

BAB 2

LANDASAN TEORI

2.1 Tentang Robonesia.id

Robonesia.id merupakan perusahaan berkembang dibawah naungan PT. Kolaborasi Kapital Indonesia yang memberikan fasilitas pendidikan robotika berbasis project kepada siswa tingkat sekolah dasar hingga sekolah menengah atas. Robonesia yang beralamat di Jl. Kp. Padi, Dago, Kecamatan Coblong, Kota Bandung, Jawa Barat 40135.

2.2 Sejarah dan Profil Robonesia.id Bandung

Robonesia atau pada awal berdirinya bernama Robonesia.id adalah salah satu perusahaan yang sedang berkembang di Kota Bandung dan bergerak dibidang robotika. Robonesia pertama didirikan pada tahun 2016, pada saat itu masih berlokasi di Komp. Bumi Panyileukan Jl. Indah Raya Bandung. Nama Robonesia sendiri merupakan singkatan dari gabungan kata “ Rumah Robot Indonesia”, singkatan tersebut memiliki filosofi tentang harapan Robonesia dimasa mendatang yaitu untuk menjadi tempat berkumpul dan lahirnya karya – karya produk robotik yang dan bermanfaat bagi kehidupan bangsa Indonesia.

Awal berdirinya Robonesia, dipelopori oleh ide beberapa mahasiswa asal Perguruan Tinggi Swasta Bandung yang melihat bahwa ada peluang usaha yang cukup besar dibidang robotika. Mengingat saat ini tekonologi kemajuannya semakin pesat, dan diperlukan dimana saja. Hal tersebut dapat dibuktikan dengan maraknya *event* atau lomba robot berbasis IT dibeberapa sekolah maupun perguruan tinggi. Dalam menghadapi hal tersebut, diperlukan persiapan sedini mungkin serta harus didukung juga dengan adanya cukup baik dan sarana-prasarana penunjang yang tepat. Salah satunya adalah dengan mempelajari hal – hal tersebut sejak usia dini, mulai dari usia anak – anak hingga remaja melalui kursus atau ekskul robotik. Dalam kegiatan kursus atau ekskul robotik anak – anak akan diperkenalkan dengan mempelajari dasar – dasar komputer, elektronika, dan fisika. Melalui kegiatan tersebut, diharapkan anak – anak atau siswa dapat menerapkan hasil pembelajaran tersebut dikehidupan sehari – hari atau dimasa

mendatang hingga menciptakan suatu project robotik yang bermanfaat dan bernilai jual.

Les Privat Robotik, merupakan kegiatan yang ditawarkan oleh Robonesia pada masa awal pendiriannya. Dengan dibantu oleh sekitar 12 pengajar, dengan total awal siswa sebanyak 50 siswa yang tersebar di beberapa lokasi di kota Bandung dari rumah ke rumah. Beberapa bulan setelah pendiriannya, jumlah siswa pun semakin bertambah dan mulai menjalin kerja dengan beberapa sekolah Swasta di Kota Bandung. Lalu pada tahun 2018, Robonesia.id resmi berubah menjadi Robonesia dan berpindah alamat ke Freenovation Kp. Padi, Dago Pojok Bandung.

Beberapa manfaat dari kegiatan pembelajaran robotik antara lain siswa dapat meningkatkan kreatifitasnya, melatih motorik halus, meningkatkan keterampilan, dan imajinasi hingga mampu menghasilkan suatu *project* yang aplikatif dan berbasis teknologi. Adapun materi – materi yang diajarkan dalam kegiatan pembelajaran robotik antara lain pembuatan *project* DIY, pembelajaran animasi melalui program *scracth*, perancangan bangun Lego (MRT) , serta pembelajaran *coding* dasar dengan aplikasi Arduino.

Selama berdirinya, Robonesia pun aktif dalam beberapa kegiatan lomba robotik baik Nasional maupun Internasional. Beberapa diantaranya pernah menjadi juara umum dalam ajang perlombaan Robot & Expo Competition piala walikota Bandung pada tahun 2017 dan Juara 2 *International Youth Robot Competition* (IYRC) pada tahun 2017 di China.

Diawal tahun 2019, Robonesia sudah memiliki sekitar 1000 siswa yang tersebar di beberapa wilayah selain Kota Bandung seperti Kab. Bandung, Kab. Bandung Barat, Purwakarta, dan Depok. Selain bergerak dibidang pendidikan robotika, Robonesia juga memperluas usahanya dengan penjualan produk robot edukasi. Dalam waktu mendatang Robonesia juga berencana memperluas jangkauan pasarnya hingga keseluruhan Jawa Barat, dengan target pencapaian sebanyak 3000 siswa.

2.2.1 Visi dan Misi CV. Robonesia

1. Visi Perusahaan

Menjadi pelopor pendidikan robotik dan teknologi untuk peradaban Indonesia dimasa depan

2. Misi Perusahaan

- 1) Menyiapkan kurikulum robotika profesional berdasarkan kepentingan masyarakat Indonesia
- 2) Menyelenggarakan pendidikan robotika berbasis real-project sosial kemasyarakatan
- 3) Mengembangkan sarana dan prasarana robotika untuk kepentingan bangsa

2.2.2 Logo dan Arti Logo Robonesia

Berikut pada Gambar 2.1 adalah logo Robonesia.id.



