BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang akan dibahas sebagai pemahaman sistem yang akan dibangun

2.1.1 Tempat Penelitian

PI (Pick up Indonesia) merupakan sebuah club mobil yang bertujuan sebagai wadah untuk pengguna kendaraan Pick up untuk menyalurkan hobi kepada pemilik kendaraan serta untuk menjalin kekeluargaan antar pengguna kendaran.

2.1.1.1 Sejarah PI

Didirikan oleh Sutris Yanto selaku ketua, PI berdiri pada tahun 2017 yang bertempat di Jalan Raya Kedu KM 3 (Depan POM Maron, Candimulyo. Kota Temanggung, Jawa Tengah. Bermodalkan kendaraan beliau mengajak teman temannya untuk membuat suatu club yang bertujuan menjadi wadah bagi pengguna kendaran Pick up.

PI adalah club mobil pick up yang mengedepankan kekeluargaan dan tolong menolong kepada pengguna kendaraan dimanapun berada. Anggota-anggota PI yang ada hampir di setiap kota yang ada diindonesia. Mengetahui dengan pentingnya teknologi yang menyediakan fitur-fitur berbasis IT dengan menggunakan media sosial facebook untuk berkomikasi kesemua anggota, bentuk pendaftaran yang dapat dilakukan tanpa harus bertatap muka atau hadir secara fisik atau tidak langsung.

Saat ini, PI merupakan club yang telah legal dan resmi terdaftar di ikatan motor indonesia (IMI) semenjak tahun 2017 yang bertujuan mengikuti tata cara membuat club yang baik, dengan terdaftarnya ke Ikatan Motor Indonesia anggota dituntut mengikut aturan yang berlaku.

2.1.1.2 Logo Komunitas

Gambar 2. 1 Logo MCCI

Spesifikasi dan Makna Logo:

1. Kotak : Wadah
2. P : Pick up
3. I : Indonesia
4. Merah : berani dan kejujuran
5. Putih : berbuat baik dalam bersaudara

2.1.1.3 Visi

Visi merupakan merupakan pernyataan kondisi masa depan yang ingin diraih oleh organisasi, serta menunjukkan bayangan, keinginan, atau cita-cita yang ingin dicapai organisasi di masa depan. PI memiliki visi sebagai organisasi yang bermartabat, kesadaran sosial tinggi, menjunjung tinggi nilai persaudaraan, solidaritas tinggi, disegani dan terkemuka dalam dunia otomotif.

2.1.1.4 Misi

Misimerupakan pernyataan yang menunjukkan maksud didirikan / dibentuknya organisasi dan lingkup bisnis / kegiatan yang harus dijalankan atau yang justru tidak boleh dijalankan oleh organisasi. PI memiliki misi berikut:

1. Menciptakan citra positif kepada masyarakat dalam setiap kegiatan.
2. Menjaga tali persaudaraan antar sesama club otomotif, anggota dan masyarakat umum.
3. Menjadikan contoh dalam berkendara yang aman (safety) dan tetib lalulintas.
4. Sebagai wadah berkumpulnya para pengendara mobil di Indonesia.
5. Menjadi mitra Masyarakat dan pihak kepolisian.
6. Menjadi wadah dunia otomotif.

2.1.1.5 Struktur Organisasi

Dalam sebuah organisasi agar tarcapai susunan kerja yang baik ditiap kinerja anggotanya memerlukan sebuah struktur yang terencana dan dapat memperlihatkan alur kerja yang baik. Adapun struktur organisasi PI adalah sebagai berikut :

