

BAB II

LANDASAN TEORI

2.1 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) dapat merepresentasikan *real word* (dunia nyata) di atas monitor komputer sebagaimana lembaran peta dapat merepresentasikan dunia nyata di atas kertas. Data yang mempresentasikan dunia nyata dapat disimpan dan diproses sedemikian rupa sehingga dapat disajikan dalam bentuk-bentuk yang lebih sederhana dan sesuai kebutuhan. SIG memiliki kekuatan lebih dan fleksibilitas daripada lembaran peta kertas (mahendra, 2017).

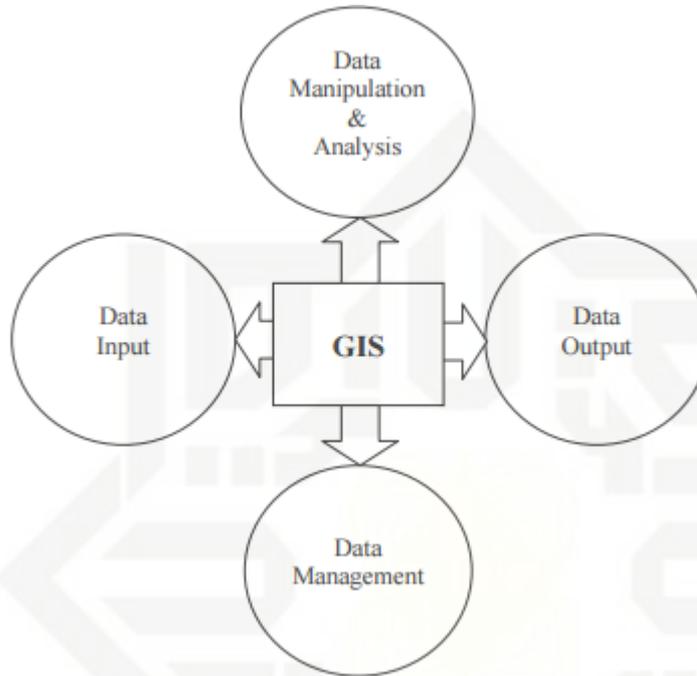
Sistem Informasi Geografis telah terbukti efektif dan ampuh alat dalam industri distribusi air. GIS memungkinkan penciptaan dan pengelolaan data yang direferensikan spasial yang berguna untuk bidang atau situasi yang memanfaatkannya informasi spasial. Sistem Informasi Geografi merupakan gabungan dari 3 unsur pokok yaitu sistem, informasi, dan geografi. Sistem Informasi Geografi merupakan suatu sistem yang menekankan pada unsur unsur “Informasi Geografi” (Prahasta, 2009).

2.1.1 Sub sistem SIG

Menurut Prahasta 2009, Sistem Informasi Geografis dapat diuraikan menjadi empat subsistem sebagai berikut:

1. Data *input*: bertugas untuk mengumpulkan data dan mempersiapkan data spasial dan atribut dari berbagai sumber dan bertanggung jawab dalam mengkonversi atau mentransfortasikan format-format data-data aslinya kedalam format yang dapat digunakan oleh GIS.
2. Data *output*: menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk softcopy maupun bentuk hardcopy seperti: tabel, grafik dan peta.
3. Data *management*: mengorganisasikan baik data spasial maupun data atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, diupdate dan diedit.

4. Data *manipulation & analysis*: menentukan informasi yang dapat dihasilkan oleh GIS dan melakukan manipulasi serta pemodelan data untuk menghasilkan informasi yang diharapkan, dapat di lihat pada Gambar 2.1.