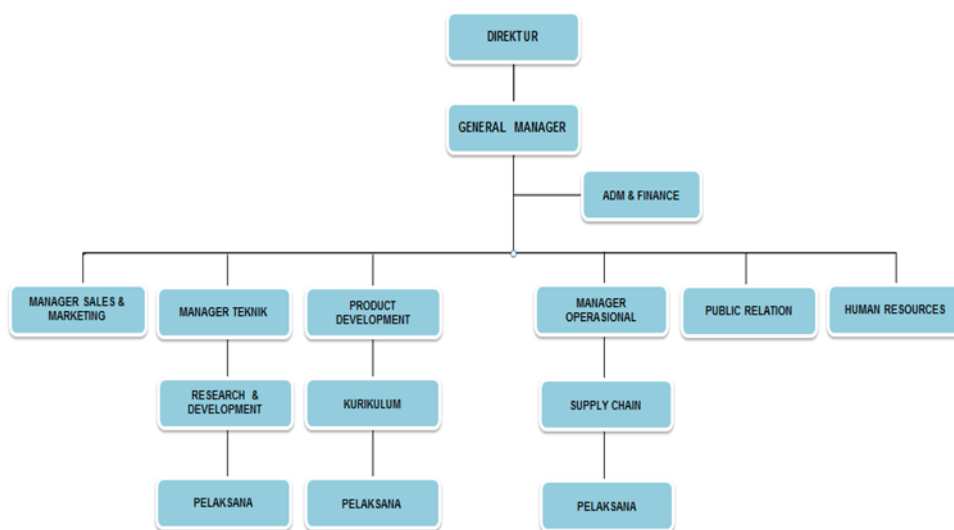
Gambar 2.1 Logo Robonesia

Pada logo Robonesia terdapat dua icon utama yaitu gambar dasar font dan logo utama yang berbentuk kepala robot. Bentuk dasar font yang dipilih yaitu jenis font Nirmala UI yang menyimbolkan arti ketegasan, optimis, dan konsistensi terhadap sesuatu. Pada logo utama berbentuk kepala robot yang merupakan bagian terpenting dari robot dan merupakan simbol yang berkaitan dengan kemajuan teknologi. Dalam hal ini memiliki arti bahwa Robonesia memiliki harapan untuk menjadi salah satu bagian dalam perkembangan teknologi dan pendidikan di Indonesia.

Bagian atas kepala robot diambil menyerupai atap rumah, menyelaraskan dengan nama Robonesia itu sendiri, yaitu rumah robot Indonesia. Bagian bawah kepala robot dibuat melengkung, memiliki filosofi senyuman anak – anak yang menandakan keramahan Robonesia terhadap perkembangan teknologi. Adapun warna yang dipilih adalah warna biru dan kuning. Warna biru yang dipilih merupakan perpaduan dari warna biru langit dan hijau, melambangkan kecerdasan, sifat percaya diri, keharmonisan, adil, empati dan kecerdasan. Sementara warna kuning, bermakna memiliki sifat yang optimis, penuh semangat dalam belajar serta identik dengan psikologis anak – anak yang ceria.

2.2.3 Struktur Organisasi Manajemen CV. Robonesia

Berikut pada Gambar 2.2 adalah Struktur Organisasi Robonesia.id.



Gambar 2.2 Struktur Organisasi Robonesia

2.3 Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data

aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu” [12].

2.4 Monitoring dan Evaluasi

Monev adalah kegiatan monitoring dan evaluasi yang ditujukan pada suatu program yang sedang atau sudah berlangsung. merupakan aktivitas yang dilakukan pimpinan untuk melihat, memantau jalannya organisasi selama kegiatan berlangsung, dan menilai ketercapaian tujuan, melihat factor pendukung dan penghambat pelaksanaan program[13].

Pada pelaksanaannya, monev haruslah dilakukan dengan prinsip-prinsip seperti berikut ini .:

- a. Berorientasi pada tujuan. Monev hendaknya dilaksanakan mengacu pada tujuan yang ingin dicapai. Hasil monev dipergunakan sebagai bahan untuk perbaikan atau peningkatan program pada evaluasi formatif dan membuat justifikasi dan akuntabilitas pada evaluasi sumatif.
- b. Mengacu pada kriteria keberhasilan Monev seharusnya dilaksanakan mengacu pada kriteria keberhasilan program yang telah ditetapkan sebelumnya. Penentuan kriteria keberhasilan dilakukan bersama antara para evaluator, pelaksana program (pimpinan dan staf), lembaga terkait (peserta diklat).
- c. Mengacu pada asas manfaat Monev sudah seharusnya dilaksanakan dengan manfaat yang jelas. Manfaat tersebut adalah berupa saran, masukan atau rekomendasi untuk perbaikan program program yang dimonev atau program sejenis di masa mendatang.
- d. Dilakukan secara obyektif Monev harus dilaksanakan secara objektif. Petugas monev dari pihak eksternal seharusnya bersifat independen, yaitu bebas dari pengaruh pihak pelaksana program. Petugas monev internal

internal harus bertindak objektif, yaitu melaporkan temuannya apa adanya.