SUSUNAN PENGURUS
KOMUNITAS PICK UP INDONESIA (PI)
MASA BAKTI 2018/2019

NO	N A M A		KEDUDUKAN DALAM PERKUMPULAN
1	SUTRISYANTO	:	KETUA
2	YUSUF ISMAIL	:	WAKIL KETUA
3	HERU PRASETYO	:	SEKRETRIS
4	NUR KHAMIM	:	BENDAHARA
5	MUAWAL SHOLEH	:	BIDANG ORGANISASI
6	SUPARNO	:	BIDANG HUMAS
7	DAVID ARNANDO	:	BIDANG HUBUNGAN ANTAR LEMBAGA
8	EKO PRASETYO	:	BIDANG SARANA & PRASARANA
9	YUDIYANTO	:	BIDANG INFORMATIKA & TEKNOLOGI
10	ABI WIDJAYA	:	KOORDINATOR WILAYAH JAWA BARAT
11	TRIYONO	:	KOORDINATOR WILAYAH JAWA TENGAH & DIY
12	ARIF SULTON	:	KOORDINATOR WILAYAH JAWA TIMUR

Semarang, 25 Maret 2018

IKATAN MOTOR INDONESIA
PENGURUS PROVINSI JAWA TENGAH

AKBP. (Purn) H. KADARUSMAN
Ketua

Gambar 2. 2 Struktur Organisasi

2.1.2 Aplikasi

Aplikasi merupakan program yang berisikan perintah-perintah untuk melaksanakan pengolahan data, aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi (*application*) juga adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word dan Microsoft Excel [5].

Dari pengertian di atas dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Selain itu aplikasi dapat mengintegrasikan berbagai kemampuan komputer.

2.1.3 Android

Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari sistem operasi, *middleware*, dan aplikasi-aplikasi utama. Awalnya, Android dikembangkan oleh oleh Android Inc. Perusahaan ini kemudian dibeli oleh google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi *Open Handset Alliance* tahun 2007. Selain Google, nama-nama besar juga ikut serta dalam *Open Handset Alliance*, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba dan Intel [6].

Gambar 2. 3 Logo Android

2.1.3.1 Android SDK (Software Development Kit)

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 memberi kesempatan untuk membuat aplikasi yang dibutuhkan [6].

2.1.3.2 ADT (Android Development Tools)

Android Develpoment Tools adalah *plugin* yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan dalam membuat aplikasi *project Android*, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya. Mengembangkan aplikasi Android dengan menggunakan ADT di Eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android. Berikut adalah versi ADT untuk Eclipse yang sudah dirilis [6]:

- ADT 12.0.0 (July 2011)
- ADT 11.0.0 (June 2011)
- ADT 10.0.1 (March 2011)
- ADT 10.0.0 (February 2011)
- ADT 9.0.0 (January 2011)
- ADT 8.0.1 (December 2010)
- ADT 8.0.0 (December 2010)
- ADT 0.9.9 (September 2010)
- ADT 0.9.8 (September 2010)

- ADT 0.9.7 (May 2010)
- ADT 0.9.6 (March 2010)
- ADT 0.9.5 (December 2009)
- ADT 0.9.4 (October 2009)

Semakin tinggi platform Android yang digunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru diikuti oleh munculnya versi ADT yang terbaru.

2.1.3.3 Arsitektur Android

Secara garis besar Arsitektur Android dapat di jelaskan dan di gambarkan sebagai berikut [7] :

1. *Application and Widgets*

Application dan Widgets adalah layer dimana user berhubungan dengan aplikasi saja, dimana biasanya user men-download aplikasi, melakukan instalasi dan menjalankan aplikasi.

2. *Applications Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau member kemampuan untuk membangun aplikasi yang inovatif.