Gambar 2.1 Subsistem SIG (Prahasta, 2009)

2.1.2 Komponen SIG

Menurut Prahasta (2009), SIG sebagai *system* terdiri dari beberapa komponen dengan berbagai karakteristik yaitu:

1. Perangkat Keras. Pada saat ini SIG sudah tersedia bagi berbagai platform perangkat keras, mulai dari kelas PC desktop, *workstations*, hingga *multi-user host* yang bahkan dapat digunakan banyak pengguna secara bersamaan dalam jaringan komputer yang tersebar luas, berkemampuan tinggi, memiliki ruang penyimpanan (*harddisk*) yang besar dan mempunyai kapasitas memori (RAM) yang besar.
2. Perangkat lunak. Dari sudut pandang yang lain, SIG biasa juga merupakan sistem perangkat lunak yang tersusun secara modular dimana sistem basis datanya memegang peranan kunci.

2.3 *Augmented Reality (AR)*

Augmented Reality (AR) adalah penggabungan objek-objek digital baik itu berupa dua dimensi maupun tiga dimensi dengan dunia nyata yang lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (*real time*) (Rozaldi, 2015). Objek-Objek digital yang menampilkan informasi tidak dapat diterima pengguna oleh inderanya sendiri. Informasi yang ditampilkan oleh objek-objek digital hanya sebagai alat yang membantu persepsi dan interaksi penggunaannya dengan dunia nyata. Sehingga informasi tersebut membantu penggunaannya melaksanakan kegiatan dalam dunia nyata (Akbar, 2012)

Menurut definisi Ronald Azuma pada tahun 1997, ada tiga prinsip dari *Augmented Reality*. Yang pertama yaitu *Augmented Reality* merupakan penggabungan dunia nyata dan virtual, yang kedua berjalan secara interaktif dalam waktu nyata (*realtime*), dan yang ketiga terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata.

Bidang-bidang yang pernah menerapkan teknologi *Augmented Reality* adalah:

1. Kedokteran (*Medical*): Teknologi pencitraan sangat dibutuhkan di dunia kedokteran, seperti misalnya, untuk simulasi operasi, simulasi pembuatan vaksin virus, dll. Untuk itu, bidang kedokteran menerapkan *Augmented Reality* pada visualisasi penelitian mereka.
2. Hiburan (*Entertainment*): Dunia hiburan membutuhkan sebagai penunjang efek-efek yang akan dihasilkan oleh hiburan tersebut. Sebagai contoh, ketika seseorang wartawan cuaca memperkirakan ramalan cuaca, dia berdiri di depan layar hijau atau biru, kemudian dengan teknologi *Augmented Reality*, layar hijau atau biru tersebut berubah menjadi gambar animasi tentang cuaca tersebut, sehingga seolah-olah wartawan tersebut, masuk ke dalam animasi tersebut.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. *Latihan Militer (Military Training)*: Militer telah menerapkan *Augmented Reality* pada latihan tempur mereka. Sebagai contoh, militer menggunakan *Augmented Reality* untuk membuat sebuah permainan perang, dimana prajurit akan masuk kedalam dunia *game* tersebut, dan seolah-olah seperti melakukan perang sesungguhnya.
4. *Engineering Design*: Seorang *engineering design* membutuhkan *Augmented Reality* untuk menampilkan hasil design mereka secara nyata terhadap klien. Dengan *Augmented Reality* klien akan tahu, tentang spesifikasi yang lebih detail tentang desain mereka.
5. *Robotics dan Telerobotics*: Dalam bidang robotika, seorang operator robot, menggunakan pengendari pencitraan visual dalam mengendalikan robotitu. Jadi, penerapan *Augmented Reality* dibutuhkan di dunia robot.
6. *Consumer Design*: Virtual reality telah digunakan dalam mempromsikan produk. Sebagai contoh, seorang pengembang menggunakan brosur *virtual* untuk memberikan informasi yang lengkap secara 3D, sehingga pelanggan dapat mengetahui secara jelas, produk yang ditawarkan.

Ada beberapa metode yang digunakan pada *Augmented Reality* yaitu *marker based tracking* dan *markerless*.