2.4.1 Monitoring

Monitoring adalah kegiatan mengumpulkan dan menganalisis informasi secara sistematis pada saat suatu program berjalan, hal ini bertujuan untuk meningkatkan efisiensi dan efektivitas suatu program. Pemantauan didasarkan pada target yang ditetapkan dan kegiatan yang direncanakan selama tahap perencanaan kegiatan. Ini membantu agar pekerjaan tetap berjalan, dan manajemen tahu jika ada yang tidak beres. Jika dilakukan dengan benar, pemantauan adalah alat yang sangat berharga untuk pengelolaan yang baik dan berguna untuk evaluasi[14]. Dalam proses medis, pemantauan sering digunakan untuk mendiagnosis dan mendiagnosis status fisiologis pasien melalui sensor bio-medis dengan menganalisis dan menganalisis informasi fisiologis[15]. Selain dunia medis, dalam prakteknya monitoring sering dilakukan dalam dunia militer menggunakan radar udara sebagai pengawasan dan udara sebagai bentuk monitoring kedaulatan negara[16].

2.5 Pembelajaran

Pembelajaran adalah proses interaksi siswa dengan guru dan sumber belajar dalam suatu lingkungan belajar. Proses pembelajaran merupakan salah satu bentuk implementasi dari kurikulum yang ada pada suatu program studi. Salah satu rangkaian pembelajaran adalah pembelajaran itu sendiri. Sistem pembelajaran terdiri dari input, proses, dan output, kemudian evaluasi evaluasi terdapat tiga yaitu evaluasi input, proses, dan output pembelajaran. Evaluasi masukan atas evaluasi mahasiswa, kelengkapan dan kondisi sarana dan prasarana pembelajaran, materi dan kesiapan dosen, kurikulum dan materi pembelajaran, strategi pembelajaran yang sesuai mata kuliah, dan kondisi lingkungan tempat pembelajaran berlangsung. [14]. Lingkungan belajar diartikan sebagai segala sesuatu yang terjadi dalam institusi pendidikan ini termasuk dalam konteks pendidikan, fisik, sosial, dan psikologis [17].

2.5.1 Projek-projek Pembelajaran Robonesia

Pembelajaran yang ada di ekskul robonesia saat ini ada tiga bentuk. Setiap bentuk memiliki beberapa materi pembelajaranyang berbeda. Berikut ini adalah beberapa bentuk-bentuk pembelajaran yang diajarkan di ekskul robotik robonesia :

1. D.I.Y (*Do It Yourself*)

D.I.Y adalah projek pembelajaran yang aktifitasnya membuat dekorasi atau membangun suatu projek tertentu dengan menggunakan komponen-komponen elektronika. Pada dasarnya projek ini ditujukan untuk aktifitas yang mengarah kepada kegiatan membangun dan merakit dengan menggunakan alat dan bahan yang biasa kita gunakan sehari-hari.

2. Lego

Lego adalah projek pembelajaran yang bertujuan untuk melatih kreatifitas dari segi kognitif (berfikir kreatif), afektif (sikap kreatif) dan psikomotor (keterampilan). Lego berbentuk balok dan berbahan plastik, balok-balok tersebut dapat disusun menjadi model apa saja seperti mobil, kereta api, helikopter, dan robot.

3. Animasi

Animasi adalah projek pembelajaran yang aktifitasnya membuat game dalam bentuk animasi. Game animasi tersebut dibuat dengan menggunakan software atau aplikasi Scratch. Scratch bisa digunakan untuk membuat cerita interaktif, permainan, seni (*art*), dan *simulator*.

2.5.2 Penilaian Hasil Pembelajaran

Penilaian merupakan pengumpulan informasi mengenai perubahan kualitas dan kuantitas di dalam diri peserta didik atau kelompok. Blaustein mengatakan bahwa penilaian (*assessment*) adalah proses mengumpulkan informasi dan membuat keputusan berdasarkan informasi itu. Assesment menurut Stock, dkk adalah kegiatan penilaian yang biasanya dihubungkan dengan kemampuan seseorang, seperti kecerdasannya, keterampilan, kecepatannya, ketepatannya, yang terkait dengan pekerjaan atau tugasnya. Hasil asesment biasanya dinyatakan dalam angka atau huruf[18].

Penilaian pembelajaran proyek robotik terbagi atas 2 yaitu penilaian proyek dan penilaian sikap. Penilaian project terdiri dari nilai pemahaman, kerapian, ketepatan, kreatifitas, presentasi sedangkan Penilaian Sikap terdiri dari nilai kerja sama, tanggung jawab, disiplin, mandiri dan percaya diri. Berikut merupakan contoh penilaian pembelajaran robotik robonesia dapat diamati pada tabel 2.1.

Tabel 2.1 Tabel penilaian pembelajaran.

NIM	Kelas	Nama Siswa	Projek						Sikap					
			Pemahaman	Kerapihan	Ketepatan	Kreatifitas	Presentasi	Jumlah	Keterangan	Kerja Sama	Tanggung Jawab	Disiplin	Mandiri	Percaya Diri
B190357	7	Abdul Malik	2	2	2	2	3	11	B	B	B	B	B	B
B190359	7	Aditya Rasyid	3	3	2	3	3	14	A	A	B	A	A	B
B190360	7	Muhammad Fais	3	2	3	2	2	12	B	B	A	A	A	A
B190363	7	Abdillah Ridho Alriza	2	2	3	3	3	13	B	B	A	B	A	B
B190364	7	Muhammad Arbi Pratama	2	2	3	3	3	13	B	A	A	B	B	B
B190365	7	Muhammad Luqman Pratama	3	2	3	2	3	13	B	B	A	B	A	B
B190366	7	Istvan Azka Fakhruy	2	2	2	2	3	11	B	B	B	B	B	B
B190369	7	Faisal Aqil	3	3	2	2	4	14	A	A	B	A	B	A
B190370	7	Ibrahim Anas	2	1	2	2	2	9	B	B	B	B	B	B