3. *Libraries*

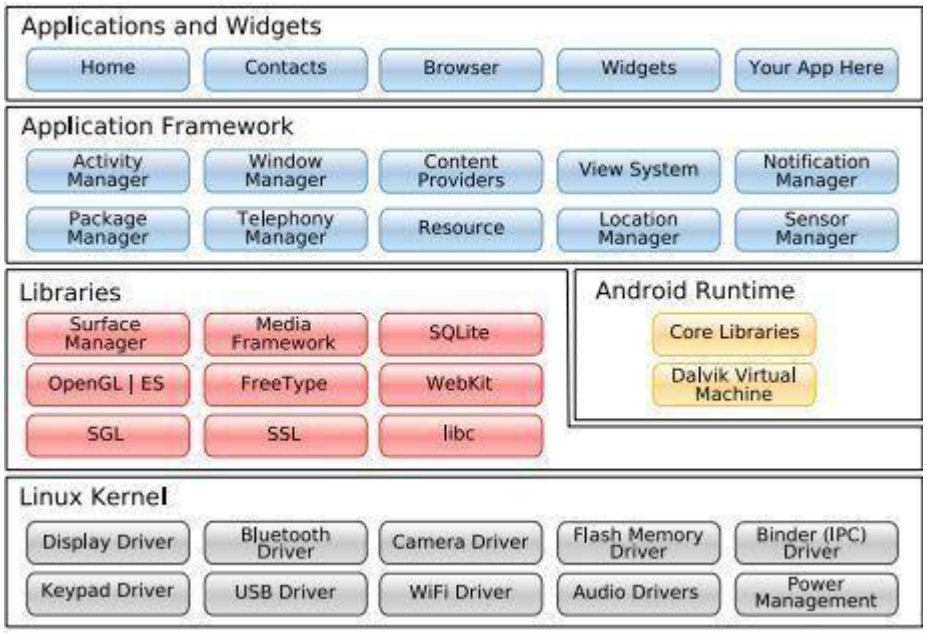
Libraries adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. *Android Run Time*

Layer yang membuat aplikasi Android dapat djalankan dimana dalam prosesnya menggunakan Implementasi Linux.

5. *Linux Kernel*

Linux Kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada. Berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resource, drivers,* dan sistem-sistem opersi Android lainnya.



Gambar 2. 4 Arsitektur Android

2.1.3.4 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data file *resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. Ada empat jenis komponen pada aplikasi Android yaitu [7]:

1) *Activities*
2) *Service*
3) *Broadcast Receiver*
4) *Content Provider*

2.1.3.5 Versi Android

Android telah mengalami sejumlah pembaruan sejak pertama kali dirilis . Rata-rata, versi terbaru dari Android dirilis setiap 6 bulan. Tabel 2.1 menunjukkan beberapa jenis Android dan nama kodenya. Penamaan kode menggunakan nama makanan dan huruf depannya urut sesuai abjad [7].

Tabel 2. 1 Daftar Android

Versi	Tanggal Rilis	Kode
1.1	9 Februari 2009	

1.5	30 April 2009	Cupcake
1.6	15 September 2009	Donut
2.0/2.1	26 Oktober 2009	Eclair
2.2	20 Mei 2010	Frozen Yoghurt (Froyo)
2.3	6 Desember 2010	Gingerbread
3.0	22 Februari 2011	Honeycomb
4.0	19 Oktober 2011	Ice Cream Sandwich
4.1	27 Juni 2012	Jelly Bean
4.2	29 Oktober 2012	Jelly Bean
4.3	24 Juli 2013	Jelly Bean
4.4	3 September 2013	Kitkat

2.1.4 Android Studio

Android Studio adalah IDE resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Di atas IntelliJ yang kuat *code editor* dan pengembang alat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda ketika membangun aplikasi Android, seperti [8] :

1. Sebuah *fleksibel* berbasis *Gradle* membangun sistem
2. Membangun varian dan beberapa generasi file APK
3. Kode template untuk membantu Anda membangun fitur aplikasi umum
4. Sebuah *layout editor* kaya dengan dukungan untuk *drag* dan *drop tema editing*
5. alat *Lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya
6. Kode menyusut dengan ProGuard dan sumber daya menyusut dengan Gradle
7. Built-in dukungan untuk Google *Cloud Platform*, sehingga mudah untuk mengintegrasikan Google *Cloud Messaging* dan App *Engine*.

2.1.5 Java

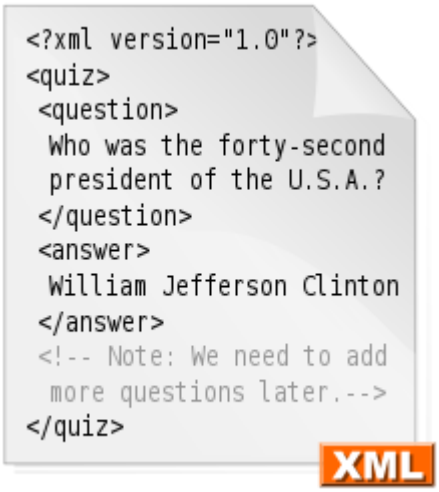
Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat 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). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web [9].

