



BAB II

Tinjauan Pustaka

2.1 Teori Umum

2.1.1 Pengertian Komputer

Hartono(2013:27), “Komputer adalah sebuah mesin yang dapat dikendalikan melalui perintah (programmable machine) yang dirancang secara otomatis melakukan serangkaian urutan perhitungan (arithmetic) atau proses-proses yang diurutkan secara logis”.

Sutarman (2012:86),Komputer adalah suatu rangkaian peralatan elektronik yang bekerja secara bersama-sama”. Komputer dapat melakukan rangkaian pekerjaan secara otomatis melalui instruksi (program)yang diberikan dan alat pengolah data menjadi informasi melalui proses tertentu.

Wahyudi (2012:3), “Komputer adalah peralatan (device) yang menerima data (input) dan menyimpan (storage) kemudian diproses (process) untuk menghasilkan data dalam bentuk lain(output)”.

Dari pengertian-pengertian di atas, penulis menyimpulkan bahwa komputer adalah seperangkat alat elektronik yang digunakan untuk melakukan pemrosesan data dan menghasilka sebuah output.

2.1.2 Pengertian Internet

Sarwono (2012), “Internet merupakan sebuah kumpulan jaringan yang memiliki skala global. ”.

Sujatmiko (2012:138), “Internet adalah interconnected Network jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit”.

Dapat disimpulkan bahwa definisi dari Internet merupakan hubungan antara berbagai jenis komputer dan jaringan yang ada diseluruhdunia meskipun berbeda sistem operasi dan mesinnya.



2.1.3 Pengertian Android

Juhara,(2016:1), “Android adalah sistem operasi berbasis Linux yang dimodifikasi untuk perangkat bergerak (mobile device) yang terdiri dari sistem operasi, middleware, dan aplikasi-aplikasi utama.”

Salbino (2010:7) , “Android merupakan sistem operasi berbasis Linux yang bersifat terbuka (open source) dan dirancang untuk perangkat seluler layar sentuh seperti smartphone dan komputer tablet.”

Dari beberapa definisi diatas penulis menyimpulkan bahwa android adalah sistem operasi berbasis Linux yang bersifat terbuka (open source) untuk perangkat mobile (smartphone) selain mobile khusus seperti Windows Phone dan iOS. Mobile Phone yang digunakan harus bersifat touchscreen sehingga sesuai dengan kegunaan android yaitu memudahkan.

2.2 Teori Judul

2.2.1 Implementasi

Kamus Besar Bahasa Indonesia (KBBI), Implementasi adalah pelaksanaan atau penerapan. Sementara, Nurdin Usman (2002:70) berpendapat bahwa implementasi bermuara pada aktivitas, aksi, tindakan, atau adanya mekanisme suatu sistem. Implementasi bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan.

Dari kedua pengertian diatas dapat disimpulkan bahwa implementasi merupakan suatu penerapan sistem pelaksanaan yang terencana untuk mencapai tujuan tertentu.

2.2.2 Pariwisata

Menurut Undang-Undang RI nomor 10 tahun 2009 tentang kepariwisataan dijelaskan bahwa wisata adalah kegiatan perjalanan yang dilakukan oleh seseorang atau sekelompok orang dengan mengunjungi tempat tertentu untuk tujuan rekreasi, pengembangan pribadi, atau mempelajari keunikan daya tarik wisata yang dikunjungi dalam waktu sementara.



2.2.3 Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

2.2.4 Location Based Service (LBS)

Location Based Service (LBS) adalah layanan informasi yang dapat diakses menggunakan piranti *mobile* melalui jaringan internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile* (Virrantasu, et al 2001).

Terdapat empat komponen pendukung utama dalam LBS, antara lain :

1. Piranti *Mobile*

Piranti *Mobile* adalah salah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu (tool) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Piranti *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, *laptop*, dan PC. Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi berbasis *GPS*.

2. Jaringan Komunikasi

Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirim data-data yang dikirim oleh pengguna dari piranti *mobile* untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

