

Bab 2

LANDASAN TEORI

Pada bab ini dijabarkan mengenai bahasa Pemrograman Aplikasi dan *Database*. Dalam aplikasi Java dan PHP digunakan sebagai bahasa pemrograman utama, sedangkan UML dan struktur navigasi adalah media perancangan aplikasi.

2.1 Jurnal Penelitian Terkait

Aplikasi ini menggunakan fitur LBS (*Location Based Services*) dan menggunakan GIS (*Geographic Information System*). Terdapat jurnal terkait yang dapat menjadi acuan penulisan dan untuk memperkuat penelitian, seperti pada Tabel 2.1.

Tabel 2.1 Jurnal Penelitian Terkait

No	Nama Peneliti	Judul	Kelebihan	Kekurangan
1	Cheverst, K., Davies, N., Mitchell, K., Friday, A., & Efstratiou, C. (2000). <i>CHI '00</i> https://doi.org/10.1145/332040.332047	Developing a context-aware electronic tourist guide. In <i>Proceedings of the SIGCHI conference on Human factors in computing systems</i> .	Aplikasi ini digunakan untuk memberikan informasi tempat wisata dan telah ditanamkan sistem pakar untuk menentukan berdasarkan kriteria yang dibutuhkan turis, mulai dari	Aplikasi ini belum menampilkan maps, aplikasi ini masih menggunakan DGPS dan belum dapat mencakup daerah secara luas. Tampilan yang masih sangat sederhana.

No	Nama Peneliti	Judul	Kelebihan	Kekurangan
			apa yang ingin dikunjungi, lalu cuaca hari ini, apa yang dapat dikunjungi di sekitar turis.	
2	Kumar, S., Qadeer, M., & Gupta, a. (2009). https://doi.org/10.1109/IMSAA.2009.5439442	Location based services using android (LBSOID)	Aplikasi ini sudah memanfaatkan GIS sebagai penyedia layanan informasi. Sudah menggunakan fitur LBS dan telah di uji coba pada Android.	Tampilan pada aplikasi ini masih sederhana dan masih menggunakan basis data local atau SQLite sehingga data tidak tercentral dan dinamis.
3	Hess, R. L., Rubin, R. S., & West, L. A. (2004).	Geographic information systems as a marketing information system technology. <i>Decision Support Systems</i>	Penelitian tersebut menjelaskan secara rinci manfaat dan penggunaan GIS untuk pengambilan keputusan. Pada penelitian ini GIS sudah di	Penelitian ini belum di implementasikan pada sebuah sistem. Pada penelitian ini kurang lengkapnya tampilan system secara mendetail hanya terdapat

No	Nama Peneliti	Judul	Kelebihan	Kekurangan
			gabungkan dengan MKIS (<i>Marketing Information System</i>) sehingga dapat menghasilkan MKDSS (<i>Marketing Decision Support</i>)	konsep secara garis besar.

2.2 Survei

Spesifikasi data yang dibutuhkan dalam survei di tentukan oleh keterbutuhan dalam kontek kegunaannya. Survei fisik tentu dibutuhkan metode teknis pendataan dan bagaimana cara mengumpulkan data tersebut.

2.3 Metode Pengumpulan Data fisik

Metode pengumpulan data menurut ahli metode pengumpulan data berupa suatu pernyataan (statement) tentang sifat, keadaan, kegiatan tertentu dan sejenisnya. Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian (Gulo, 2002 : 110).

Metode pengumpulan data ini termasuk kategori laporan diri (personal report) / Deskripsi diri (self descriptive). Individu melaporkan tentang keadaan dirinya berdasarkan pertanyaan atau perintah yang diberikan kepadanya.

Sumber data dan metode pengumpulan data ada dua sumber data dan metode pengumpulan data, dua hal tersebut yaitu :

1. Data Primer

Data penelitian yang diperoleh sendiri melalui proses pengamatan langsung lapangan. Seperti :

- a. Wawancara, Observasi, Tes,
- b. Kuesioner (Daftar Pertanyaan)
- c. Pengukuran Fisik

2. Data Sekunder

Data yang diperoleh dari sumber kedua, dokumentasi lembaga, artinya penulis hanya menggunakan dokumentasi/literatur dari suatu Lembaga tertentu, seperti :

- a. Biro Pusat Statistik (BPS)
- b. Rumah sakit
- c. Lembaga atau institusi

Metode Pengumpulan Data :

Seperti tulisan diatas bahwa pengamatan langsung lapangan melalui cara observasi, wawancara, foto atau dokumentasi fisikm dan kuisoner.

a. Observasi

Pengamatan melibatkan semua indera (penglihatan, pendengaran, penciuman, pembau, perasa). Pencatatan hasil dapat dilakukan dengan bantuan alat rekam elektronik.

b. Wawancara

Pengambilan data melalui wawancara /secara lisan langsung dengan sumberdatanya, baik melalui tatap muka atau lewat telephone, teleconference. Jawaban responden direkam dan dirangkum sendiri oleh peneliti.

c. Dokumen

Pengambilan data melalui dokumen tertulis mamupun elektronik dari lembaga/institusi. Dokumen diperlukan untuk mendukung kelengkapan data yang lain.

d. Kuesioner

Kuesioner adalah daftar pertanyaan tertulis yang ditujukan kepada

responden. Jawaban responden atas semua pertanyaan dalam kuesioner kemudian dicatat/direkam.