Gambar 2. 5 Logo Java

2.1.6 XML (*Extensible Markup Language*)

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet [10].



Gambar 2. 6 Contoh Dokumen XML

2.1.7 PHP (*Hypertext Preprocessor*)

PHP adalah sebuah bahasa pemograman yang berjalan dalam sebuah *web server* (*server side*). PHP adalah bahasa skrip yang sangat populer digunakan untuk mengembangkan aplikasi-aplikasi web, meskipun sebenarnya PHP juga dapat digunakan untuk mengembangkan aplikasi desktop, baik *console* maupun yang berbasis GUI [11].

2.1.9 Web Service

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan web service menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML [12].

Definisi diatas diberikan oleh *World Wide Web Consortium*(W3C) yang merupakan badan yang menciptakan dan mengembangkan standar *web service*. Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui internet menggunakan protokol standar internet dan menggunakan XML sebagai format pesannya.

2.1.10 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman 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. Kelebihan-kelebihan dari JSON adalah [13]:

1. Format Penulisan : Untuk merepresentasikan sebuah struktur data yang rumit dan berbentuk hirarkis penulisan JSON relatif lebih terstruktur dan mudah.
2. Ukuran karakter yang dibutuhkan JSON lebih kecil dibandingkan XML untuk data yang sama.
3. Proses parsing merupakan proses pengenalan token atau bagian-bagian kecil dalam rangkaian dokumen XML/JSON.

   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*).

2.1.11 API (*Application Programming Interface*)

API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi. API atau *Application Programming Interface* juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak.

Dengan adanya API, maka memudahkan seorang *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau

diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan *developer* menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi [14].

Keuntungan dengan menggunakan API adalah sebagai berikut:

1. Portabilitas.

Developer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut.

2. Lebih Mudah Dimengerti

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal editing dan pengembangan.

Cara menggunakan API:

1) Dilakukan dengan mengimpor *package*/kelas.
2) Ada beberapa kelas bernama sama dipackage yang berbeda, yaitu:

a. Import salah satu dan gunakan nama lengkap untuk yang lain.
b. Gunakan nama lengkap semua kelas

Kebanyakan Sistem Operasi seperti Windows, menyediakan fasilitas API sehingga *programmer* dapat melakukan aktivitas programming dengan lebih konsisten. Meskipun API didesain untuk programer, namun API juga baik untuk user karena setidaknya dapat menjamin bahwa program tersebut memiliki interface yang sama, sehingga lebih mudah untuk dipelajari.

2.1.12 GPS (*Global Position Service*)

GPS (*Global Position Service*) adalah sistem navigasi berbasis satelit yang dikembangkan oleh U.S. Department of Defense (DoD). pada awal tahun 1970 awalnya GPS dikembangkan sebagai sistem militer untuk memenuhi kebutuhan militer. Namun, yang terakhir dibuat tersedia untuk warga sipil, dan sekarang sistem penggunaan ganda yang dapat diakses oleh pengguna militer dan sipil. GPS memberikan posisi terus menerus dan informasi waktu setiap tempat di dunia di bawah setiap kondisi cuaca. Karena, melayani jumlah pengguna yang tidak

terbatas serta digunakan untuk alasan keamanan. GPS menggunakan *one-way-ranging* (*passive*) sistem. Artinya, pengguna hanya dapat menerima sinyal satelit [15].