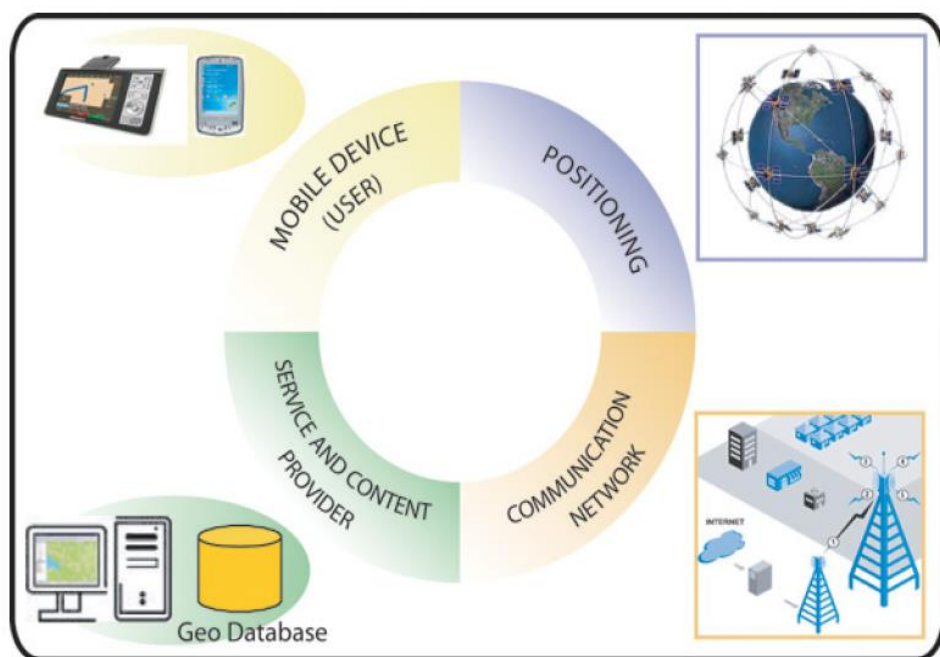
3. Komponen *Positioning* (Penunjuk Posisi/Lokasi)

Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).



4. Penyedia Layanan dan Konten (*Service and Content Provider*)

Penyedia layanan merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tau posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di Yellow Pages sesuai dengan permintaan, dan masih banyak lagi yang lain.



(Sumber : Anwar Badrul, dkk. 2014. *Implementasi Location Based Service Berbasis Android Untuk Mengetahui Posisi User*. Jurnal Ilmiah SAINTIKOM. Vol. 13, No. 2, Mei 2014.)

Gambar 2.1 Komponen Pendukung LBS

Location Based Service (LBS) terdiri dari 2 metode, yaitu :

a. *Device based* (*Location base service* berdasarkan alat)



Metode ini berdasarkan alat GPS (*Global Positioning System*) yang ada di ponsel atau device lainnya. Ponsel mengandalkan kemampuan dari device GPS. Apa itu GPS, bisa kita bahas di artikel selanjutnya.

b. *Network Base* (Berdasarkan Jaringan)

Terdapat beberapa metode pada network base ini.

1. *Cell of Origin*

Metode ini adalah metode yang paling sederhana. Bekerja berdasarkan cell jaringan yang ada, di mana masing-masing cell dijangkau oleh satu BTS (*Base Tower Station*). Lokasi ditentukan berdasarkan letak dari ponsel yang berada pada cell. Jadi penentuan Lokasi dari ponsel akan terdeteksi dari BTS yang menjangkanya. Metode ini adalah metode paling sederhana, namun mempunyai tingkat akurasi yang rendah, sebab hanya mengandalkan satu BTS dan ponsel untuk mendapatkan sudut lokasi.

2. *Angle of Arrival* (AOA)

Metode ini membutuhkan minimal tiga BTS. Penentuan lokasi didapatkan dari sudut (*angle*) yang dibentuk oleh Ponsel dengan BTS-BTS yang menjangkanya. Sudut ini biasa disebut Triangulasi (*segitiga*). Tiga BTS akan membentuk segitiga, dan setiap BTS akan mendeteksi lokasi ponsel berdasarkan garis sudutnya. Metode ini lebih akurat dibanding *Cell of Origin*.

3. *Time difference of Arrival* (TDOA)

Metode ini berdasarkan pada perbedaan waktu kedatangan. Jadi informasi lokasi dihasilkan dari perpotongan jarak bujur lingkaran yang dibentuk atas perbedaan waktu kedatangan yang dihasilkan. Metode ini memiliki tingkat akurasi yang lebih baik.

4. *Enhanced Observed Time Difference* (TDOA)

Metode ini adalah pengembangan dari metode *Difference Of Arrival* (TDOA). Perbedaanannya adalah Tiga BTS akan selalu mengamati perbedaan sinyal waktu perjalanan ponsel yaitu dengan cara mentransmisikan sinyal ke ponsel tersebut. Metode ini membutuhkan perangkat tambahan pada ponsel.