2.4 Bangun Gedung

Obyek pengamatan dalam pengembangan metode survei dikhususkan pada obyek bangunan gedung negara atau milik pemerintah. Beberapa pemahaman terkait dengan bangunan gedung pemerintah secara definitif berdasarkan undang-undang maupun peraturan, apa itu bangunan gedung, bangunan negara serta klasifikasinya.

2.4.1 Bangunan gedung

Bangunan gedung adalah wujud fisik hasil pekerjaan konstruksi yang menyatu dengan tempat kedudukannya, sebagian atau seluruhnya berada di atas dan/atau di dalam tanah dan/atau air, yang berfungsi sebagai tempat manusia melakukan kegiatannya, baik untuk hunian atau tempat tinggal, kegiatan keagamaan, kegiatan usaha, kegiatan sosial, budaya, maupun kegiatan khusus. Menurut Peraturan Menteri Pekerjaan Umum Nomor 24 tahun 2008 tentang pedoman pemeliharaan bangunan gedung, fungsi dari bangunan gedung meliputi fungsi hunian, keagamaan, usaha, sosial dan budaya serta fungsi khusus adalah ketentuan mengenai pemenuhan persyaratan administratif dan persyaratan teknis bangunan gedung

2.4.2 Gedung negara

Dalam Peraturan Presiden Republik Indonesia Nomor 73 Tahun 2011 Tentang Pembangunan Bangunan Gedung Negara pada bab I pasal 1 "Bangunan gedung negara adalah bangunan gedung untuk keperluan dinas yang menjadi barang milik negara/daerah dan diadakan dengan sumber pembiayaan yang berasal dari dana APBN, dan/atau APBD, atau perolehan lainnya yang sah.

Berdasarkan klasifikasinya bangunan gedung milik negara dalam penyelenggaraannya dibagi menjadi beberapa tiga yaitu bangunan sederhana, bangunan tidak sederhana dan bangunan khusus :

1. **Bangunan Sederhana**

Klasifikasi bangunan sederhana adalah bangunan gedung negara dengan karakter sederhana serta memiliki kompleksitas dan teknologi sederhana. Masa penjaminan kegagalan bangunannya adalah selama 10 (sepuluh) tahun. Yang termasuk klasifikasi Bangunan Sederhana, antara lain:

1. Gedung kantor yang sudah ada disain prototipenya, atau bangunan gedung kantor dengan jumlah lantai s.d. 2 lantai dengan luas sampai dengan 500m².
2. Bangunan rumah dinas tipe c, d, dan e yang tidak bertingkat;
3. Gedung pelayanan kesehatan: puskesmas;
4. Gedung pendidikan tingkat dasar dan/atau lanjutan dengan jumlah lantai s.d. 2 lantai.

2. **Bangunan Tidak Sederhana**

Klasifikasi bangunan tidak sederhana adalah bangunan gedung negara dengan karakter tidak sederhana serta memiliki kompleksitas dan/atau teknologi tidak sederhana. Masa penjaminan kegagalan bangunannya adalah selama paling singkat 10 (sepuluh) tahun.

Yang termasuk klasifikasi Bangunan Tidak Sederhana, antara lain:

1. Gedung kantor yang belum ada disain prototipenya, atau gedung kantor dengan luas di atas dari 500 m², atau gedung kantor bertingkat lebih dari 2 lantai;
2. Bangunan rumah dinas tipe a dan b; atau rumah dinas c, d, dan e yang bertingkat lebih dari 2 lantai, rumah negara yang berbentuk rumah susun;
3. Gedung rumah sakit klas a, b, c, dan d;
4. Gedung pendidikan tinggi universitas/akademi; atau gedung pendidikan dasar/lanjutan bertingkat lebih dari 2 lantai.

3. **Bangunan Khusus**

Klasifikasi bangunan khusus adalah bangunan gedung negara yang memiliki penggunaan dan persyaratan khusus, yang dalam perencanaan dan pelaksanaannya memerlukan penyelesaian/teknologi khusus. Masa penjaminan kegagalan bangunannya paling singkat 10 (sepuluh) tahun. Yang termasuk klasifikasi Bangunan Khusus, antara lain:

1. Istana negara dan rumah jabatan presiden dan wakil presiden;
2. Wisma negara;
3. Gedung instalasi nuklir;
4. Gedung instalasi pertahanan, bangunan polri dengan penggunaan dan persyaratan khusus;
5. Gedung laboratorium;
6. Gedung terminal udara/laut/darat; stasiun kereta api;
7. Stadion olah raga; rumah tahanan; gudang benda berbahaya;
8. Gedung bersifat monumental; dan gedung perwakilan negara R.I. di luar negeri.