2.6 Rekomendasi

Rekomendasi adalah saran yang mengajurkan, membenarkan, atau memperkuat suatu hal atau pendapat seseorang. Rekomendasi diberikan untuk suatu hal yang di benarkan dan layak di dapatkan oleh seseorang. Misalkan seseorang yang akan membeli sebuah barang pada sebuah toko online, dimana seseorang tersebut dapat melihat barang-barang yang telah dijual oleh toko tersebut dan dibeli oleh orang lain. Seseorang tersebut dapat melihat dan testimoni pada barang yang akan dipilihnya apakah barang yang telah dibeli orang lain sebelumnya memebrikan respon positif atau tidak, dengan begitu seseorang tersebut dapat mempertimbangkan untuk memebli atau tidak barang tersebut.

Dalam penelitian ini, rekomendasi project robotik berdasarkan level kemampuan siswa dalam menyelesaikan tiap projek robotik. Cara untuk mendapatkan rekomendasi dilakukan perbandingan pada hasil penilain pada projek, adapun metode perbandingannya dapat dilihat pada gambar 2.3 :

2.7 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux. Pada awalnya dikembangkan oleh android Inc. Android terdiri dari system operasi,

middleware dan aplikasi. Android adalah sebuah istilah dalam bahasa Inggris yang berartikan robot yang menyerupai manusia. Dalam proses pengembangan sistem operasi Android, dibentuklah organisasi Open Handset Alliance. Google merilis software open source untuk Android, sehingga dapat berkontribusi untuk mengembangkan Android[19].

Berikut beberapa kelebihan Android:

1. Sangat mudah mengoperasikan Smartphone Android.
2. Banyak fitur.
3. Dari segi tampilan, Android tidak kalah bagusnya dengan OS lain.
4. Operating system dibuat open source untuk pengembangan.
5. Aplikasi yang gratis hingga berbayar, dan bisa di-download di Google Play.

2.7.1 Sejarah Android

Perjalanan Android dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan Android.Inc, di California US. Visi Android untuk mewujudkan mobile device yang lebih peka dan mengerti pemilikinya, kemudian menarik raksasa dunia maya Google. Google kemudian mengakuisisi Android pada Agustus 2005. OS Android dibangun berbasis platform Linux yang bersifat open source, senada dengan Linux, Android juga bersifat Open Source. Dengan nama besar Google dan konsep open source pada OS Android, tidak membutuhkan waktu lama bagi android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windows Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa Operating System bagi Smartphone[20].

Hingga awal dan menjelang tahun 2020 terdapat beberapa versi Android yang tersedia untuk digunakan, diantaranya sebagai berikut tabel 2.2:

Tabel 2.2 Tabel Versi Android.

Codename	Version	API level/NDK release
Android10	10	API level 29
Pie	9	API level 28
Oreo	8.1.0	API level 27
Oreo	8.0.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no codename)	1.1	API level 2
(no codename)	1.0	API level 1

2.8 MPAndroidChart Library

MPAndroidChart adalah pustaka yang digunakan untuk membuat grafik di aplikasi android. MPAndroidChart mendukung grafik garis, batang, lingkaran, radar, gelembung, kandil, serta skala, seret, dan animasi. Perpustakaan MPAndroidChart ini juga dilengkapi dengan fitur yang memungkinkan pengguna untuk memperbesar, menggulir, dan membuat grafik untuk fokus pada area yang dipilih. Pengembang bahkan dapat mengubah grafik animasi dan menyesuaikannya dengan keinginan masing-masing pengguna [21].

Library ini dikembangkan oleh Philipp Jahoda dengan lisensi Apache. Contoh aplikasi yang menggunakan MPAndroidChart diantaranya My Expenses, Phone Addiction, Drivvo. Library MPAndroidChart mendukung minimal SDK 8 (Android 2.2–2.2.3 Froyo).

Beberapa pilihan grafik yang bisa anda coba:

- a) Simple Bar Chart
- b) Grouped Bar Chart
- c) Horizontal Bar Chart
- d) Simple Line Chart
- e) Line Chart with Cubic Lines
- f) Grouped Line Chart
- g) Combined Line and Bar Chart
- h) Pie Chart
- i) Scatter Chart
- j) Candlestick Chart
- k) Radar Chart

2.9 Payment Gateway

Payment Gateway adalah mediator antara bank, pedagang, dan pengguna. Dengan teknologi dan basis penggunaanya, pembayaran elektronik adalah praktik yang umum[22]. Pinjaman jangka pendek yang ditawarkan kepada pelanggan yang memiliki gateway pembayaran dikenal sebagai pinjaman gateway pembayaran[23]. Fungsi *Payment gateway* sebenarnya sebagai saluran terenkripsi, yang dengan aman mengirimkan rincian transaksi dari pembeli menggunakan komputer pribadi ke bank untuk mendapatkan persetujuan[24].