1. *Marker Based Tracking*: *Marker Based Tracking* adalah AR yang menggunakan *marker* atau penanda objek dua dimensi yang memiliki suatu pola yang akan dibaca komputer melalui media webcam atau kamera yang terhubung dengan komputer, biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih.
2. *Markerless*: dengan metode *markerless* pengguna tidak perlu lagi mencetak sebuah *marker* untuk menampilkan elemen-elemen digital. Dalam hal ini. *Marker* yang dikenali berbentuk posisi perangkat, arah maupun lokasi.

2.3.1 *Markerless Augmented Reality*

Salah satu metode *augmented reality* yang saat ini sedang berkembang adalah metode *Markerless augmented reality*, dengan metode ini pengguna tidak perlu lagi mencetak sebuah marker untuk menampilkan elemen-elemen digital. Dalam hal ini, *marker* yang dikenali berbentuk posisi perangkat, arah, maupun lokasi. Total immersion dan Qualcomm adalah salah satu perusahaan yang mengembangkan *Augmented Reality* dengan berbagai macam teknik *Markerless Tracking* diantaranya adalah *Face Tracking*, *3D Object Tracking*, *Motion Tracking* dan *GPS Based Tracking* (Fernando, 2013)

2.3.2 *GPS Based Tracking*

Teknik *GPS Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi *smartphone* (iPhone dan Android). Dengan memanfaatkan fitur GPS dan kompas yang ada didalam *smartphone*, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *realtime*, bahkan ada beberapa aplikasi menampilkannya dalam bentuk 3D. Salah satu pelopor *GPS Based Tracking* adalah aplikasi yang bernama Layar (Fernando, 2013).

2.4 **Android**

Android adalah sistem operasi *mobile* yang dikembangkan oleh Google, berdasarkan kernel Linux dan dirancang terutama untuk perangkat *mobile touchscreen* seperti *smartphone* dan *tablet*. Awalnya dikembangkan oleh *Android Inc.*, yang dibeli Google pada tahun 2005, *Android* diresmikan pada tahun 2007, bersamaan dengan berdirinya *Open Handset Alliance* sebuah konsorsium perangkat keras, perangkat lunak, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar terbuka untuk perangkat *mobile*.

Antarmuka pengguna *Android* terutama didasarkan pada manipulasi langsung, menggunakan gerakan sentuhan yang secara longgar sesuai dengan tindakan dunia nyata, seperti menggesek, mengetuk dan mencubit, untuk memanipulasi objek di layar, bersama dengan *keyboard* virtual untuk *input* teks. Selain perangkat *touchscreen*, Google telah mengembangkan *Android TV* untuk televisi, *Android Auto* untuk mobil, dan *Android Wear* untuk jam tangan, masing-

masing dengan antarmuka pengguna khusus. Varian *Android* juga digunakan pada *game* konsol, kamera digital, PC dan elektronik lainnya. Disamping itu produsen perangkat keras juga dapat menambahkan extensionnya sendiri kedalam *android* sebagai kebutuhan produk mereka. Model pengembangannya yang sederhana menarik bagi vendor-vendor perangkat keras. Keuntungan utama dari *Android* adalah adanya pendekatan aplikasi terpadu. Pengembang hanya berkonsentrasi pada aplikasi saja, aplikasi tersebut dapat berjalan pada beberapa perangkat yang berbeda selama masih ditenagai *Android*.

2.5 Web Service

Web Service adalah aplikasi sekumpulan data (database), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. Web service dapat diartikan juga sebuah metode penukaran data, tanpa memperhatikan dimana sebuah database ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. Web service mampu menunjang interoperabilitas. Sehingga web service mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada. Menurut W3C *Web services Architecture Working Group* pengertian Web service adalah sebuah sistem software yang didesain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. Interface web service dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin khusus WSDL. (Pandwinata, Vina. 2013).

2.6 Representational State Transfer (REST)

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui link-link HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan state satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Balasan yang

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi client.

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk web service yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resourcenya* itu, sedangkan aplikasi lowongan pekerjaan membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data lowongan. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API(*Application Program Interface*), YouTube API, Amazon API.(Abeysinghe, 2008)

2.7 JSON (*Java Script Object Natation*)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer.

Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3–Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai

Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2. Daftar nilai terurutkan (*an ordered list of values*)

Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini (Hakim dan Rizki, 2012).