2.5 Java

Java merupakan perangkat lunak produksi Sun Microsystem Inc yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya seperti Patrick Naughton, Crhis Warth, Ed Frank dan Mike Sheridan pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak” namun pada tahun 1995 diganti menjadi “Java”. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++[12].

Java mempunyai beberapa kelebihan yaitu Java bersifat *Multiplatform* yang berarti Java dapat dijalankan di beberapa sistem operasi komputer (*platform*), sesuai dengan prinsip “Tulis sekali, jalan di mana saja”. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Java juga merupakan pemrograman Berorientasi Objek (*Object Oriented Programming* (OOP)), Java memiliki perpustakaan kelas yang lengkap dengan pengumpulan sampah yang otomatis sehingga memudahkan pengguna dalam memasukkan kelas[12].

Java juga mempunyai beberapa kekurangan yaitu untuk keterbatasan beberapa *platform* satu dengan *platform* lain. Untuk J2SE misalnya SWT-AWT *bridge* yang sampai sekarang tidak berfungsi pada Mac OS X. Java juga mudah untuk didekompilasi sehingga kode sumber dapat diketahui dengan mudah sehingga memudahkan untuk pembajakan kode sumber. Java menggunakan

memori yang jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal.[12].

Java dibagi menjadi beberapa bagian diantaranya adalah J2SE (Java 2 *Standard Edition*), J2EE (Java 2 *Platform Enterprise Edition*), J2ME (Java 2 *Mobile Edition*), JDK (Java *Development Kit*), JSP (Java *Server Pages*), JVM (Java *Virtual Machine*) dan *Java Script*[12].

2.5.1 Pengertian JDK (*Java Development Kit*)

JDK adalah *software development kit* untuk bahasa pemrograman Java. Alat bantu yang digunakan untuk membuat program dalam bahasa pemrograman Java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan *compiler* dan *debugger* yang diperlukan untuk mengembangkan *applet* dan aplikasi[3].

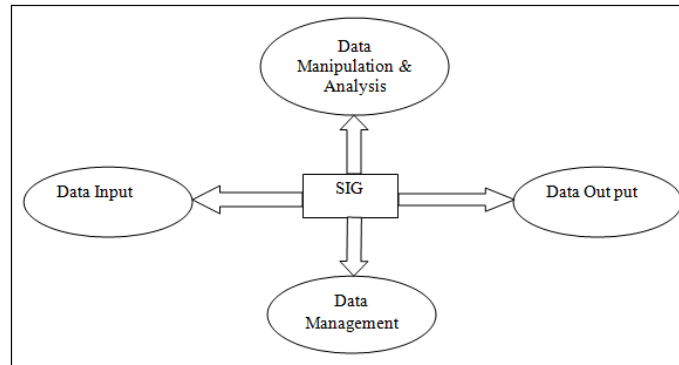
2.6 *Geographic Information System (GIS)*

Geographic Information System (GIS) merupakan sistem yang dirancang untuk bekerja dengan data yang terreferensi secara spasial atau koordinat-koordinat geografi. GIS memiliki kemampuan untuk melakukan pengolahan data dan melakukan operasi-operasi tertentu dengan menampilkan dan menganalisa data. Aplikasi GIS saat ini tumbuh tidak hanya secara jumlah aplikasi namun juga bertambah dari jenis keragaman aplikasinya. Pengembangan aplikasi GIS kedepannya mengarah kepada aplikasi berbasis Android[14]. SIG adalah sistem komputer yang digunakan untuk memasukan (*capturing*), menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data-data yang berhubungan dengan posisi-posisi di permukaan bumi

Selain itu SIG adalah kumpulan yang terorganisir dari perangkat keras komputer, perangkat lunak, data geografi dan personal yang dirancang secara efisien untuk memperoleh, menyimpan, mengupdate, memanipulasi, menganalisa dan memanipulasi semua bentuk informasi yang bereferensi geografi [14].

Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG lain, subsistem ini juga melakukan manipulasi dan pemodelan data untuk

menghasilkan informasi yang diharapkan. Menurut Eddy Prahasta subsistem SIG bisa dilihat di gambar 2.1.



