

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Yayasan Bina Insan Kamil

Profil Yayasan Bina Insan Kamil akan menjelaskan tentang sejarah, logo, visi dan misi yayasan dan struktur organisasi Yayasan Bina Insan Kamil.

2.1.1 Sejarah Pendirian

Yayasan Bina Insan Kamil yang terletak di wilayah Provinsi Banten merupakan yayasan Islam yang dibentuk dan dinotariskan pada tanggal 12 Juni 2009, mempunyai visi dan misi serta bertujuan menyediakan pusat / kawasan pendidikan yang Islami dengan konsep Pendidikan terpadu. Berdirinya YPI dilatar belakangi kepedulian para pendiri tentang konsep pendidikan yang berkualitas, yaitu mempersiapkan generasi penerus dengan dibekali oleh pendidikan yang sesuai dengan karakter ke-islaman, khususnya di wilayah Tangerang dan sekitarnya; di satu sisi kehidupan mereka sudah terpengaruh dengan budaya konsumtif namun di sisi lain perhatian mereka terhadap pendidikan masih minim khususnya pendidikan Islam.

2.1.2 Logo

Berikut ini adalah logo Yayasan Bina Insan Kamil:



Gambar 2.1 Logo Yayasan Bina Insan Kamil

2.1.3 Visi dan Misi Yayasan Bina Insan Kamil

Visi misi adalah suatu konsep perencanaan yang disertai dengan tindakan sesuai dengan apa yang direncanakan untuk mencapai suatu tujuan.

Visi dari Yayasan Bina Insan Kamil adalah:

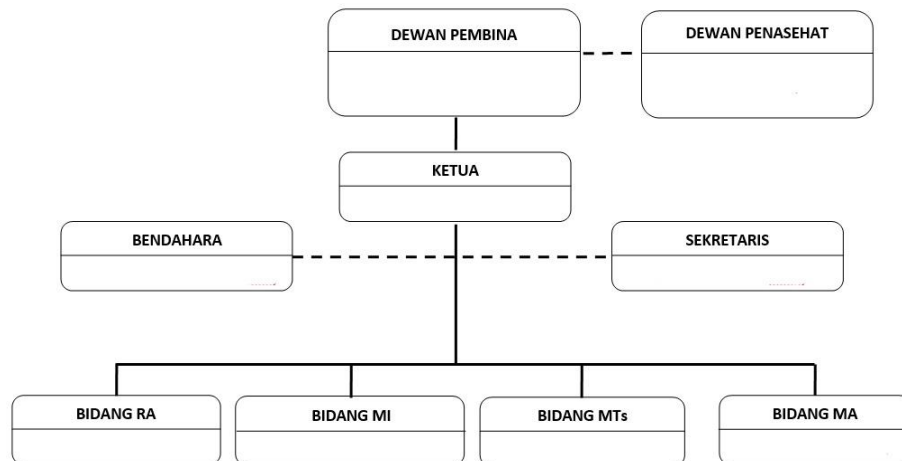
“Bernuansa Islami, unggul dalam prestasi, santun dalam bersikap, diminati masyarakat dan meraih kemuliaan hidup untuk kebahagiaan masa depan.”

Sedangkan Misi dari Yayasan Bina Insan Kamil adalah:

1. Melakukan usaha maksimal dalam pendidikan ahklak dan ilmu berdasar alquran dan hadits.
2. Mewujudkan tercapainya peningkatan mutu pendidikan.
3. Mengembangkan diri sejalan dengan perkembangan Iptek dan Kebudayaan.
4. Meningkatkan pelayanan pendidikan bagi masyarakat.

2.1.4 Struktur Organisasi dan Deskripsi Jabatan

Struktur Organisasi merupakan suatu susunan antara tiap bagian di suatu instansi. Berikut adalah struktur Organisasi yang ada di Yayasan Bina Insan Kamil :



Gambar 2.2 Struktur Organisasi Yayasan Bina Insan Kamil

Adapun tugas pokok dari masing-masing jabatan sebagai berikut :