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS *receiver*, sehingga harus selalu dipelihara agar posisinya tepat. Posisi satelit-satelit tersebut selalu dipantau oleh stasiun pengendali. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS *receiver* mengambil informasi tersebut dan melakukan perhitungan triangulation untuk menentukan lokasi pengguna dengan tepat. GPS *receiver* membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit. Dengan mengetahui jarak tersebut, GPS *receiver* dapat melakukan perhitungan dan menentukan posisi pengguna dan menampilkan dalam peta elektronik. Setelah menentukan posisi pengguna, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS sangat presisi dalam mengirim informasi karena satelit tersebut menggunakan jam atom. Jam atom yang ada pada satelit merupakan partikel atom yang diisolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Keistimewaan GPS adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat GPS bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi *Wide Area Augmentation System* (WAAS) keakuratannya sampai 3 meter. Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan di tempat terbuka. Penggunaan di dalam ruangan, atau di tempat yang menghalangi arah satelit (di angkasa), maka GPS tidak akan bekerja secara akurat dan maksimal. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi. Semakin banyak sinyal satelit yang dapat diterima maka akan

semakin presisi data posisi yang dihasilkan. Selain itu, ketinggian juga mempengaruhi proses kerja GPS, karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit.

2.1.13 LBS (*Location Based Service*)

LBS merupakan konsep baru yang menunjukkan aplikasi yang mengintegrasikan lokasi geografis (yaitu, spasial koordinat) dengan gagasan umum layanan. Konsep LBS dapat didefinisikan sebagai layanan yang mengintegrasikan lokasi perangkat mobile atau posisi dengan informasi lain sehingga dapat memberikan nilai tambah untuk pengguna. Lokasi layanan telah ada sejak tahun 1970-an dengan kedudukan global di seluruh dunia yang dikenal sebagai GPS(*Global Position Service*) yang dikembangkan oleh Departemen Pertahanan Amerika Serikat. Tapi dalam dekade berikutnya ketika pemerintah luar AS memutuskan untuk membuat data posisi sistem bebas tersedia untuk industri lainnya di seluruh dunia, yang industri ini telah diambil sampai kesempatan untuk mengakses data posisi melalui GPS dan sekarang menggunakannya untuk meningkatkan produk dan layanan mereka [3].

2.1.14 Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghidari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga update yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system* | DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan user untuk memelihara, mengontrol, dan mengakses

data secara praktis dan efisien. Dengan kata lain semua akses ke basis data akan ditangani oleh DBMS. Ada beberapa fungsi yang harus ditangani DBMS yaitu mengolah pendefinisian data, dapat menangani permintaan pemakai untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (*Database Administrator*), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun disk, dan menangani unjuk kerja semua fungsi secara efisien.

Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada user (pengguna). Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil (akses) secara efisien. Pertimbangan efisien disini adalah bagaimana merancang struktur data yang kompleks tetapi masih tetap bisa digunakan oleh pengguna awam tanpa mengetahui kompleksitas strukturnya. Database dibagi menjadi dua jenis yaitu [16]:

1. Basis Data Flat File

Basis data flat-file ideal untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada dasarnya, mereka tersusun dari sekumpulan string dalam satu atau lebih file yang dapat diurai untuk mendapatkan informasi yang disimpan. Basis data flat-file baik digunakan untuk menyimpan daftar atau data yang sederhana dan dalam jumlah kecil. Basis data flat-file akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks walaupun dimungkinkan pula untuk menyimpan data semacam itu. Salah satu masalah menggunakan basis data jenis ini adalah rentan pada korupsi data karena tidak adanya penguncian yang melekat ketika data digunakan atau dimodifikasi.

2. Basis Data Relasional

Basis data ini mempunyai struktur yang lebih logis terkait cara penyimpanan. Kata "relasional" berasal dari kenyataan bahwa tabel-tabel yang berada di basis data dapat dihubungkan satu dengan lainnya. Basis data relasional menggunakan sekumpulan tabel dua dimensi yang masing-masing tabel tersusun atas baris (*tupel*) dan kolom (atribut). Untuk membuat hubungan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu *primary key* di salah satu tabel dan *foreign key*

di tabel yang lain. Saat ini, basis data relasional menjadi pilihan karena keunggulannya. Beberapa kelemahan yang mungkin dirasakan untuk basis data jenis ini adalah implementasi yang lebih sulit untuk data dalam jumlah besar dengan tingkat kompleksitasnya yang tinggi dan proses pencarian informasi yang lebih lambat karena perlu menghubungkan tabel-tabel terlebih dahulu apabila datanya tersebar di beberapa tabel.