Gambar 2.1 Subsistem dalam SIG

SIG terdiri atas komponen-komponen yang mendukung proses kerja sebagai suatu sistem informasi yang akurat. Komponen-komponen SIG terdiri dari: [14]

1. Perangkat Keras
2. Perangkat Lunak
3. Data dan Informasi Geografi
4. Manajemen.

Sedangkan model data dalam SIG antara lain:

- a. Data Spasial yaitu adalah jenis data yang merepresentasikan aspek-aspek keruangan (menyangkut titik koordinat) dari fenomena atau keadaan yang terdapat di dunia nyata. Data spasial ini sering disebut pula sebagai data posisi, koordinat atau keruangan.
- b. Data atribut atau data non spasial adalah jenis data yang merepresentasikan aspek-aspek deskriptif dari fenomena yang dimodelkan. Aspek deskriptif ini mencakup *item* atau *properties* dari fenomena yang bersangkutan hingga dimensi waktunya.

2.7. Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang dikeluarkan oleh Google Inc pada bulan November 2007, bersifat *open source* sehingga para pengembang dapat membuat program aplikasi sendiri untuk

perangkat *mobile* sesuai dengan kebutuhan[9]. Pemrograman perangkat ini menggunakan bahasa pemrograman *Extensible Markup Language*(XML) dan *Java* yang dikhususkan untuk *platform* ini, sehingga program aplikasi yang dibuat tidak dapat dijalankan pada perangkat *mobile* lain seperti: *Symbian OS* dan *Blackberry OS* yang juga mendukung program aplikasi berbasis *Java*. Terdapat dua distributor sistem operasi ini, yaitu: *GMS*(*Google Mail Services*) yang mendapat dukungan langsung dari Google dan *OHD*(*Open Handset Distribution*) yang tidak mendapat dukungan langsung dari Google dan melakukan pendistribusian sistem operasi ini secara bebas[9].

2.7.1 Keunggulan Android

Android dipuji sebagai platform *mobile* pertama yang lengkap, terbuka, dan bebas.

1. Lengkap (*Complete Platform*)

Para *designer* dapat melakukan pendekatan yang komperhensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan program aplikasi.

2. Terbuka (*Open Source Platform*)

Platform Android disediakan melalui lisensi *open source*. Pengembangan dapat dengan bebas untuk mengembangkan program aplikasi Android sendiri menggunakan Linux Kernel 2.6

3. Bebas (*Free Platform*)

Android adalah *platform* atau program aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Program aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.8 Versi Android

Berbagai varian android yang telah dilahirkan seperti versi pertama yaitu Android v1.0 *Astro (Alpha)*. Kedua adalah Android v1.1 *Bender (Beta)*. Versi ini dirilis pada tanggal 5 November 2007 yang merupakan versi lanjutan dari Android *Astro (Alpha)*. Ketiga adalah Android v1.5 *Cupcake* yang dirilis pada tanggal 30 April 2009. Keempat adalah Android v1.6 *Donut* yang dirilis pada tanggal 15 September 2009. Kelima adalah Android v2.0 – 2.1 *Eclair* yang dirilis pada tanggal 26 Oktober 2009. Ke enam adalah Android v2.2 *Frozen Yoghurt (Froyo)* yang dirilis pada tanggal 20 Mei 2010. Ke tujuh adalah Android v2.3 *Gingerbread*. Dirilis pada tanggal 6 Desember 2010. Ke delapan adalah Android v3.0 – 3.2 *Honeycomb*. Ke sembilan adalah Android v4.0 *Ice Cream Sandwich*. Dirilis pada 16 Desember 2011. Ke sepuluh adalah Android v4.1 – 4.3 *Jelly Bean* Dirilis pada 9 Juli 2012. Ke sebelas adalah Android v4.4 *Kitkat* yang dirilis pada tanggal 31 Oktober 2013. Ke dua belas adalah Android v5.0 – 5.1 *Lollipop*. Dirilis pada tanggal 15 Oktober 2014. Ke tiga belas adalah Android v6.0 *Marshmallow*[18].

2.9 Android SDK

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan program aplikasi pada *platform*. Android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan program aplikasi kunci yang rilis oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan program aplikasi pada platform Android menggunakan bahasa pemrograman java. Sebagai *platform* program aplikasi-netral, Android memberi anda kesempatan untuk membuat program aplikasi yang kita butuhkan yang bukan merupakan program aplikasi bawaan *handphone* atau *smartphone*[9].

Android SDK terdiri dari *debugger*, *libraries*, *handset*, *emulator*, dokumentasi, contoh kode, dan *tutorial*. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk *desktop* modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK,

Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *pluginAndroid Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks *editor* untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* program aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh)[20].