2.10 API Midtrans

Veritrans atau Midtrans adalah *payment gateway* yang mendukung *ECommerce (online shop)* di Indonesia untuk dengan mudah menerima pembayaran dari pelanggan.. Dengan Veritrans atau Midtrans , Anda dapat mempermudah proses pembayaran di situs web atau di aplikasi seluler. Veritrans

atau Midtrans merupakan *tools* jenis *API Key* yang dapat digunakan dalam mengintegrasikan pada website *e-commerce* hingga *mobile apps* sehingga website ataupun perangkat *mobile* yang terintegrasi dengan *API Key* Veritrans atau Midtrans dapat melakukan pembayaran secara *online*. *API Key* Veritrans atau Midtrans dapat diunduh secara gratis pada situs resminya veritrans.co.id. atau midtrans.com. Namun, veritrans atau Midtrans juga menyediakan beberapa layanan premium berbayar karena di fitur premium terdapat transaksi online [24].

2.11 Firebase Cloud Messaging

Berikut pada Gambar 2.3 adalah logo Firebase.



Gambar 2.3 Firebase Cloud Messaging

Firebase merupakan teknologi BaaS (*back-end as a service*) berbasis cloud dari Google yang memiliki beberapa layanan, salah satunya adalah layanan fitur notifikasi (*push notification*) yang disebut FCM (*firebase cloud messaging*). Pengembangan FCM berasal dari GCM (*Google cloud messaging*) yang menyediakan layanan sehingga developer dapat mengirim pesan notifikasi ke semua perangkat android yang terhubung dari server. Dalam studi ini FCM digunakan untuk mengirimkan notifikasi kepada pengguna aplikasi berupa pembayaran alert melalui push notifikasi[25]. Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas platform yang memungkinkan pengiriman pesan tanpa biaya [26] dan tidak punya kuota. Perpesanan cloud Firebase digunakan melalui jaringan seluler atau Wi-Fi, dengan biaya hampir nol. Dengan menggunakan JSON untuk menarik (retrieve) data dari firebase cloud messaging. Pengembangan FCM berasal dari GCM (*Google cloud messaging*) yang menyediakan layanan sehingga developer dapat mengirim pesan notifikasi ke semua perangkat android yang terhubung dari server[27].

2.12 Push Notification

Push Notification adalah sebuah layanan yang banyak digunakan untuk keperluan pemberitahuan melalui pesan pendek yang ada di smartphone. Dengan adanya layanan Push Notification tersebut, pengguna dapat terbantu dalam hal yang bersifat pemberitahuan secara singkat. Pada implementasinya *Push Notification* dapat dimanfaatkan dalam berbagai keperluan sehari-hari misalnya untuk monitoring absensi, *update* berita terbaru, dan sebagainya. Kurangnya pengetahuan dalam memberikan informasi secara *real time*, berakibat informasi yang tersampaikan tidak *up-to-date*, sehingga dalam berbagai situasi dan kondisi informasi yang diberikan telah usang. *Push Notification* salah satu layanan yang dapat menjawab masalah tersebut sehingga tidak ada lagi informasi yang terbaru tidak tersampaikan, dengan penggunaan layanan ini setiap terjadi update informasi maka akan langsung terkirim sebagai pesan notification, sehingga informasi yang terbaru tidak akan terlewatkan[28].

2.13 Web Services

Menurut W3C, yang merupakan lembaga internasional yang bergerak dalam membangun standarisasi layanan web, itu adalah perangkat lunak sistem yang dirancang untuk dioperasikan dari mesin ke mesin melalui jaringan. Menurut Chen, dkk, arsitektur layanan web memodelkan interaksi antara tiga peran penyedia layanan, konsumen layanan, dan pendaftar layanan. Penggunaan layanan web juga digunakan oleh Pruter dkk untuk menghubungkan secara dinamis ke perangkat yang dikenal dan tidak dikenal dalam jaringan komputer. Dengan demikian, layanan web memungkinkan sarana operasi antar perangkat lunak menjadi standar pada berbagai platform yang berbeda[29].

Layanan web adalah sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem di jaringan. Layanan web didefinisikan sebagai antarmuka yang menggambarkan sekumpulan operasi yang dapat diakses melalui jaringan, misalnya internet dalam bentuk pesan XML (eXtensible Markup Language) [30].

2.14 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) 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.

JSON terbuat dari dua struktur:

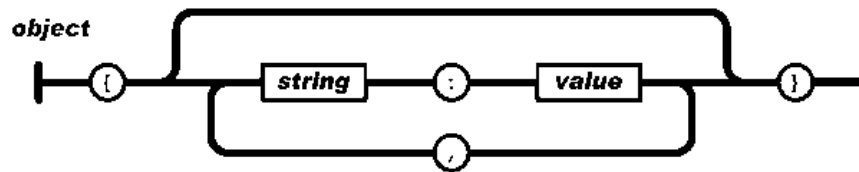
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). 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:

1. Objek

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).