1. Pembina, mempunyai tugas dan wewenang :
 - a. Memberikan nasehat, arahan dan pertimbangan kepada Pengurus dan Pelaksana, diminta maupun tidak diminta.
 - b. Mengawasi dan mengevaluasi kinerja Bidang-bidang.
2. Ketua, mempunyai tugas dan wewenang :
 - a. Meminta pertanggung jawaban kepada Pengurus.
 - b. Memberi penjelasan kepada masyarakat.
 - c. Mengangkat dan memberhentikan anggota Pengurus, Kepala Madrasah, Staf, Guru, dan Karyawan.
 - d. Membuat Rancangan Anggaran Penerimaan dan Belanja Yayasan (RAPBY).
 - e. Mengawasi dan memeriksa keuangan Yayasan.
3. Sekretaris, mempunyai tugas dan wewenang :
 - a. Mengagendakan dan mengarsip surat keluar masuk.
 - b. Menyusun dan mengagendakan bersama-sama Ketua, mengkoordinasikan dan menertibkan administrasi Yayasan secara umum.
 - c. Mensosialisasikan program Yayasan kepada masyarakat.
 - d. Mengakomodir aspirasi masyarakat dan menyampaikan kepada Pengurus Yayasan.
4. Bendahara, mempunyai tugas dan wewenang :
 - a. Menerima, membukukan dan mengamankan keuangan Yayasan.
 - b. Menyediakan keuangan berdasarkan kebutuhan.
 - c. Mendistribusikan keuangan berdasarkan anggaran.
 - d. Menyampaikan laporan berkala pada musyawarah Pengurus.
 - e. Mengelola dan mengembangkan keuangan Yayasan.
 - f. Mengeluarkan bisyarah Pengurus, Kepala, Staf, Guru dan Karyawan.
 - g. Mengeluarkan uang Yayasan harus ada rekomendasi Ketua dan Sekretaris Yayasan.
 - h. Bersama Kepala Madrasah menyusun RAPBM (unit).

- i. Mengawasi dan mengevaluasi pelaksanaan Anggaran Penerimaan dan Belanja Madrasah (APBM).
 - j. Merencanakan, mengatur dan menertibkan keuangan Unit.
 - k. Melaporkan seluruh program dan hasil kerjanya kepada Koordinator.
 - l. Mengontrol setoran uang dari unit ke Yayasan.
5. Koordinator Bidang Pendidikan, mempunyai tugas dan wewenang :
- a. Melaksanakan program Yayasan dalam bidang kependidikan formal dan non formal
 - b. Mengetahui Rancangan Anggaran Penerimaan dan Belanja Madrasah (RAPBM).
 - c. Melaporkan seluruh kegiatan Madrasah kepada Yayasan.
 - d. Bersama-sama Kepala Madrasah meningkatkan, kualitas dan kuantitas pendidikan dan pengajaran.
 - e. Bersama-sama Kepala Madrasah menyusun kurikulum dan kalender pendidikan.
 - f. Mengawasi dan mengevaluasi pelaksanaan kurikulum pada masing-masing unit.
 - g. Melaporkan seluruh program dan hasil kerjanya kepada Koordinator.
 - h. Berhak mengadakan rapat jika dianggap perlu.

2.2 Landasan Teori

Landasan teori berfungsi sebagai penuntun arah agar fokus penelitian dengan fakta lapangan. Landasan teori juga memberikan suatu gambaran teori-teori yang sesuai dengan penelitian.

2.2.1 Aplikasi

Aplikasi adalah Program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik

pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

Pengertian Aplikasi Secara Umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi pengguna [3].

2.2.2 Teknologi Mobile

Merupakan teknologi komunikasi yang bergerak, dengan kata lain teknologi yang telah tidak memakai penghubung kabel (nirkabel) dalam pemakaian teknologinya. Teknologi komunikasi *mobile* bisa disebut jaringan tanpa kabel yang pemakaiannya memakai penghubung sinyal. Teknologi komunikasi *mobile* adalah perkembangan dari teknologi komunikasi nirkabel yang semakin pesat dalam perkembangannya [4].

2.2.3 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [4].

2.2.4 iOS

iOS adalah kepanjangan dari iPhone OS, yang merupakan sistem operasi perangkat gadget/*smartphone* yang dikembangkan dan didistribusikan oleh Apple Inc. Sistem operasi ini pertama kali diresmikan pada tanggal 29 Juni 2007, yang ketika itu muncullah produk ponsel bernama iPhone. Tidak hanya mendukung perangkat iPhone,

sistem operasi iOS juga berjalan di iPad, iPod Touch, Apple TV dan perangkat lainnya yang dikembangkan secara aktif hingga saat ini [5].