2.10 Android Studio IDE

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform* Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan *Ant*, Android Studio menggunakan Gradle sebagai *build environment*. Android studio menggunakan bahasa pemrograman Java. [4]. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Android Studio menawarkan beberapa fitur dan kelebihan seperti berikut:

1. Dikembangkan IntelliJ IDEA Community Edition, Java IDE yang populer oleh JetBrains.
2. Fleksibel *build system* berbasis Gradle.
3. Membuat beberapa APK dalam *project* yang sama.
4. Editor kode cerdas yang mampu menyelesaikan kode, *refactoring*, dan analisis kode.
5. Diperluas dukungan template untuk layanan Google dan berbasis jenis perangkat seperti ponsel Android, tablet, Android Wear, Android TV, Android Auto dan Google Glass.
6. Terintegrasi dengan GitHub. Kita bisa memulai proyek menggunakan kode template bahkan mengimpor contoh kode Google dari gitHub.
7. Layout editor kaya dengan dukungan ntuk mengedit tema.
8. Built-in mendukung untuk *Google Coud Platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine*.

2.11 XML

XML merupakan bahasa markup berupa sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. XML terletak pada inti *Web Service* yang digunakan untuk mendeskripsikan data. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan partner luar. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dengan mudah berkomunikasi antar satu dengan yang lain[20].

XML dikembangkan mulai tahun 1996 dan mendapatkan pengakuan dari W3C pada bulan Februari 1998. Teknologi yang digunakan pada XML merupakan turunan dari SGML yang pada awal 80 dan telah banyak digunakan pada dokumentasi teknis proyek-proyek berskala besar. Ketika HTML dikembangkan pada tahun 1990, para pengagas XML mengadopsi bagian paling penting pada SGML dan dengan berpedoman pada pengembangan HTML menghasilkan *markup language* yang tidak kalah hebatnya dengan SGML[20].

2.12 Diagram UML

Perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk mengspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung[8].

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek[8].

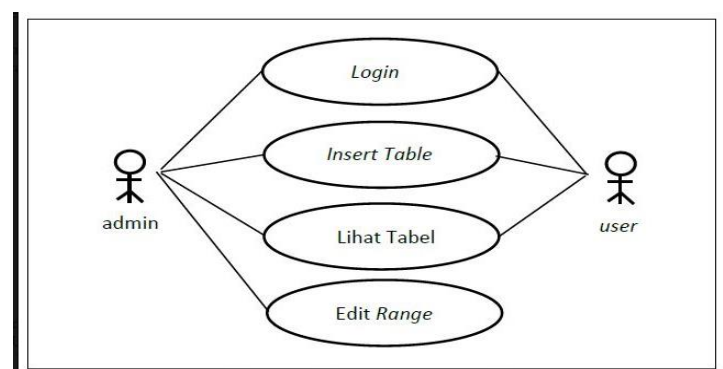
Seperti yang kita ketahui bahwa banyak hal di dunia sistem informasi yang tidak dapat dibakukan, semua tergantung kebutuhan lingkungan, lingkungan dan

konteksnya. Begitu juga dengan perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi, belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin di visualisasikan. Secara analogi jika dengan bahasa yang kita gunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalah hal yang salah. Sistem informasi bukanlah ilmu pasti, maka jika ada banyak perbedaan dan interpretasi didalam bidang sistem informasi merupakan hal yang sangat wajar[8].

2.12.1 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. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya[8].

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Contoh dari Use Case Diagram:



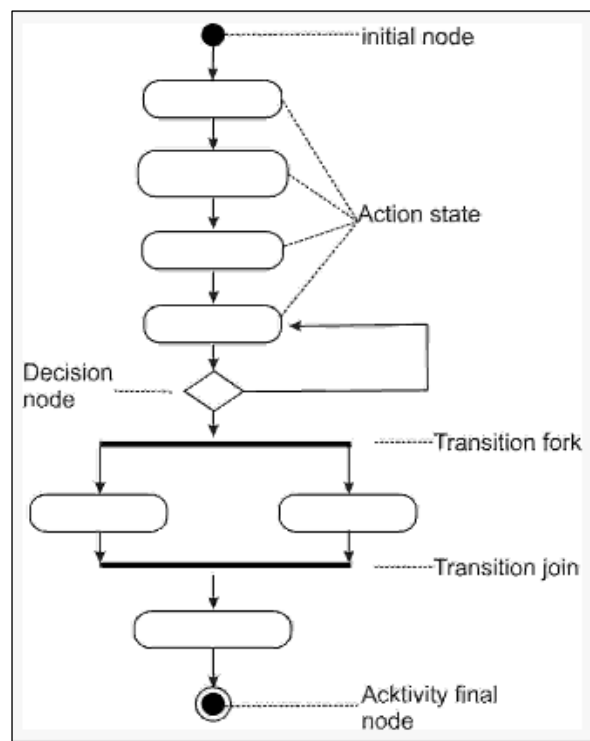
Gambar 2.2 contoh Use Case Diagram

2.12.2 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[8].