2.8 JQuery

Pengertian JQuery, JQuery merupakan sebuah Javascript Library atau disebut juga perpustakaan dan kumpulan code/listing Javascript yang siap pakai. Dalam artian sederhana, JQuery dapat digunakan untuk meringkas sebuah listing Javascript yang panjang dalam sebuah proyek pembuatan website, sehingga akan memberikan kemudahan dalam menghadapi bagian yang mengandung Javascript. JQuery merupakan program yang berjalan pada sisi server dan akan ditampilkan pada web browser. JQuery dapat berjalan di dalam HTML, atau bahasa pemrograman berbasis web lainnya seperti PHP atau JSP.

Keunggulan JQuery, mudah dioperasikan, karena kita hanya tinggal mengetikkan beberapa baris kode saja, sehingga tidak perlu menuliskan program panjang. JQuery menyediakan fasilitas Plugins yang beragam. Ini sangat berguna dalam pembuatan aksesoris maupun teknologi suatu website yang dibangun menggunakan JQuery. JQuery dapat menyesuaikan CSS style dalam semua web browser, sehingga pengunjung dapat menikmati tampilan web yang bagus dari semua browser. Juga menyediakan fasilitas untuk tampilan animasi seperti pada pembuatan animasi menggunakan flash. Selain itu didukung oleh komunitas dan pengembang JQuery yang tersebar diseluruh Dunia. JQuery merupakan open source, sehingga bebas dan gratis digunakan oleh siapa saja (Prasetyo, Sanjaya, 2012).

2.9 *Bootstrap*

Bootstrap merupakan sebuah framework css yang memudahkan pengembang untuk membangun website yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharannya. *Bootstrap* adalah css tetapi dibentuk dengan LESS, sebuah pre-processor yang memberi fleksibilitas dari css biasa. *Bootstrap* memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas interface yang setiap pengembang hadapi. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan design butuhkan (Otto, 2011).

Keunggulan dalam menggunakan *Bootstrap* adalah semua bagian untuk antarmuka pengguna menggunakan style css, *Bootstrap* dapat menggunakan LESS preprocessor sebuah teknologi yang mengurangi dan mengefisienkan penulisan kode CSS. *Bootstrap* dapat diintegrasikan dengan JavaScript untuk menjadikan lebih menarik dengan efekefek yang dapat diberikan dengan JavaScript (Tectale, 2012).

Kelemahan dalam menggunakan *Bootstrap* adalah dengan adanya penggunaan bootstrap menjadi tidak adanya keunikan didalam website karena akan samanya tampilan yang diberikan. Terdapat juga laporan bahwa sistem grid pada bootstrap tidak responsif (Tectale, 2012).

2.10 *Unified Modelling Language (UML)*

UML adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini di sebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah di mengerti serta dilengkapi dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka yang lain (Nugroho, Adi, 2005).

Tujuan utama perancangan UML adalah:

1. Menyediakan bahasa pemodelan Visual yang Ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang
Syaikh Muhammad Ali bin Abdurrahman bin Yusuf
Sultan Syarif Kasim Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

UML menyediakan beberapa diagram yang disediakan dalam UML antara lain:

- a. Diagram *use case* (*Use case diagram*)
- b. Diagram aktivitas (*activity diagram*)
- c. Diagram Sekuensial (*sequence diagram*)
- d. Diagram kolaborasi (*collaboration diagram*)
- e. Diagram kelas (*class diagram*)
- f. Diagram *statechart* (*statechart diagram*)
- g. Diagram komponen (*component diagram*)
- h. Diagram *deployment* (*deployment diagram*)

2.10.1 Diagram-Diagram UML

UML memiliki beberapa diagram yang digunakan untuk menggambarkan suatu sistem. Tujuan pembuatan diagram ini adalah agar sistem mudah dimengerti oleh semua pihak, baik yang teknis maupun nonteknis (Fowler, 2005). Beberapa contoh dari diagram tersebut, antara lain:

1. *Class Diagram*

Diagram ini terdiri dari sekumpulan class dan interface lengkap dengan kolaborasi dan hubungan antara mereka.