2.2.5 Web Service

Web service merupakan kumpulan suatu layanan berbasis web dengan menggunakan jaringan protokol HTTP, layanan tersebut dapat diakses dan dimanfaatkan oleh pengguna dengan bahasa pemrograman, arsitektur dan sistem operasi yang berbeda (*interoperability*). *Web service* harus bersifat *stateless*, sehingga dapat dibaca dan di akses multi-platform. Arsitektur pada *web service* dapat dibangun menggunakan *Simple Object Access Protocol* (SOAP) atau *Representational State Transfer* (REST), sementara *service* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML [6].

2.2.6 Restful

Representational State Transfer yang disingkat REST yang merupakan gaya arsitektur untuk penerapan *web service* dalam menerapkan konsep perpindahan antar state. REST populer karena kesederhanaannya, sebagai lawan menciptakan standar baru, kerangka kerja dan teknologi.

Keuntungan REST: interaksi berbasis REST menggunakan *Hypertext Transfer Protocol* (HTTP) internet yang merupakan hal yang sudah umum digunakan. Contoh dari pengaturan ini adalah interaksi berbasis REST semua berkomunikasi status mereka menggunakan kode status HTTP standar. Jadi, 404 berarti sumber daya yang diminta tidak ditemukan; kode 401 berarti permintaan tidak diotorisasi; kode 200 berarti semuanya OK; dan 500 berarti ada kesalahan aplikasi yang tidak dapat dipulihkan pada server.

Demikian pula, detail seperti enkripsi dan integritas *transport* data diselesaikan bukan dengan menambahkan kerangka kerja atau teknologi baru, tetapi dengan mengandalkan enkripsi *Secure Sockets Layer* (SSL) dan *Transport*

Layer Security (TLS) yang terkenal. Jadi, seluruh arsitektur REST dibangun di atas konsep yang sudah dikenal sebagian besar pengembang.

Namun, manfaat REST menggunakan konstruk HTTP juga menciptakan batasan. Banyak keterbatasan HTTP juga berubah menjadi kekurangan gaya arsitektur REST. Misalnya, HTTP tidak menyimpan informasi berbasis negara di antara siklus permintaan-respons, yang berarti aplikasi berbasis REST harus tanpa kewarganegaraan dan tugas manajemen negara apa pun harus dilakukan oleh klien [6].

2.2.7 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 [7]. JSON terbuat dari dua struktur:

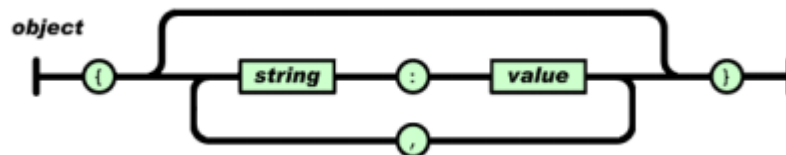
1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array.
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan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

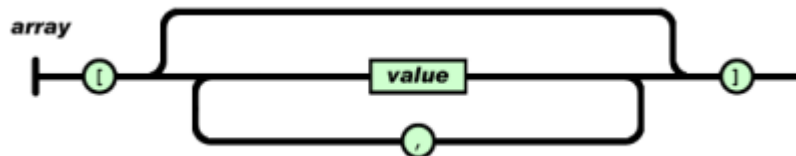
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



Gambar 2.3 Objek JSON

2. Larik

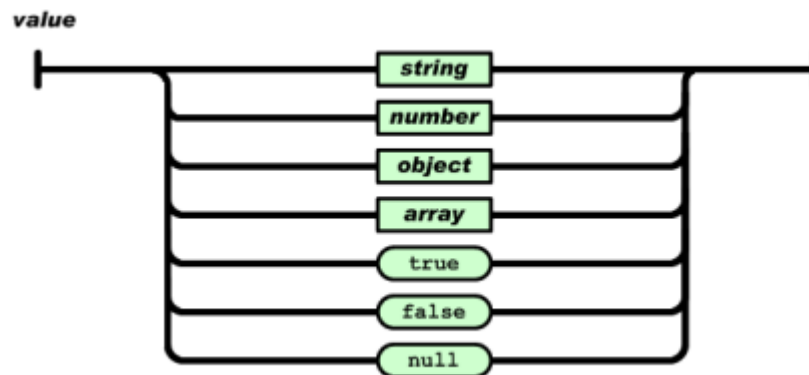
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar 2.4 Larik JSON