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Contoh dari Activity Diagram:



Gambar 2.3 contoh *Activity Diagram*

2.13 Google Map API (*Application Programming Interface*)

API merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Maps* yang terdiri dari HTML, Javascript, dan AJAX serta XML, memungkinkan untuk menampilkan peta *Google Maps* di *website* lain[2].

Google juga menyediakan layanan *Google Maps API* yang memungkinkan para pengembang untuk mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan 18

Google Maps API, *Google Maps* dapat ditampilkan pada *website* eksternal. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan kode unik yang digenerasikan oleh Google untuk suatu *website* tertentu, agar server *Google Maps* dapat mengenali. Google juga menyediakan layanan untuk arah ke tempat yang dituju[2].

2.14 API (*Application Programming Interface*)

Antarmuka pemrograman aplikasi (*Application Programming Interface/API*) adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk system operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi.

API juga dapat diartikan sebagai sekumpulan perintah, fungsi, class dan protocol yang memungkinkan suatu software dapat menggunakan fasilitas dan saling berhubungan dengan software lainnya. Tujuan dari API adalah untuk menghilangkan kerumitan dari sistem dengan cara membuat sebuah blok besar

yang terdiri dari software di seluruh dunia dan menggunakan kembali perintah, fungsi, class atau protokol yang API miliki, dengan cara ini programmer tidak perlu lagi membuang waktu untuk membuat dan menulis infrastruktur sehingga akan menghemat waktu kerja dan lebih efisien[6].

2.15 Google API

Google api adalah sekumpulan aplikasi antarmuka pemrograman (APIs) dikembangkan oleh Google yang memungkinkan komunikasi dengan layanan Google dan integrasi mereka ke layanan lain. Contoh pencarian, Gmail, menerjemahkan atau Google Maps. Aplikasi pihak ketiga dapat menggunakan api untuk mengambil keuntungan dari atau memperluas fungsi layanan yang ada[15].

Api menyediakan fungsi seperti analytics, mesin belajar sebagai layanan (prediksi API) atau akses ke data pengguna (ketika diberikan izin untuk membaca data). Contoh penting lain adalah Google peta tertanam di website, yang dapat dicapai dengan menggunakan API maps statis, tempat API atau Google Earth API.

2.16 Global Positioning System (GPS)

GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana lokasinya berada (secara global) di permukaan bumi yang berbasis satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun pengguna tersebut berada, maka *GPS* bisa membantu menunjukkan arah. Layanan *GPS* ini tersedia gratis.

Awalnya *GPS* hanya digunakan hanya untuk kepentingan militer, tapi pada tahun 1980-an dapat digunakan untuk kepentingan sipil. *GPS* dapat digunakan dimanapun juga dalam 24 jam. Posisi unit *GPS* akan ditentukan berdasarkan titik-titik koordinat *latitude* dan *longitude*[7].

2.17 Location Based Services (LBS)

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk

menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning services* (GPS) dan *cell-based location* dari Google[5].

Selain itu, LBS tersebut terdiri dari beberapa komponen diantaranya *mobile devices*, *communicatin network*, *position component* dan *services and content provider*. *Mobile device* merupakan komponen yang sangat penting, piranti mobile tersebut diantaranya adalah smartphone dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning service* (GPS) dan *cell-based location* dari Google.

Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan *service and content provider*. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya.

Sederhananya, dengan layanan LBS kita dapat mengetahui posisi dimana kita berada, posisi teman, dan posisi rumah sakit atau pom bensin yang jaraknya dekat dengan kita. Dalam mengukur posisi, digunakan lintang dan bujur untuk menentukan lokasi geografis. Tetapi, Android menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, kita dapat mengkonversi nilai lintang bujur menjadi alamat dunia nyata atau sebaliknya[5].

2.18 Pengertian Basis Data (*Database*)

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak/program aplikasi untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data 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 mengorganisasi data, menghindari duplikasi data, hubungan antar data yang tidak jelas dan juga update yang rumit. Basis data memiliki beberapa jenis *software* untuk membuatnya yaitu Microsoft Office Access, Ms SQL Server, Oracle, MySql, Postgre SQL dan Firebird[10].

2.19 Codeigniter

CodeIgniter adalah framework pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka untuk kerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu membuat program dari awal (from scrach), karena CI menyediakan sekumpulan ribrari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan[17].

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view, dan controller.

1. Model

Bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. Seperti misalnya pengambilan data dari database, menginput dan pengelolaan database lainnya.

2. View

Bagian yang menangani halaman user interface atau halaman yang muncul pada user. Tampilan dari user interface di kumpulkan pada view untuk memisahkan dengan controller dan model sehingga memudahkan designer dalam melakukan pembangunan halaman website.

3. Controller

Merupakan kumpulan intruksi yang menghubungkan model dengan view. Jadi user tidak akan berhubungan dengan model secara langsung.