2.1.15 MySQL

MySQL merupakan *software database* yang paling populer di kalangan para pengembang aplikasi database. MySQL juga adalah suatu perangkat lunak yang menganut atau mengimplementasikan model basis data relasional maka MySQL disebut sebagai *Relational Database Management System* (RDBMS). MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*) [17].

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar. Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk open source dibawah GNU *General Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. Beberapa contoh aplikasi yang menggunakan MySQL adalah Joomla, Wordpress, MyBB, phpBB, dan masih banyak lagi.

2.1.16 HTTPS (Hypertext Transfer Protocol)

HTTPS (*Hypertext Transfer Protocol Secure*) adalah protokol komunikasi internet yang melindungi integritas dan kerahasiaan data pengguna antara komputer pengguna dan situs. Data yang dikirim menggunakan HTTPS diamankan melalui protokol *Transport Layer Security* (TLS), yang memberikan tiga lapis perlindungan kunci [18] :

1. Enkripsi: mengenkripsi data pertukaran untuk menjaga keamanannya dari penyadap. Artinya, saat pengguna menjelajahi situs web, tidak ada yang dapat "menguping" percakapan, melacak aktivitas di berbagai laman, atau mencuri informasi mereka.
2. Integritas data: data tidak dapat diubah atau dirusak selama transfer, dengan sengaja atau tidak, tanpa terdeteksi.
3. Autentikasi: membuktikan bahwa pengguna Anda berkomunikasi dengan situs web yang diinginkan. Hal tersebut melindungi dari serangan *Man In The Middle* (MITM) dan membangun kepercayaan pengguna, yang dapat memberikan keuntungan lain untuk bisnis Anda.

2.1.17 OOAD (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 [19].

2.1.18 OOP (Object Oriented Programming)

Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat

beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme) dan *inheritance* (pewarisan) [20].

2.1.19 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (*Object Management Group*) [19].

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya yaitu : *Grady Booch*OOD (*Object Oriented Design*), James Rumbaugh OMT (*Object Modeling Technique*) dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*).

2.1.19.1 *Use Case* Diagram

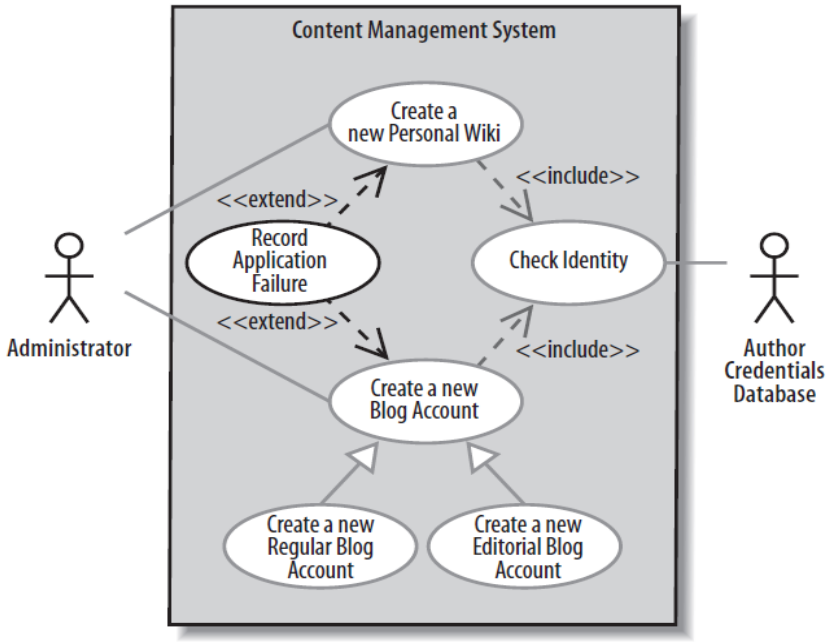
Use case Diagram merupakan salah satu diagram untuk memodelkan aspek perilakau sistem. Masing-masing diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. Diagram *use case* adalah penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram-diagram *use case* merupakan pusat pemodelan perilaku sistem, subsistem, dan kelas. Diagram *use case* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Ada empat hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan use case [19] :

1. Sistem yaitu sesuatu yang hendak kita bangun.
2. Relasi adalah relasi antara aktor dengan *use case*.
3. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu

sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

4. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut Contoh *Use Case* Diagram dapat dilihat pada Gambar 2.7.