5. *Location Pattern Matching (LPM)*

Metode ini berdasarkan lokasi dari ponsel. Cara kerjanya adalah BTS menganalisis lokasi ponsel tersebut dari pantulan gelombang elektromagnetik yang dihasilkan oleh ponsel terhadap kondisi lokasi daerah. Hasil analisis berupa tekstur daerah. Lalu BTS membandingkan tekstur daerah tersebut dengan database yang sudah ada. Database ini berisi dari tekstur area. Metode ini mengandalkan dari database lokasi dari BTS-BTS.

6. *Hybrid Method*

Metode ini mengkombinasikan antara metode *device based* (GPS) dengan *Network Based*. Pada proses pemanfaatannya, setiap vendor smartphone mempunyai wewenang masing-masing untuk memilih metode mana yang digunakan.

2.3 Teori Khusus

2.3.1 Pengertian *UML (Unified Modeling Language)*

Sukamto dan Shalahuddin (2013:133), “UML (Unified Modeling Language) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an. Pada 1996, Object Management Group (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 Unified Modeling Language (UML) diakomodasi oleh Object Management Group (OMG) sehingga sampai saat ini Unified Modeling Language (UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.



2.3.2 Use case Diagram

Sukamto dan Shalahuddin (2013:155), “Use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.


Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

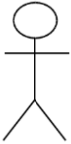

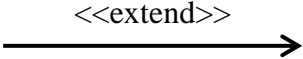
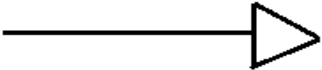

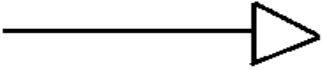
Berikut adalah simbol-simbol yang ada pada diagram use case menurut Rosa dan Shalahuddin (2013:156):

Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Keterangan
-----	--------	------------

No.	Simbol	Keterangan
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case



2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
3.	<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor</p>
4.	<p>Ekstensi / <i>extend</i></p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
5.	<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi use case tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case.</p>


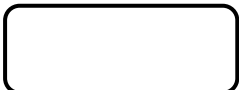
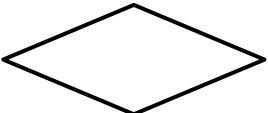



Sumber : Rosa A.S dan M. Shalahudin (2014:165)


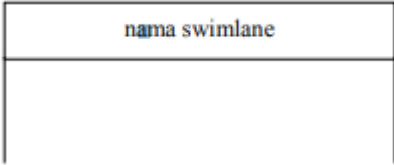
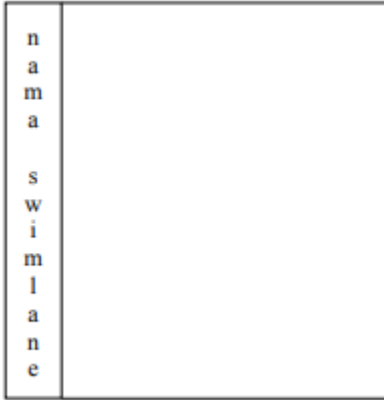
2.3.3 Activity Diagram

Sukanto dan Shalahuddin (2013:161), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut Rosa dan Shalahuddin (2013:162):

Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Keterangan
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu



No.	Simbol	Keterangan
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	Swimlane  atau 	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rosa A.S dan M. Shalahudin (2014:165)

2.3.4 Class Diagram

Sukanto dan Shalahuddin (2013:141), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:



1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian use case (*controller*)

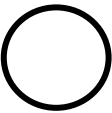

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

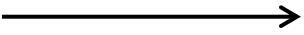
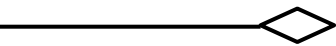
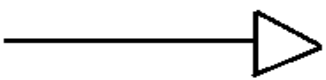

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Rosa dan Shalahuddin (2013:146):

Tabel 2.3 Simbol-simbol *class diagram*

No.	Simbol	Keterangan
1.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p>nama_kelas</p> <hr/> <p>+ atribut</p> <hr/> <p>+ operasi()</p> </div>	Kelas pada struktur sistem
2.	<p>Antarmuka / <i>interface</i></p> <div style="text-align: center;">  <p>nama_interface</p> </div>	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	<p>Asosiasi / association</p> <div style="text-align: center;">  </div>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity



No.	Simbol	Keterangan
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
5.	Agregasi / aggregation 	Relasi antarkelas dengan makna semua- bagian (whole-part)
6.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
7.	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas

Sumber : Rosa A.S dan M. Shalahudin (2014:165)

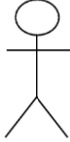
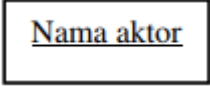

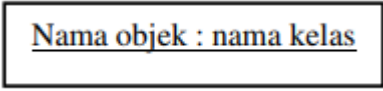

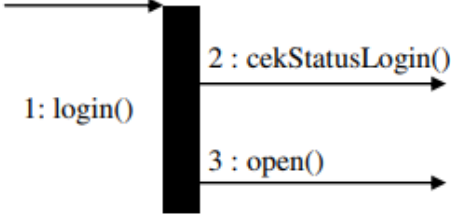
2.3.5 Sequence Diagram

Sukanto dan Shalahuddin (2013:165), “Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek”.