2.20 JavaScript

JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT. JavaScript pertama kali dikembangkan oleh Brendan Eich dari Netscape dibawah nama Mocha, yang nantinya namanya diganti menjadi LiveScript, dan akhirnya menjadi JavaScript. Navigator sebelumnya telah mendukung Java untuk lebih bisa dimanfaatkan para programmer yang non-Java. Maka dikembangkanlah bahasa pemrograman bernama LiveScript untuk mengakomodasi hal tersebut. Bahasa pemrograman inilah yang akhirnya berkembang dan diberi nama JavaScript, walaupun tidak ada hubungan bahasa antara Java dengan JavaScript. JavaScript bisa digunakan untuk banyak tujuan, misalnya untuk membuat efek rollover baik di gambar maupun teks, dan yang penting untuk membuat AJAX. JavaScript adalah bahasa yang digunakan untuk AJAX[13].

2.21 PHP

PHP merupakan Bahasa pemrograman script yang disisipkan pada HTML. PHP merupakan Bahasa pemrograman yang bersifat open source atau gratis. PHP hanya dapat bekerja pada sisi *server side* oleh karena itu PHP membutuhkan aplikasi yang dapat membuat komputer menjadi local server agar php dapat bekerja[21].

Pada tahun 1997 php ditulis ulang oleh perusahaan zend menjadi lebih baik, bersih dan cepat. Kemudian pada juni 1998, merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0. pada pertengahan 1999 zend merilis PHP 4.0. versi ini adalah versi yang paling banyak dipakai pada awal abad ke-21[21].

Pada juni 2004, zend merilis PHP 5.0. dalam versi ini PHP mengalami perubahan besar. Model terbaru ini juga memasukkan pemrograman berorientasi objek ke dalam PHP. Hingga penghujung tahun 2013 PHP sudah merilis versi 5.6.2[21].

2.22 MySQL

HTML atau *Hypertext Markup Language* merupakan Bahasa markah yang digunakan untuk mendesain halaman web yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam browser yang ditulis dalam format ASCII agar dapat menghasilkan wujud terintegrasi. HTML adalah standart yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar yang dikendalikan oleh *World Wide Web*[1].

2.23 Xampp

Xampp adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama Xampp merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya[11].

2.24 Pengertian JSON (*Java Script Object Notation*)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat 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 dan lain-lain[12].

2.25 Teknologi REST API

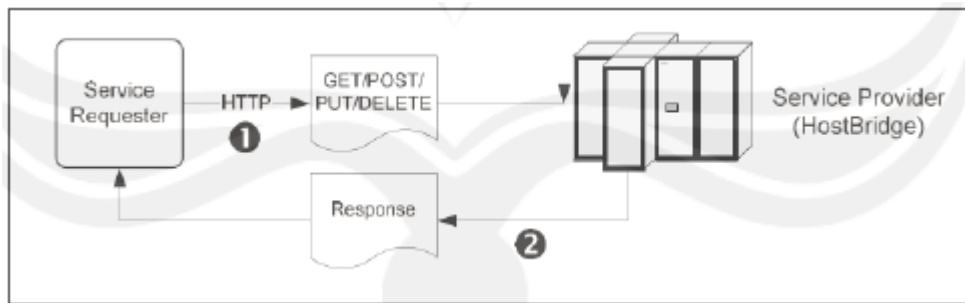
Ada beberapa pendekatan teknologi *web services* yang dapat digunakan, yaitu metode SOAP dan REST. Dalam penelitian ini akan digunakan pendekatan REST.

REST merupakan singkatan dari *REpresentative State Transfer*. Pertama kali dikemukakan dalam disertasi seorang program doktor bernama Roy Thomas Fielding pada tahun 2000 (Rozali, 2011). REST adalah sebuah metode dalam menyampaikan *resource* melalui media web. Sedangkan *resource* sendiri didefinisikan sebagai segala sesuatu yang dapat disimpan didalam sebuah komputer dan ditampilkan sebagai urutan bit, misalnya sebuah dokumen, tabel dalam sistem basis data, atau hasil dari sebuah perhitungan (Sandoval, 2008).

REST (*REpresentational State Transfer*) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti HTTP (*Hypertext Transfer Protocol*) dan XML (Riyadi, 2013).

2.25.1 Arsitektur REST

Pada Gambar 2.4 Dibawah ini merupakan model dasar dari RESTful *Web services*.



Gambar 2.4 Model dasar RESTful *Web services* (HostBridge,2009).

Berikut merupakan langkah-langkah yang dilakukan dalam model dasar RESTfull *Web services* (HostBridge, 2009):

1. Query Request Provider melalui HTTP dengan menggunakan URI (*Uniform Resource Identifier*). *Request* menggunakan *methods* (metode) HTTP untuk menentukan apakah request tersebut dimaksudkan untuk *Create* (menciptakan), *Read* (membaca), *Update* (memperbarui), atau *Delete* (menghapus) data.
2. *HostBridge* mengembalikan sebuah dokumen dalam bentuk *XML* untuk *Requester* (pemohon) dengan *CICS data enclosed*.

