

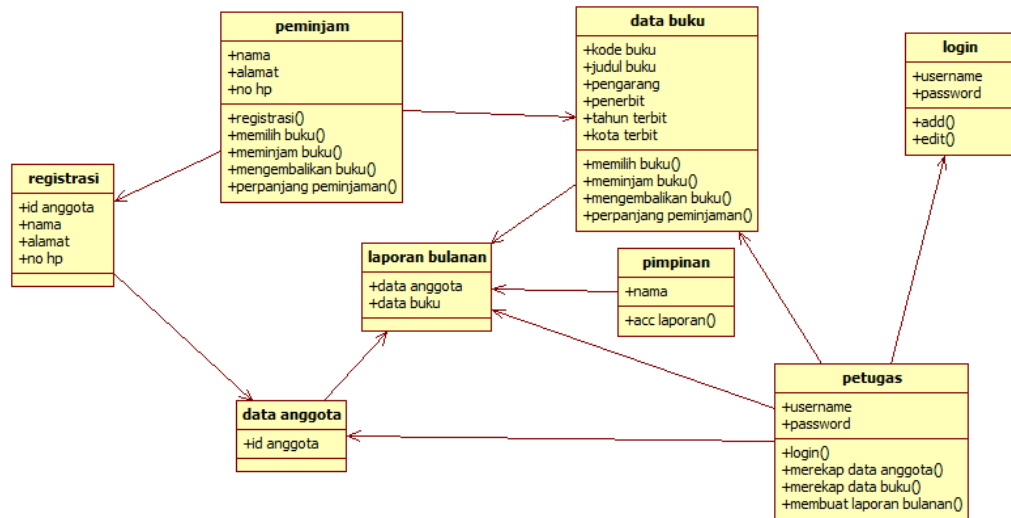
Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dimana biasanya dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. *Activity Diagram* ini sendiri memiliki struktur diagram yang mirip *Flowchart* atau *data flow diagram* pada perancangan terstruktur. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *Use Case Diagram*.



Gambar 2. 20 Activity Diagram

3. Class Diagram

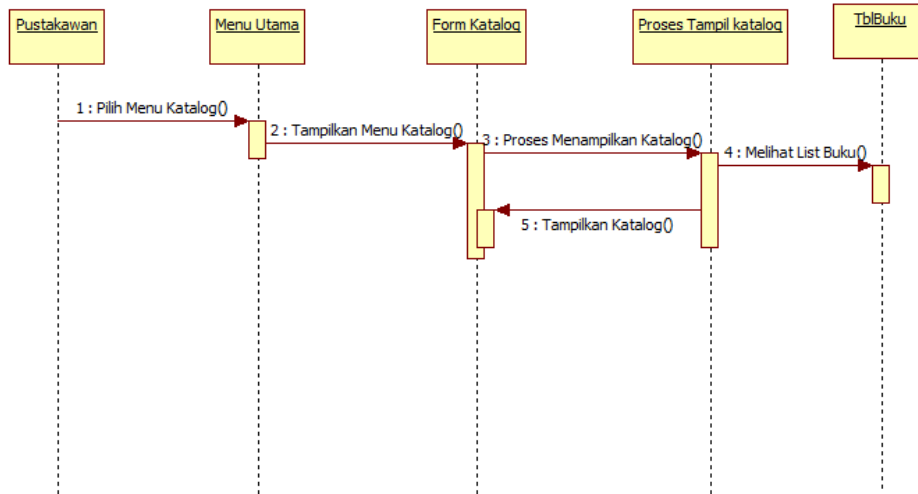
Class diagram merupakan diagram yang selalu ada di pemodelan system berorientasi objek. Class diagram menunjukkan hubungan antar class dalam system yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai satu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi property dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private dan protected.



Gambar 2. 21 Class Diagram

4. Sequence Diagram

Sequence Diagram adalah suatu diagram yang digunakan untuk menggambarkan dan menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case. Adapun urutan proses yang dijelaskan yaitu interaksi yang terjadi antar class, operasi yang terlibat, urutan antar operasi, dan informs yang diperlukan oleh masing-masing operasi. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, message yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan progress *vertical* .



Gambar 2. 22 Sequence Diagram