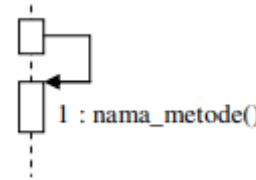
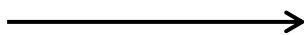
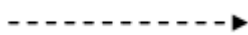
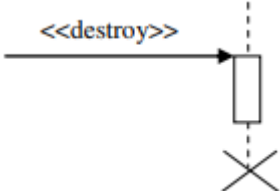
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen menurut Rosa dan Shalahuddin (2013:146):

Tabel 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>tanpa waktu aktif</p>	<p>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, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis Hidup / lifeline</p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>



No.	Simbol	Keterangan
5.	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 
		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

Sumber : Rosa A.S dan M. Shalahudin (2014:165)



2.3.6 Pengertian Kamus Data (Data Dictionary)

Sukamto dan Shalahuddin (2014:73) Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Sukamto dan Shalahuddin (2014:73), menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

Tabel 2.5 Simbol-simbol dalam Kamus Data

No	Notasi	Arti
1.	=	disusun atau terdiri dari
2.	+	<i>Dan</i>
3.	[]	baik... atau...
4.	{ } ⁿ	n kali diulang/bernilai banyak
5.	()	data operasional
6.	*...*	Batas komentar

(Sumber: Sukamto dan Shalahuddin, 2014:74)

2.4 Teori Program

2.4.1 Pengertian *Restful API*

Restful API merupakan implementasi dari *API* (*Application Programming Interface*). *REST* (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan terutama dalam pertukaran dan komunikasi data. Konsep *REST* pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Cara



kerjanya, *REST server* menyediakan jalur untuk akses *resource* atau data, sedangkan *REST client* melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan sebenarnya berupa teks, namun formatnya bisa bermacam-macam tergantung keinginan developer, umumnya adalah JSON dan XML.

Restful API memiliki 4 komponen penting di dalamnya diantaranya adalah:

1. URL Design
2. HTTP Verbs
3. HTTP Response Code
4. Format Response

Beberapa keuntungan dari *Restful API* yaitu:

- a. Lebih sederhana/simpel untuk dikembangkan ketimbang SOAP.
- b. Mudah dipelajari, tidak bergantung pada tools.
- c. Ringkas, tidak membutuhkan layer pertukaran pesan tambahan.
- d. Secara desain dan filosofi lebih dekat dengan web.

2.4.2 Pengertian Java

Sukanto dan Shalahuddin (2013:103) menjelaskan, Java adalah bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas.

Menurut Asropudin (2013:52), Java adalah bahasa pemrograman untuk menciptakan isi yang aktif dalam halaman web, juga dapat dijalankan dalam semua komputer.

Sedangkan menurut Siallagan (2009:13), Bahasa pemrograman java adalah bahasa pemrograman berorientasi objek (PBO) atau Object Oriented Programming (OOP). Java bersifat netral, tidak bergantung pada suatu platform, dan mengikuti prinsip WORA (Write Once and Run Anywhere).