Diagram di atas menunjukkan dua *web services* : sebuah Provider (HostBridge) yang menyediakan layanan *web services* dan sebuah Request yang digunakan pada *web services*. Dalam model RESTful layanan dideskripsikan secara sendiri (*self-describing*), sehingga tidak diperlukan WSDL dan Server UDDI yang bertindak sebagai Broker (HostBridge, 2009).

Sehingga pada development web modern aksi CRUD tersebut dapat dimappingkan dengan HTTP method sebagai berikut : *CREATE* sebagai *POST*, *RETRIVE* sebagai *GET*, *UPDATE* sebagai *PUT*, dan *DELETE* sebagai *DELETE*. Berikut merupakan penggunaan *methods*(metode) HTTP dalam REST *Web services* (Riyadi, 2013):

Tabel 2.2 Metode HTTP dan Penggunaannya dalam REST (Riyadi, 2013).

Metode	Deskripsi
GET	Mendapatkan (<i>read</i>) sebuah sumber daya (<i>resource</i>) yang diidentifikasi dengan URI (<i>Uniform Resource Identifier</i>)
POST	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk membuat (<i>create</i>) sumber daya baru.
PUT	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk memasukkan (<i>insert</i>) atau memperbarui (<i>update</i>) sumber daya yang tersimpan.
DELETE	Menghapus (<i>delete</i>) sumber daya (<i>resource</i>) yang diidentifikasi dengan URI.
HEAD	Mendapatkan metadata (<i>response header</i>) dari sumber daya(<i>resource</i>) yang diidentifikasi dengan URI.

Arsitektur REST dibangun dengan sifat sebagai berikut (Gröhbiel, 2011):

1. *Addressability*

Dalam prinsip ini seluruh suberdaya atau resource harus tersedia melalui sebuah alamat unik, pengalamatan ini dilakukan dengan menggunakan URI(*Unique Resource Identifiers*)

2. *Uniform Interface*

Semua interaksi sebaiknya dibangun dengan interface yang seragam. RESTful service menampilkan semua *resource* dan interaksinya dengan *interface* yang seragam, dalam metode REST antarmuka yang digunakan adalah dengan menggunakan HTTP. HTTP menawarkan semua operasi yang diperlukan, dikenal, dan tersebar luas. Semua interaksi antara klien dan sumber daya (*resource*) didasarkan pada metode dasar HTTP. Metode-metode interaksi yang digunakan dengan dasar HTTP ditunjukkan pada tabel 2.1

3. *Representation-oriented*

Representasi menjelaskan dalam bentuk apa data sedang dipertukarkan antara *client* dan *server*. Pada umumnya data dipertukarkan dalam bentuk XML, JSON, dan HTML.

4. *Statelessness*

Setiap interaksi antara *client* dan *server* harus memiliki state sendiri (atau dengan kata lain tidak dipengaruhi *session client*). Jadi *server* hanya akan memantau *resource* state bukan *client session*.

5. *Hypermedia As The Engine Of Application State* (HATEOAS)

Hypermedia sebagai state dari sebuah aplikasi (HATEOAS), menyatakan REST dapat menggunakan *link* untuk menghubungkan sumber daya atau *resource* ke sumber daya lain yang berkaitan. Hal ini mirip dengan web, dimana kami menggunakan *hyperlink* untuk menghubungkan antara situs web.

2.26 **BlackBox Testing**

Untuk mengembangkan suatu perangkat lunak yang baik, diperlukan pengujian yang tepat. Pentingnya pengujian secara komprehensif dan kualitatif dalam menguji setiap modul perangkat lunak sehingga biaya pemeliharaan dapat berkurang. Namun karena kode sebuah perangkat lunak tidak selalu tersedia, pengujian dalam kondisi tersebut adalah dengan menggunakan metode Blackbox Testing. Pengujian Blackbox Testing adalah pengujian yang mengabaikan mekanisme internal suatu system atau komponen lain dan hanya berfokus pada *output* yang dihasilkan sebagai balasan terhadap masukan yang dipilih dan kondisi yang dieksekusi. Ada banyak jenis pengujian *blackbox* dapat dilakukan, beberapa diantaranya *Boundary value*, *Robustness*, *Worst Case*, *Equivalence*, *cause effect* dan *State Transate Transition Table*. Dalam kondisi di mana kode perangkat lunak tidak tersedia, pengujian yang dihasilkan melalui Teknik pengujian Blackbox harus sedemikian sehingga kualitas pengujian tetap sama. Tujuan utama pengujian adalah

untuk mencari sebuah kesalahan dalam perangkat lunak dan untuk menghindari potensi kegagalan.