3. Nilai

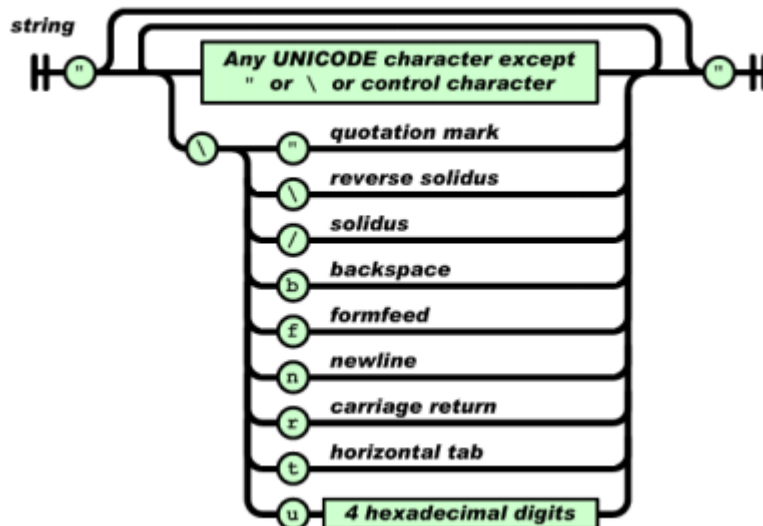
Nilai (value) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2.5 Nilai JSON

4. String

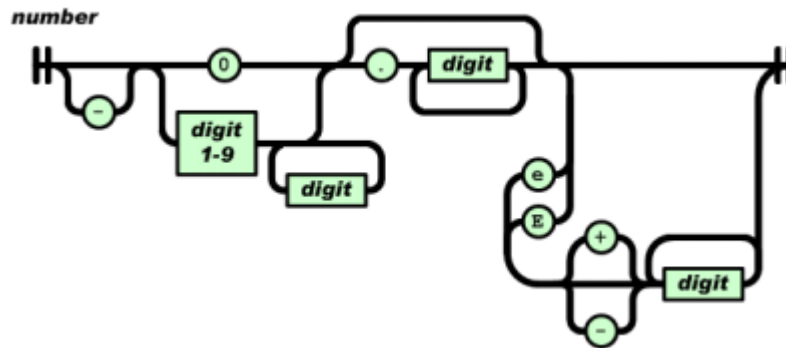
String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar 2.6 String JSON

5. Angka

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



Gambar 2.7 Angka JSON

2.2.8 Node.js

Node.js pertama kali diciptakan dan diperkenalkan untuk pengguna pada sistem Linux pada tahun 2009. Node.js dikembangkan oleh Ryan Dahl dan disponsori oleh Joyent, perusahaan tempat ia bekerja. Node.js adalah salah satu teknologi baru dalam JavaScript. Node.js adalah platform yang dibangun di atas runtime JavaScript Chrome yang bertujuan untuk memudahkan dalam membangun aplikasi jaringan yang cepat dan *scalable*. Node.js menggunakan model I/O *nonblocking* yang membuatnya ringan dan efisien, sempurna untuk aplikasi *real-time* intensif data yang berjalan di perangkat terdistribusi. Survei popularitas Node.js yang dilakukan oleh situs web resmi menunjukkan bahwa unduhan rata-rata lebih dari 35.000 sejak versi 0.10 dirilis pada Maret 2013 [8].

2.2.9 MongoDB

MongoDB merupakan basis data yang tidak relasional. MongoDB ini bersifat open source yang memiliki High Performance. MongoDB adalah sebuah database yang menggunakan konsep manajemen database berorientasi dokumen (document-oriented)

yang dibuat menggunakan pemrograman C++. Orientasi dokumen ini adalah sebuah program komputer yang dirancang untuk menyimpan, mengambil, dan mengelola data yang berorientasi dokumen. MongoDB sudah dikembangkan oleh Iogen sejak Oktober 2007, namun baru dipublikasikan mulai Februari 2009 [9].