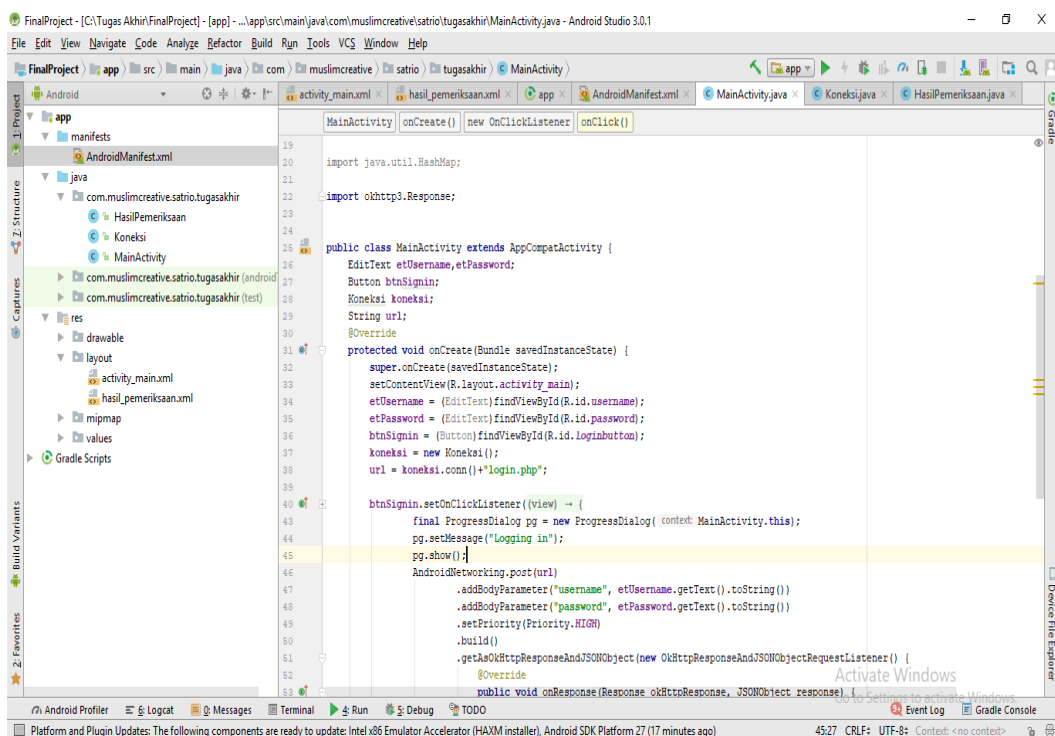
2.4.3 Pengertian *Object Oriented Programming (OOP)*

Sukanto dan Shalahuddin (2013:100), Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Dari definisi diatas dapat disimpulkan bahwa pemrograman berorientasi objek adalah program komputer dari berbagai objek yang melakukan suatu tindakan terhadap masing-masing objek.

2.4.4 Pengertian *Android Studio*

Android Studio adalah sebuah *Integrated Development Environment (IDE)* untuk *platform* Android. *Android Studio* tersedia bagi pengembang untuk mencoba secara gratis. *Android Studio* berada di awal tahap *preview access* mulai dari versi 0.1 pada Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Berdasarkan software JetBrains *IntelliJ IDEA*, *Android Studio* dirancang khusus untuk pengembangan Android.



(Sumber : http://en.wikipedia.org/wiki/Android_Studio)

Gambar 2.2 Tampilan Android Studio

2.5 Penelitian Terkait

Berdasarkan penelitian yang telah dilakukan oleh Berbudhi Rachman Hidayat dan Heryn Februariyanti pada tahun 2013 dengan judul Aplikasi *Location Based Service* (LBS) Pencarian Lokasi Taxi Pada Android Di Kota Semarang. Dengan adanya aplikasi ini pengguna dapat dengan mudah memesan taxi dan melihat lokasi basecamp taxi terdekat.

Ahmad Fauzi pada tahun 2015 melakukan penelitian dengan judul Penerapan *Location Based Service* Pada Layanan Informasi Budaya Indonesia Di Perangkat *Mobile*. Dengan adanya aplikasi ini seseorang akan dapat mengetahui budaya setempat berdasarkan lokasi keberadaannya dan juga memberikan informasi mengenai beberapa tempat yang mungkin belum terdaftar dalam sistem.

Menurut Nova Agustina, Slamet Risnanto, dkk pada tahun 2016 dengan judul Pengembangan Aplikasi *Location Based Service* Untuk Informasi Dan



Pencarian Lokasi Pariwisata Di Kota Cimahi Berbasis Android. Aplikasi *Location Based Service* yang dibuat dapat membantu *user* untuk mencari informasi, lokasi pariwisata dan juga menentukan rute yang dapat ditempuh menuju lokasi pariwisata yang ingin dikunjungi.

Dedy Hidayat Kusuma dan Moh. Nur Shodiq pada tahun 2017 dengan judul Sistem Rekomendasi Destinasi Pariwisata Menggunakan *Hibrid Case Based Reasoning* dan *Location Based Service* Sebagai Pemandu Wisatawan Di Banyuwangi. Hasil yang didapat adalah menyediakan sistem yang dapat membantu wisatawan dalam memilih objek wisata di Banyuwangi sesuai dengan preferensi atau kriteria permintaannya.

Badrul Anwar, Hendra Jaya, dan Putra Indra Kusuma pada tahun 2014 dengan judul Implementasi *Location Based Service* Berbasis Android Untuk Mengetahui Posisi *User*. Hasil dari pembuatan aplikasi adalah Menyediakan sebuah aplikasi smartphone berbasis android yang dapat mengetahui keberadaan lokasi user menggunakan *location based service*.