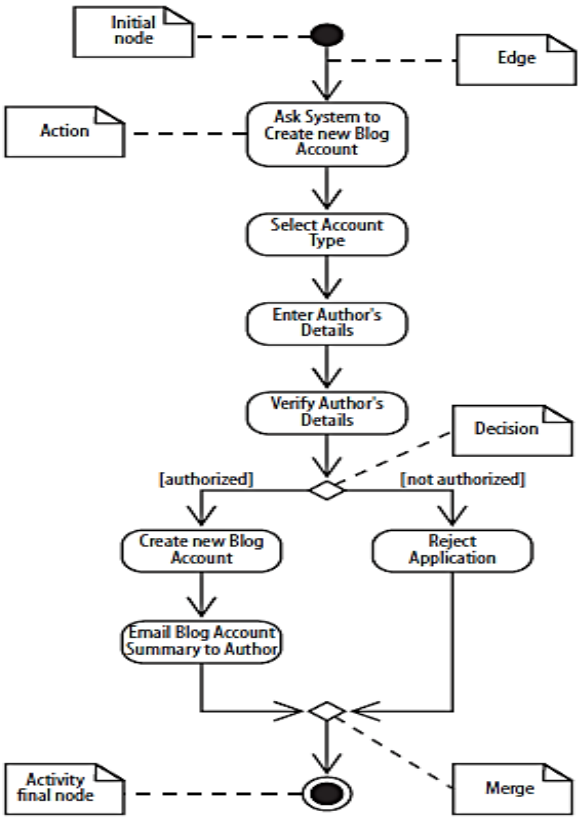
Gambar 2. 7 Contoh Use Case Diagram

2.1.19.2 *Activity* Diagram

Diagram aktivitas adalah diagram *flowchart* yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut [19]:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut Contoh *Activity* Diagram dapat dilihat pada Gambar 2.8.