2.2.10 Expo

Expo adalah kerangka kerja dan platform untuk seluruh aplikasi yang dibangun menggunakan *library* React . Expo menyediakan seperangkat alat dan layanan yang dibangun di sekitar platform React Native dan platform *native* yang membantu dalam mengembangkan, membangun, menyebarkan dengan cepat di iOS, Android, dan aplikasi web dari basis kode JavaScript / TypeScript yang sama [10].

2.2.11 ReactJS

ReactJS adalah library JavaScript yang digunakan untuk mengembangkan komponen antarmuka pengguna yang dapat digunakan kembali. Berdasarkan dokumentasi resmi React, ReactJS adalah library untuk membangun antarmuka pengguna yang modular. React pada dasarnya memungkinkan pengembangan aplikasi berbasis web yang besar dan kompleks yang dapat mengubah datanya tanpa me-refresh halaman. Cara ini digunakan sebagai View (V) dalam *Model-View-Controller* (MVC). React mengabstrakkan *Document Object Model* (DOM), sehingga menawarkan aplikasi yang sederhana, berkinerja memberikan pengalaman pengembangan yang mudah [11].

2.2.12 React Native

React Native adalah kerangka kerja *native scripting* yang digunakan untuk membuat aplikasi *mobile* lintas platform yang pertama kali diperkenalkan dalam konferensi React.js 2015. Pada awal 2015, React Native hanya mendukung pengembangan aplikasi iOS, namun kerangka kerjanya diperluas untuk menyertakan dukungan Android pada bulan September 2015. React Native menjanjikan pengembangan lintas platform dalam arti bahwa sebagian besar kode aplikasi dapat

dibagikan di antara berbagai platform meskipun diperlukan beberapa kode spesifik platform yang harus ditambahkan ketika menggunakan fitur tertentu. Selain itu, ini adalah kerangka kerja open source yang memungkinkan komunitas untuk berkontribusi pada pengembangannya [12].

2.2.13 JSON Web Token

JWT merupakan sebuah *token* berbentuk *string* yang terdiri dari tiga bagian yaitu: *header*, *payload* dan *signature* yang digunakan untuk proses *autentikasi* dan pertukaran informasi. *Token* terdiri dari dua jenis: yang pertama termasuk *token* pembawa dan *token* pemegang kunci. Sedangkan berdasarkan tujuan terdapat dua skema : *token* identitas dan *token* akses. Cara kerja JWT hampir sama seperti *password*, ketika pengguna dapat berhasil *login* maka server tersebut akan memberikan *token* yang disimpan di *local storage* atau *cookies* browser. Token digunakan untuk mengakses halaman tertentu, pengguna akan mengirim balik *token* tersebut sebagai bukti bahwa pengguna sudah berhasil login. Berikut adalah karakteristik dari JSON Web Token (JWT):

1. Header Header biasanya terdiri dari dua bagian: jenis *token*, JWT, dan algoritma hashing yang digunakan, seperti HMAC SHA256 atau RSA.
2. Payload Bagian kedua dari *token* adalah *payload*, yang berisi klaim. Klaim adalah pernyataan tentang suatu entitas (biasanya, pengguna) dan metadata tambahan. Ada tiga jenis klaim: reserved, public, dan private.
3. Signature untuk membuat bagian tanda tangan kita harus mengambil header yang dikodekan, *payload* yang dikodekan, secret, algoritma yang ditentukan dalam header, dan menandatangani itu.
4. Signature digunakan untuk memverifikasi pengirim bahwa ia adalah pengirim yang benar dan untuk memastikan bahwa pesan yang dikirim tidak diubah.

Mekanisme autentikasi dilakukan dengan memberikan data kredensial *client* kepada server untuk diidentifikasi kemudian apabila berhasil melakukan autentikasi

maka diberikan sebuah *token* yang digunakan untuk melakukan *request* terhadap resource yang tersedia pada sisi server [13].

2.2.14 SMS

Short Message Service (SMS) adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telepon selular, dalam hal ini perangkat nirkabel yang digunakan adalah telepon selular. Salah satu kelebihan dari SMS adalah biaya yang murah.