2. *Use Case Diagram*

Menggambarkan sekumpulan use case dan actor dan hubungan antara mereka. Use case diagram mempunyai peranan penting dalam pengorganisasian dan pemodelan behavior dari sistem.

3. Activity Diagram

Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang.

4. Sequence Diagram

Merupakan diagram interaksi yang menekankan pada urutan waktu dari pertukaran message.

2.10.2 Diagram Kelas (Class Diagram)

Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Biasanya, dibuat beberapa diagram kelas untuk satu sistem.

Tabel 2.1 Simbol Diagram Kelas

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

(Fowler, 2005)

2.10.3 Diagram Use Case (Use Case Diagram)

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun.

Tabel 2.2 Simbol Diagram *Use Case*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

(Fowler, 2005)

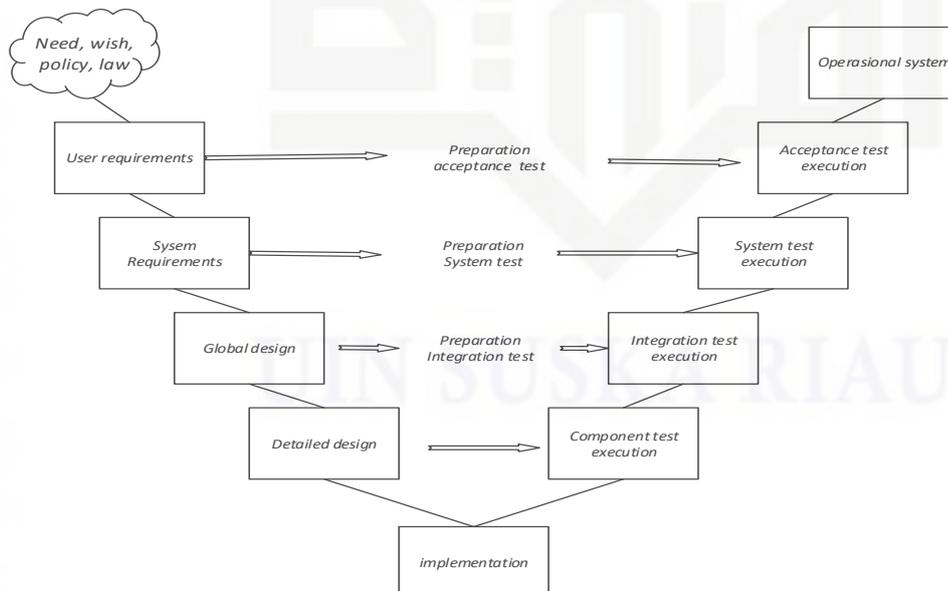
Tabel 2.2 Simbol Diagram *Use Case* (Lanjutan)

No	Gambar	Nama	Keterangan
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

(Fowler, 2005)

2.11 Metode Pengembangan Sistem Menggunakan V-Model

Metode pengembangan sistem *V-model* merupakan perluasan dari metode *waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model *waterfall*. Jika dalam model *waterfall* proses dijalankan secara linear, maka dalam V-model proses dilakukan bercabang. Dalam V-model ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya.



Gambar 2.2. Tahapan Metode V-Model (Graham, 2006)

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Proses pengembangan ini seimbang dan bergantung pada verifikasi dari langkah sebelumnya dilanjutkan. Hasil dari setiap tahap perlu diperiksa dan disetujui sebelum dilanjutkan (Balaji, 2012).

Berikut penjelasan masing-masing tahap beserta tahap pengujiannya.

1. *User Requirement & Acceptance Testing*

Tahap *User Requirement* sama seperti yang terdapat dalam model *waterfall*. Keluaran dari tahap ini adalah dokumentasi kebutuhan pengguna. *Acceptance Testing* merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak.