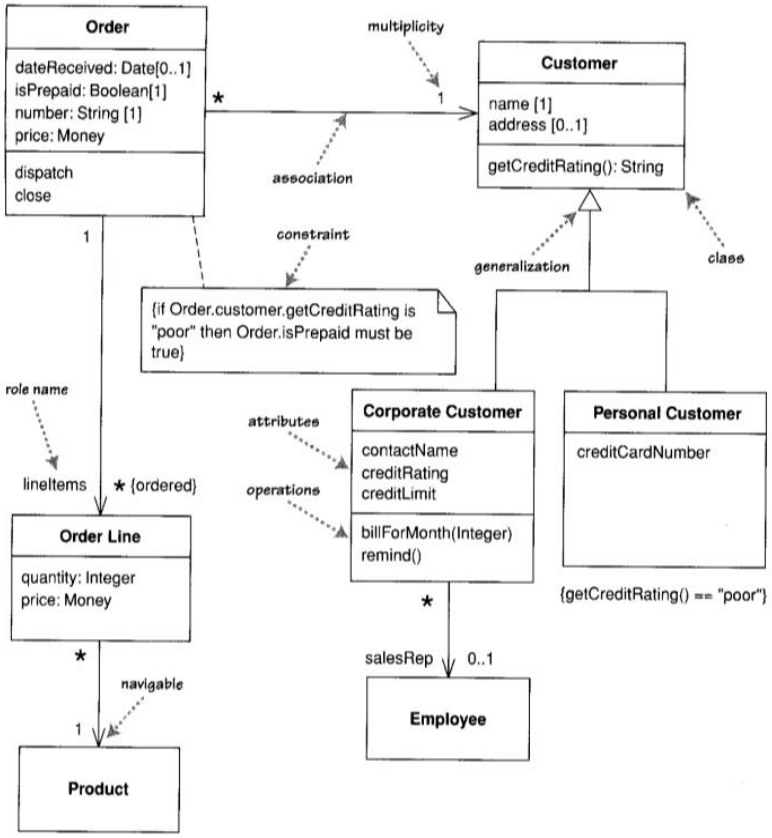


Gambar 2. 8 Contoh Activity Diagram

2.1.19.3 *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* [19].

Berikut contoh *Class* Diagram dapat dilihat pada Gambar 2.9.

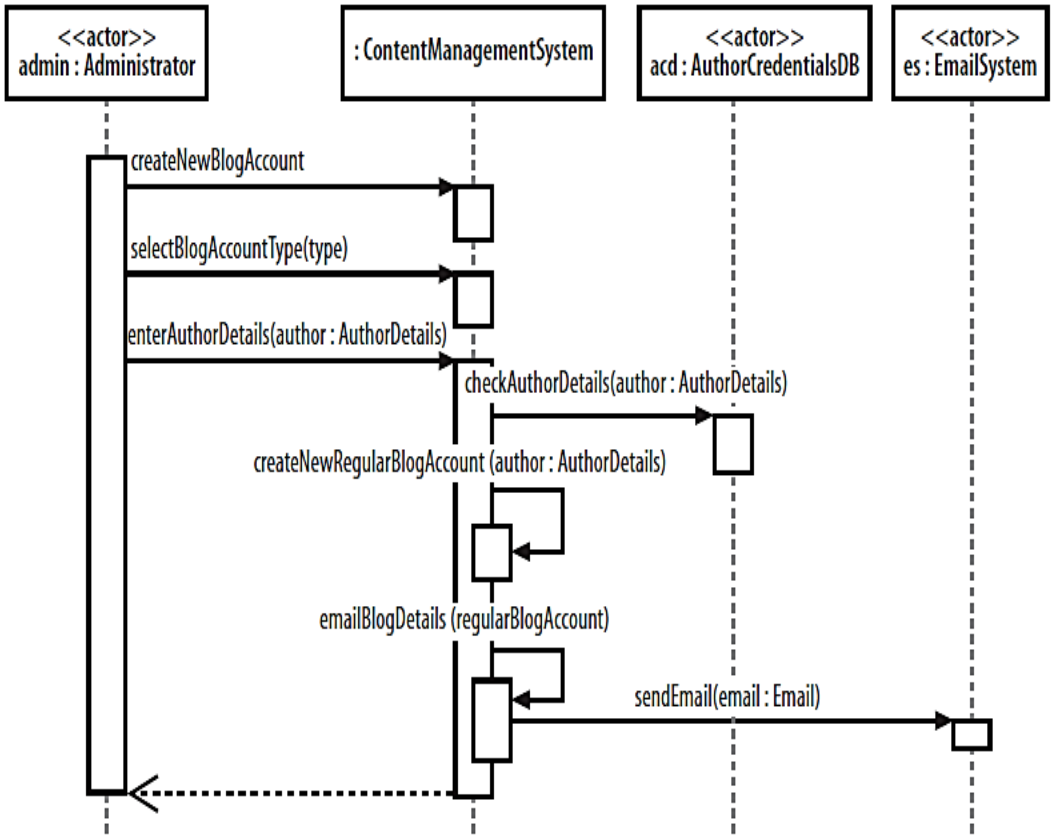


Gambar 2. 9 Contoh Class Diagram

2.1.19.4 *Sequence* Diagram

Sequence Diagram menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada sisi basis keberurutan waktu dari pesan-pesan yang terjadi. Diagram sekuen menunjukan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukan barisan kejadian, bukan perwaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan terjadi [19].

Berikut Contoh *sequence* diagram dapat dilihat pada gambar 2.10.



Gambar 2. 10 Contoh Sequence Diagram