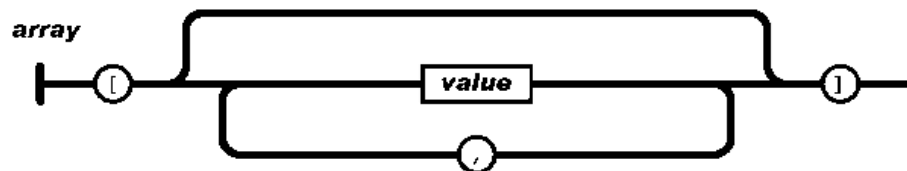


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2.4 JSON Object

2. Larik

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

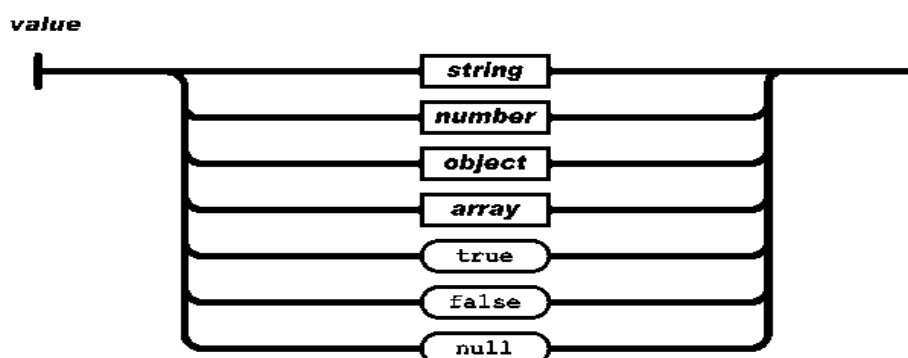


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2.5 JSON Array

3. Nilai

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

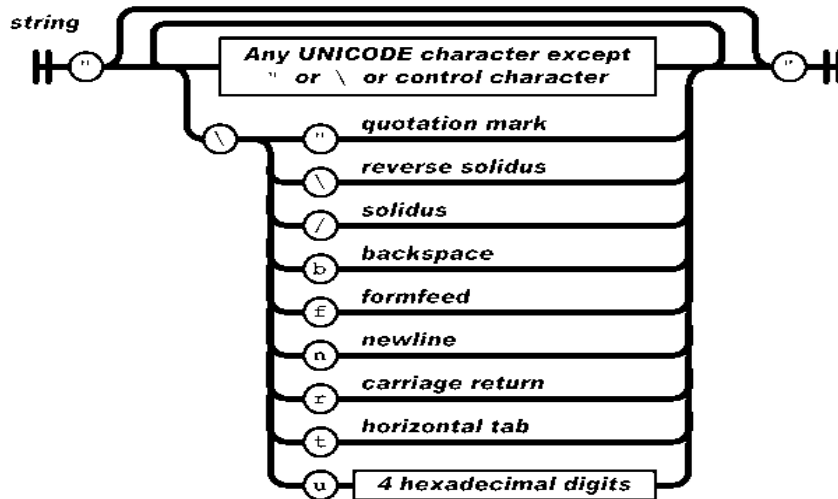


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2.6 Value JSON

4. String

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.

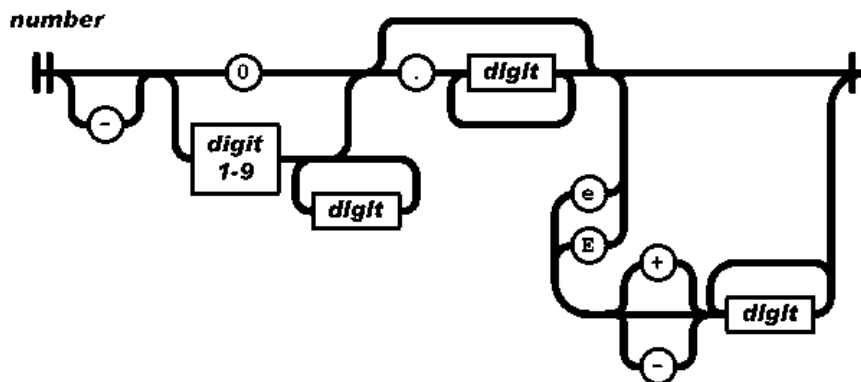


Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2.7 String JSON

5. Angka

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



Sumber Gambar : <http://www.json.org/json-id.html>

Gambar 2.8 JSON Angka

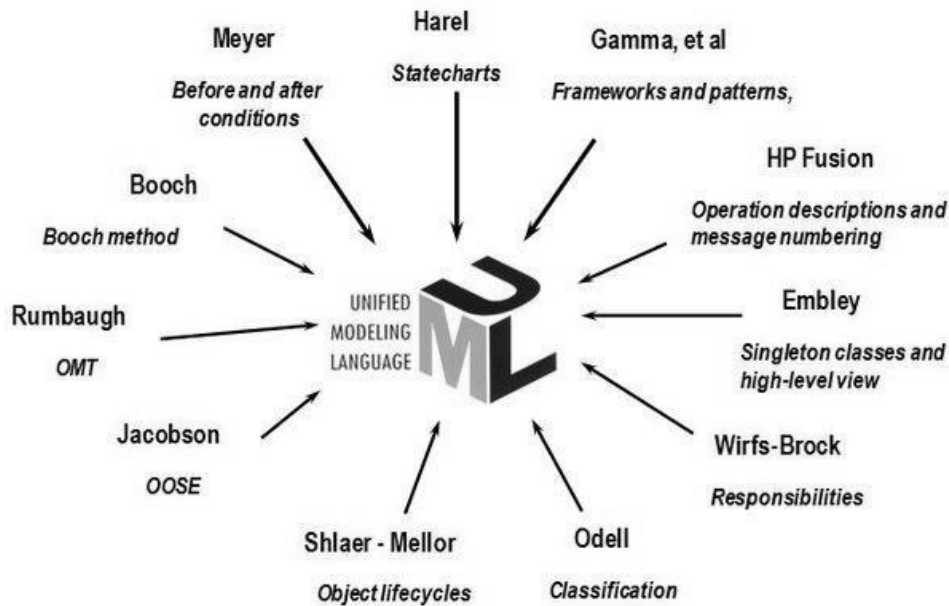
Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh bahasa pemrograman yang bersangkutan.