2. *System Requirements & System Testing*

Dalam tahap ini analisis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data, dan yang lain.

3. *Global Design & Integration Testing*

Sering juga disebut *Hight Level Design*. Dasar dari pemilihan arsitektur yang akan digunakan berdasarkan kepada beberapa hal seperti: pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar interface, detail teknologi yang dipakai. Penyajian pada tahap desainpun berbagai macam, seperti menggunakan UML dan diagram.

4. *Detail Design & Unit Testing*

Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan programmer melakukan *coding*. Tahap ini menghasilkan spesifikasi program seperti: fungsi dan logika tiap modul, pesan kesalahan, proses input-output untuk tiap modul, dan lain-lain.

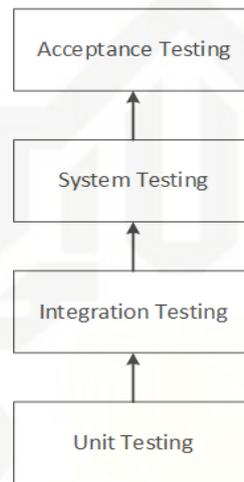
Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

5. Implementation

Design yang telah dirancang kemudian ditranslasi kedalam kode melalui *event-event* untuk mengimplementasikan logika program. Proses implementasi ini dilakukan pada perangkat lunak pengembangan.

Menurut Rifai (2015), terdapat empat fase pengujian yang dapat dilihat pada gambar 2.3



Gambar 2.3 Tahapan Fase Pengujian
(Sumber: Rifai, 2015)

1. Unit Testing

Unit Testing adalah proses pengujian perangkat lunak dimana masing-masing unit/komponen diuji. Tujuannya adalah untuk memvalidasi bahwa setiap unit perangkat lunak sudah melakukan seperti apa yang telah dirancang.

Menurut Pressman (2010), *unit testing* berfokus pada upaya verifikasi terhadap unit terkecil dari perancangan perangkat lunak. Pengujian unit berfokus pada logika pemrosesan internal dan struktur data didalam komponen.

Unit testing merupakan proses dimana pengujian dilakukan pada bagian *basic* dari kode program. Contohnya adalah memeriksa kode program pada *event*, *procedure*, dan *function*. *Unit testing* meyakinkan bahwa masing-masing unit tersebut berjalan sebagaimana mestinya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. *Integration Testing*

Integration Testing adalah proses pengujian perangkat lunak dimana unit individu digabungkan dan diuji sebagai sebuah kelompok. Sehingga pengujian ini mampu menampilkan kesalahan dalam interaksi antar unit. Menurut Pressman (2010), pengujian integrasi adalah teknik untuk membangun arsitektur perangkat lunak, sementara pada saat yang sama melakukan pengujian untuk menemukan kesalahan terkait antarmuka. Tujuannya adalah untuk mengambil komponen yang diuji dan membangun struktur program yang telah ditentukan oleh perancangan.

Setelah melakukan Unit dan *Component Testing*, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Sebagai contoh, kita memiliki sebuah proses yang dikerjakan oleh dua *function*, di mana satu *function* menggunakan hasil output dari *function* yang lainnya. Kedua *function* ini telah berjalan dengan baik secara individu pada *Unit Testing*.

Pada tahap *Integration Testing*, kita memeriksa hasil dari interaksi kedua *function* tersebut, apakah bekerja sesuai dengan hasil yang diharapkan. Kita juga harus memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antar unit tersebut menghasilkan output yang diharapkan.

3. *System Testing*

System Testing adalah proses pengujian dimana perangkat lunak yang diuji sudah lengkap dan terintegrasi. Tujuan dari pengujian ini adalah untuk mengevaluasi kesesuaian sistem dengan persyaratan yang telah ditentukan.