Selain itu SMS merupakan metode store dan forward sehingga keuntungan yang didapat adalah pada saat telepon selular penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar service area, penerima tetap dapat menerima SMS-nya apabila telepon selular tersebut sudah aktif kembali. SMS menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju media-media wireless dengan menggunakan sebuah Short Messaging Service Center (SMSC), yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan-pesan singkat. Jaringan wireless menyediakan mekanisme untuk menemukan station yang dituju dan mengirimkan pesan singkat antara SMSC dengan wireless station. SMS mendukung banyak mekanisme input sehingga memungkinkan adanya interkoneksi dengan berbagai sumber dan tujuan pengiriman pesan yang berbeda [14].

2.2.15 SMS Gateway

SMS *Gateway* adalah sebuah perangkat lunak pada computer yang menggabungkan teknologi komputer dan teknologi perangkat *mobile* (handphone, PDA, dll.) kemudian diintegrasikan dengan sebuah database yang mempunyai tujuan tertentu seperti SMS nilai pada sebuah sekolah atau universitas, SMS untuk layanan CRM (Customer Relationship Management) pada sebuah rumah makan, café, lembaga pendidikan, dll [15].

2.2.16 Midtrans

Midtrans berdiri pada tahun 2012 dengan menggunakan nama Veritrans yang bertujuan untuk memfasilitasi bisnis online di Indonesia dengan sistem pembayaran yang terpercaya dan mudah untuk digunakan. Pada masa awal berdiri, ruang gerak eCommerce sangat terbatas pada saat itu dikarenakan banyak komponen dasar yang tidak tersedia, sehingga Midtrans berusaha menciptakan solusi yang memungkinkan para pelaku industri agar lebih mudah beroperasi.

Selama beroperasi dalam kurun waktu tiga tahun, Midtrans telah bekerjasama dengan berbagai Bank terkemuka di Indonesia seperti BCA, BNI, Bank Mandiri, CIMB, dan para pemuka di industri teknologi digital. Midtrans kini telah menyediakan 16 metode pembayaran hanya dalam satu langkah integrasi yang mudah untuk berbagai jenis bisnis online agar dapat melakukan pembayaran dengan mudah dan lancar [16].

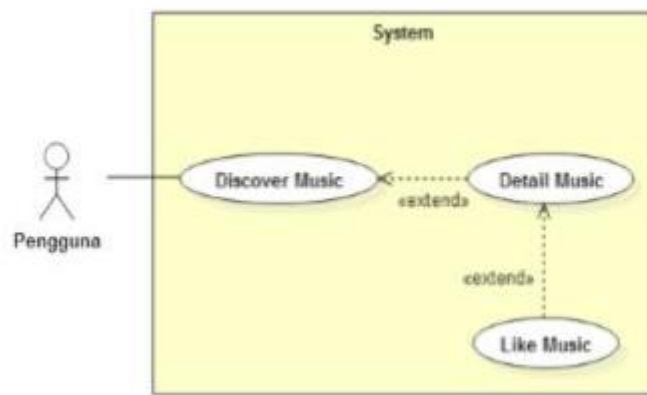
2.2.17 Unified Modeling Language

Unified Modelling Language (UML) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menggunakan class dan operation object dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek [17].

2.2.18 Use Case Diagram

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. Sebuah use case dapat meng-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 meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang serupa.

Sebuah use case juga dapat meng-extend usecase lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain. Dasar menentukan sebuah use case adalah use case merupakan sesuatu yang menyediakan beberapa hasil terukur kepada pengguna atau sistem eksternal. Use case harus memiliki sangat jelas kriteria lulus / gagal. Pengembang, tester, penulis teknis, dan pengguna harus secara eksplisit tahu apakah sistem memenuhi kasus penggunaan atau tidak. Setiap bagian dari use case yang memenuhi tes sederhana ini mungkin menjadi kandidat yang baik untuk use case [18].