2.15 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa permodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atau svisi mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Unified Modeling Language (UML) merupakan kesatuan dari Bahasa permodelan yang dikembangkan booch, *Object Modeling Technique* (OMT) dan *Object Orented Software Engineering* (OOSE)[31]. Metode ini menjadikan proses analisis dan design kedalam empat tahapan iterative, yaitu identifikasi kelas-kelas, dan objek-objek, identifikasi semantic dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen, permodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah nalisis, design sistem, design objek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung konsep OO. Metode OOSE dari Jacobson lebih memberi menekankan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian. Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya[28]. Gambar 2.9 berikut adalah unsur-unsur yang membentuk UML.



Sumber Gambar : Pemodelan Visual dengan UML

Gambar 2.9 Unsur-unsur pembentuk UML [31]

Sebagai sebuah notasi grafis yang relative sudah dibakukan (*open standard*) dan dikontrol oleh OMG (*Object Management Group*) mungkin lebih dikenal sebagai badan yang berhasil membakukan CORBA (Common Object Request Broker Architecture), UML menawarkan banyak keistimewaan. UML tidak hanya dominan dalam penotasian di lingkungan OO tetapi juga populer di luar lingkungan OO. Paling tidak ada tiga karakter penting yang melekat di UML yaitu sketsa, cetak biru dan Bahasa pemrograman. Sebagai sebuah sketsa, UML bisa berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dari sistem. Dengan demikian semua anggota tim akan mempunyai gambaran yang sama tentang suatu sistem. UML bisa juga berfungsi sebagai sebuah cetak biru karena sangat lengkap dan detil. Dengan cetak biru ini maka akan bisa diketahui informasi detail tentang coding program (*forward engineering*) atau bahkan membaca program dan menginterpretasikannya kembali kedalam diagram (*reverse engineering*). *Reverse engineering* sangat berguna pada situasi dimana code program yang tidak terdokumentasi asli hilang atau bahkan elum dibuat sama sekali. Sebagai bahasa

pemrograman, UML dapat menterjemahkan diagram yang ada di UML menjadi code program yang siap untuk di jalankan[31].

2.15.1 Diagram UML

UML menyediakan berbagai macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek[31]. Namun, pada penelitian ini hanya menggunakan 4 macam diagram saja untuk memodelkannya. Yaitu:

1. *Use Case Diagram*

Use case diagram adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkahlangkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu[31]. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan Bersama-sama oleh tujuan umum pengguna.

Use case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu system dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah system. Karena system pengembangan tradisional sering cerboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu[31].

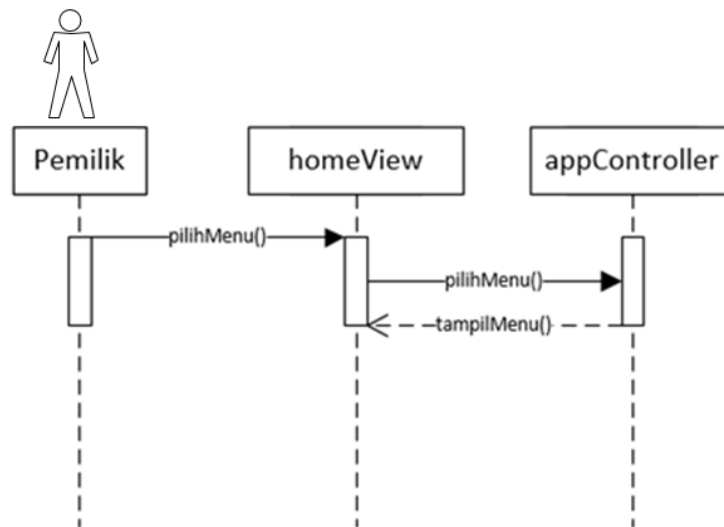
Diagram use case menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan *sistem* atau *sub sistem boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case. Gambar 2.10 mengilustrasikan *actor*, *use case* dan *boundary*.



Gambar 2.10 Use Case Model

2. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakan diantara obyek-obyek ini di dalam use case. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*[31]. Gambar 2.11 mengilustrasikan Sequence diagram.