4. *Acceptance Testing*

Acceptance Testing atau uji penerimaan adalah pengujian formal dilakukan untuk menentukan apakah sistem menerima kriteria penerimaan dan memastikan jika pengguna dapat menerima sistem. Tujuan dari pengujian ini adalah untuk mengetahui tingkat kelayakan dari perangkat lunak.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Seperti *Integration Testing*, *Acceptance Testing* juga meliputi pengujian keseluruhan aplikasi. Perbedaannya terletak pada siapa yang melakukan testing. Pada tahap ini, *end-user* yang terpilih melakukan testing terhadap fungsi-fungsi aplikasi dan melaporkan permasalahan yang ditemukan. Proses ini merupakan salah satu tahap final sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru.

2.12 Metode pengujian *Blackbox Testing*

Teknik pengujian *blackbox* berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam. *Blackbox* merupakan metode pengujian perangkat lunak yang dilakukan hanya mengamati hasil eksekusi data uji dan memeriksa fungsional dari perangkat lunak. Cakupan pengujian yang dilakukan pada *blackbox testing* adalah perihal pengujian *interface* dan *form validation*. Pengujian *Interface* adalah pengujian yang dilakukan secara langsung terhadap desain *interface* yang dibuat pada sistem. Tujuan yang diharapkan dalam melakukan *blackbox testing* dapat membuat desain yang sesuai dengan kebutuhan perusahaan/masyarakat. Pengujian *blackbox testing*, dilakukan untuk menemukan hal-hal sebagai berikut

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*Interface errors*).
3. Kesalahan pada struktur data dan akses *database*.
4. Kesalahan performansi (*Performance errors*),
5. Kesalahan inisialisasi dan terminasi.

2.13 Penelitian yang Terkait

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperbanyak referensi yang digunakan dalam mengkaji penelitian yang dilakukan. Untuk melihat perbandingan penelitian sebelumnya maka dapat dilihat pada tabel 2.3 di bawah ini:

Tabel 2.3 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1.	Budi Dwi Satoto, Eza Rahmanita (2013)	Integrasi <i>Augmented reality</i> pada <i>Mobile Virtual Tour</i> berbasis android untuk pencarian lokasi dan rute terdekat	Menghasilkan tingkat akurasi posisi pengguna dalam peta <i>Google</i> dipengaruhi oleh kuat tidaknya sinyal provider dan sinyal GPS yang sampai di lokasi dengan manfaat dapat membantu pengguna dalam mendapatkan informasi pada rute perjalanan yang lebih interaktif.
2.	Alfi Syahrin, Meyti Eka Apriyani, Sandi Prasetyaningsih (2016)	Analisis dan implementasi metode Marker Base Tracking pada <i>Augmented Reality</i> pembelajaran buah-buahan	Menghasilkan jarak pendeteksian yang ideal terhadap marker yang terdiri dari jarak maksimum dan jarak minimum, jika pixel pada marker semakin terlihat maka berpengaruh pula pada keberhasilan sistem untuk menampilkan objek.
3.	Muhammad Rifa'i, Tri Listyorini, Anastasya Latubessy (2014)	Penerapan teknologi <i>Augmented Reality</i> pada aplikasi katalog rumah berbasis Android	Menghasilkan aplikasi yang berjalan pada platform mobile android dengan menggunakan AR yang diambil dari kamera smartphone kemudian dapat melacak dan mendeteksi marker (penanda) dengan menggunakan sistem tracking, setelah marker terdeteksi, model rumah 3D pada katalog akan muncul diatas marker seolah-olah model rumah tersebut nyata.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.3 Penelitian Terdahulu (Lanjutan)