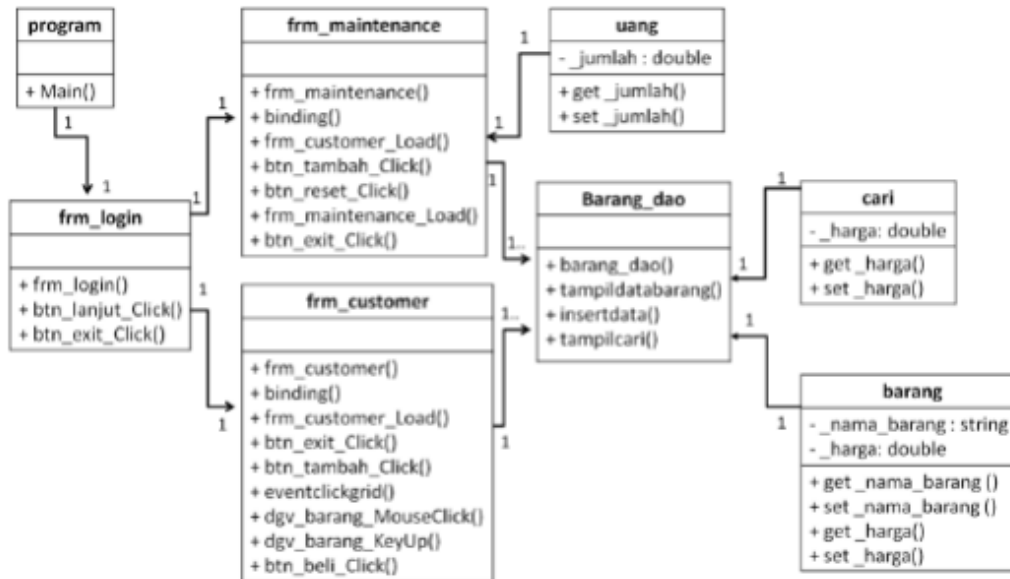


Gambar 2.8 Contoh Use Case Diagram

2.2.19 Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu 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

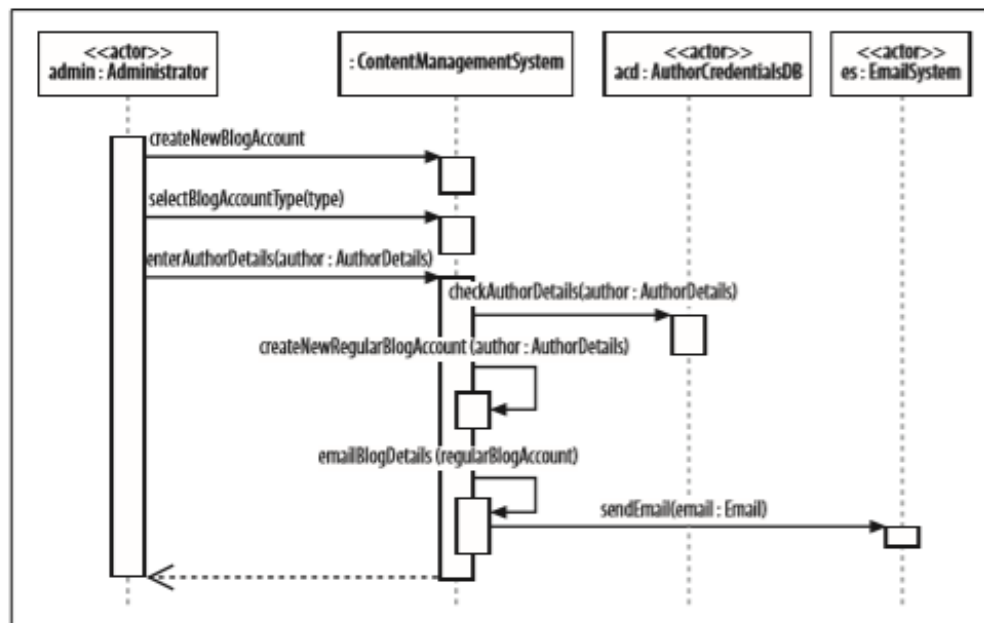
dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas.



Gambar 2.9 Contoh Class Diagram

2.2.20 Sequence Diagram

Sequence diagram adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. Sequence diagram menangkap objek dan class yang terlibat dalam scenario dan urutan pesan yang ditukar antara objek diperlukan untuk 22 melaksanakan fungsionalitas skenario. Sequence diagram berasosiasi dengan use case selama proses pengembangan. Dalam Unified Model Language (UML), objek dalam sequence diagram di dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara : (nama objek), (nama objek dan class) atau (hanya nama class (anonymous object)).



Gambar 2.10 Contoh Sequence Diagram

2.2.21 Object Oriented Analysis and 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.