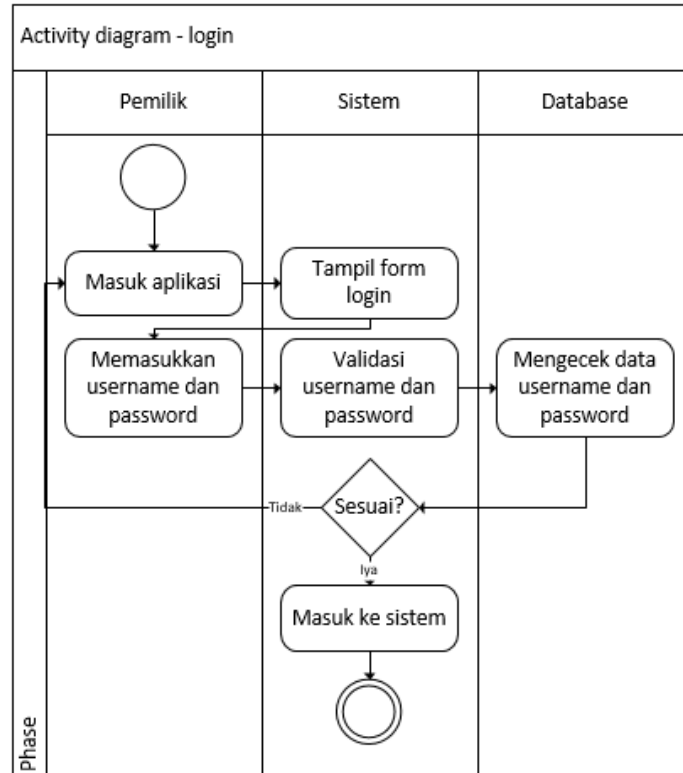


Gambar 2.11 Contoh Sequence Diagram Sederhana

3. Activity Diagram

Activity diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika procedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Tujuan dari *activity diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya. Secara umum *activity diagram* digunakan untuk menggambarkan diagram alir yang terdiri dari banyak aktifitas dalam sistem dengan beberapa fungsi tambahan seperti percabangan, aliran paralel, swim lane, dan sebagainya. Sebelum menggambarkan sebuah *activity diagram*, perlu adanya pemahaman yang jelas tentang elemen yang akan digunakan di *activity diagram*. Elemen utama dalam *activity diagram* adalah aktifitas itu sendiri. Aktifitas adalah fungsi yang dilakukan oleh sistem Setelah aktifitas teridentifikasi, selanjutnya yang perlu diketahui adalah bagaimana semua elemen tersebut berasosiasi dengan constraint dan kondisi. Langa selanjutnya perlu penjabaran tata letak dari keseluruhan aliran agar bisa

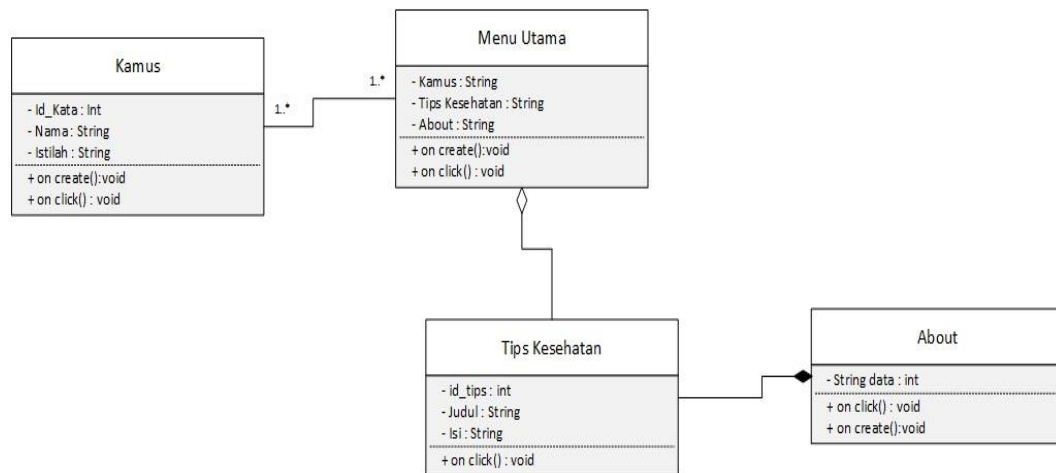
ditransofrmasikan ke activity diagram[31]. Gambar 2.12 menunjukkan contoh *activity diagram* sederhana.



Gambar 2.12 Contoh Activity Diagram Sederhana

4. Class Diagram

Class diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga membangun kode eksekusi dari aplikasi perangkat lunak. *Class diagram* menggambarkan *atribut*, *operation* dan juga *constraint* yang terjadi pada sistem. *Class diagram* banyak digunakan dalam pemodelan sistem OO karena mereka adalah satu-satunya diagram UML, yang dapat dipetakan langsung dengan bahasa berorientasi objek. *Class diagram* menunjukkan koleksi Class, antarmuka, asosiasi, kolaborasi, dan *constraint*. Dikenal juga sebagai diagram structural[31]. Gambar 2.13 menunjukkan contoh *class diagram*.



Gambar 2.13 Contoh Class Diagram.

Class diagram mempunyai 3 relasi dalam penggunaannya, yaitu :

a. *Assosiation*

Assosiation adalah sebuah hubungan yang menunjukkan adanya interaksi antar *class*. Hubungan ini dapat ditunjukkan dengan garis dengan mata panah terbuka di ujungnya yang mengindikasikan adanya aliran pesan dalam satu arah.

b. *Generalization*

Generalization adalah sebuah hubungan antar *class* yang bersifat dari khusus ke umum

c. *Constraint*

Constraint adalah sebuah hubungan yang digunakan dalam sistem untuk memberi batasan pada sistem sehingga didapat aspek yang tidak fungsional.

2.16 Metode Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean[31].

2.16.1 Black Box Testing

Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian *black box* atau bisa disebut pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan perangkat lunak, serta dilihat apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah. *Black box* testing juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program. *Equivalence Partitioning* merupakan metode *black box* testing yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *equivalence partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean[32].

Pengujian *black box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi – fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.

Inisialisasi dan kesalahan terminasi[32].