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
4.	Fadilah Fahrul Hardiansyah, Yuliana Setiowati,S.K om, M.Kom, Kholid Fathoni,S.Ko m (2012)	Augmented Reality untuk mengetahui fasilitas umum berbasis Android	Menghasilkan data tempat, gedung dan object lain dengan menggunakan teknologi Augmented Reality yang didapat dari Google Places Services dan juga database aplikasi. Aplikasi ini dapat data penunjuk arah jalan yang di dapat dari Google maps juga beberapa fitur yang dapat membantu user seperti fitur tracking, get direction, find nearest places, dan juga menyimpan lokasi atau gedung baru.
5.	Tri Wenda Ginting, Virginia Tulenan, Hans Wowor (2016)	Pengenalan Gedung kampus Universitas Sam Ratulangi dengan pemanfaatan Augmented Reality dan layanan berbasis lokasi	Menghasilkan sebuah aplikasi pengenalan gedung kampus Universitas Sam Ratulangi dengan pemanfaatan augmented reality dan layanan berbasis lokasi pada smartphone dengan empat menu utama pada aplikasi ini yaitu menu AR, LBS, tentang dan keluar. Sehingga dengan aplikasi ini mempermudah untuk mengetahui lokasi gedung kampus dan memberikan informasi mengenai layanan yang disediakan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.3 Penelitian Terdahulu (Lanjutan)

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
6.	Rizal Kusumajati Nugroho, Afrizal Doewes. Abdul Aziz(2016)	Implementasi Markerless Augmented Reality Using Android Sensor For Identification of Buildings in Sebelas Maret University	Menghasilkan sebuah aplikasi identifikasi gedung kampus 1 UNS menggunakan <i>augmented reality markerless</i> pada <i>smartphone</i> android dengan menggunakan GPS, sensor akselerometer, sensor <i>magnetic field</i> (kompas) dan dari hasil pengujian black box menunjukkan bahwa aplikasi ini memiliki tingkat akurasi yang baik dibuktikan melalui pengujian terhadap 3 sampel gedung menghasilkan akurasi sebesar 92.65 %.
7.	Steven Pragestu, Herry Sujiani, Arif Bijaksana Putra Negara(2015)	Implementasi <i>Augmented Reality</i> dengan memanfaatkan <i>GPS Based Tracking</i> Pada sistem pengenalan Gedung Universitas Tanjungpura	Menghasilkan sebuah aplikasi yang dapat mengenali gedung dari depan, sisi belakang, sisi kiri maupun sisi kanan gedung dan dari jarak dekat, sedang maupun jauh. Implementasi teknologi <i>Augmented Reality</i> lebih mudah untuk mengenali gedung di UNTAN dengan menggunakan fitur AR (96%) lebih besar dibandingkan presentasi pemilih fitur <i>MAP</i> (4%)
8.	Bdur Rahman, Ernawati, Funny Farady Coastera (2014)	Rancang Bangun Aplikasi Informasi Universitas Bengkulu sebagai panduan pengenalan kampus menggunakan metode <i>Markerless Augmented Reality</i> berbasis Android	Menghasilkan sebuah aplikasi informasi kampus Universitas Bengkulu sebagai pedoman yang dapat digunakan pada <i>smartphone</i> android dengan layar <i>userfriendly</i> , mudah digunakan dan baik dalam kinerja sistem. Aplikasi ini juga memudahkan mencari data karyawan dana menemukan lokasi fasilitas kampus dengan memanfaatkan <i>Google maps</i> dan <i>Markerless Augmented Reality</i> Teknologi menunjukkan lokasi fasilitas didalam kampus.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 2.3 Penelitian Terdahulu (Lanjutan)

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
9.	Selvia Lorena Br Ginting, Meylly Pamungkas, Ypgie Rinaldy Ginting	Metode <i>Markerless</i> untuk membangun Aplikasi pemandu wisata wilayah ciayumajakuning berbasis mobile android	Menghasilkan Aplikasi yang dibutuhkan oleh wisatawan mencakup informasi posisi wisatawan berada, informasi objek wisata, jarak dari posisi wisatawan berada dengan posisi objek wisata, rute dan informasi waktu tempuh serta objek-objek penting yang terdapat dalam perjalanan. Aplikasi ini menerapkan teknologi <i>markerless augmented reality</i> yang dapat menginformasikan keberadaan POI sesuai posisi sebenarnya dengan menunjukkan posisi POI berada sesuai dengan interaksi <i>user</i> pada aplikasi terhadap lingkungan disekitarnya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.