

BAB II

LANDASAN TEORI

2.1. Futsal

Menurut Hatta (2003) olahraga futsal merupakan olahraga sepakbola mini yang dilakukan dalam ruangan dengan panjang lapangan 38-42 meter dan lebar 15-25 meter. Dimainkan oleh 5 pemain termasuk penjaga gawang. Futsal adalah permainan yang hampir sama dengan sepakbola, dimana dua tim memainkan dan memperebutkan bola diantara para pemain dengan tujuan dapat memasukkan bola ke gawang lawan dan mempertahankan gawang dari kemasukan bola.

Menurut Lhaksana (2008) sebelum berkembang menjadi cabang olahraga yang kedudukannya sejajar dengan sepakbola rumput, futsal ditekuni sebagai sarana pengarahan dan pembentukan para pemain muda yang ingin berkarir dalam bidang futsal.

2.2. Aplikasi

Menurut Jogiyanto (2001) aplikasi adalah penggunaan dalam suatu perangkat komputer, instruksi atau pernyataan yang disusun hingga sedemikian rupa komputer dapat memproses masukan (*input*) menjadi keluaran (*output*). Sedangkan menurut Kamus Besar Bahasa Indonesia (KBBI), aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Sedangkan pengembangan aplikasi dipahami sebagai suatu kegiatan dimana seorang *programmer* melakukan *coding* atau sebuah proses penulisan kode, akan tetapi yang sebenarnya dimaksud dengan pengembangan aplikasi yaitu merupakan serangkaian proses yang dilakukan dari saat pembuatan konsep aplikasi hingga aplikasi tersebut selesai dan siap digunakan.

Dalam proses pengembangan aplikasi terdapat berbagai proses yang perlu dilakukan sebagai syarat untuk membuat sebuah aplikasi yang telah terancang dengan baik, dan dikerjakan secara berstruktur. Proses pengembangan aplikasi berdasarkan terdiri dari 5 tahap utama, walaupun di dalam setiap tahap 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.

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

terdapat berbagai fase lainnya yang dapat dijalankan sesuai kebutuhan pengembangan. Tahap tersebut antara lain adalah pengumpulan informasi, perencanaan, pengembangan, penggunaan, dan perawatan.

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Aplikasi perusahaan (*enterprise*)
2. Aplikasi infrastruktur perusahaan
3. Aplikasi informasi kerja
4. Aplikasi media dan hiburan
5. Aplikasi pendidikan
6. Aplikasi pengembangan media
7. Aplikasi rekayasa produk

Saat ini, aplikasi menjadi sebuah kebutuhan termasuk aplikasi yang menggunakan teknologi *Global Positioning System* (GPS) yang berfungsi untuk melakukan pemetaan secara *real time*.

2.3. *Mobile Application*

Menurut Irwansyah (2014) *mobile application* adalah aplikasi perangkat lunak yang dibuat khusus untuk dijalankan di dalam tablet dan juga *smartphone*. Umumnya, *developer mobile apps* memerlukan IDE atau *Integrated Development Environments* dan juga SDK untuk pengembangan dari *mobile apps* itu sendiri.

Pada saat ini, pada *smartphone* dan juga tablet, ada satu aplikasi yang berguna untuk menyediakan berbagai macam aplikasi yang dapat dijalankan di *device* tersebut. Aplikasi ini sering disebut *store*. Contoh *store* yaitu *Apple Apps Store*, *Samsung Apps*, *Amazon Kindle Fire*, *Windows Store* dan *Google Playstore*.

Aplikasi *mobile* berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Kata *mobile* mempunyai arti bergerak atau berpindah, sehingga aplikasi *mobile*

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

menurut Purnama (2010) adalah sebutan untuk aplikasi yang berjalan di perangkat *mobile*.

Dengan menggunakan aplikasi *mobile*, dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh pengguna telepon seluler, karena dengan memanfaatkan adanya fitur *game*, *music player*, sampai *video player* membuat kita mejadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

2.4. Navigasi

Proses navigasi dalam *Google Maps* dapat dilakukan dengan 2 cara, yaitu:

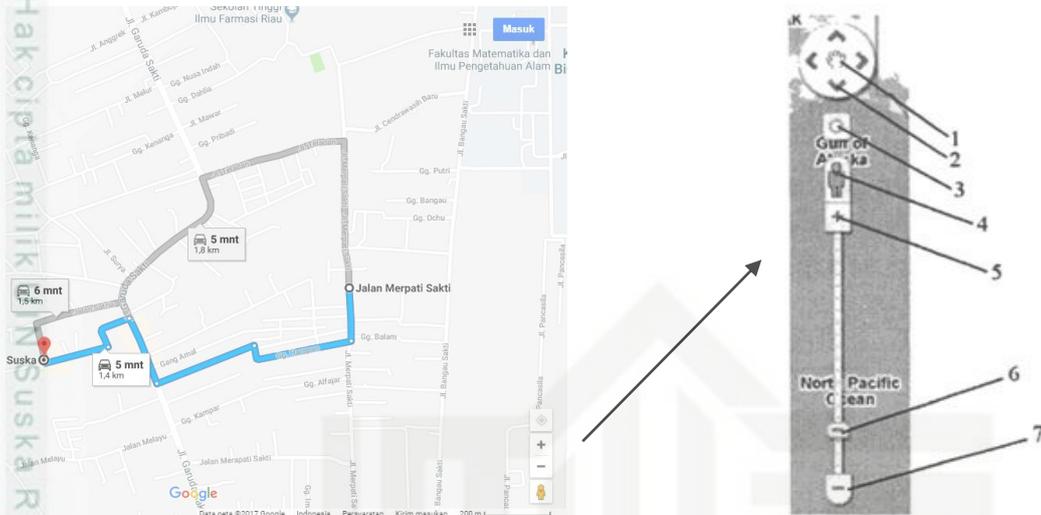
1. Navigasi dengan *mouse*

Untuk memulai navigasi menggunakan *mouse*, letakkan *mouse* di area tengah peta hingga kursor berubah menjadi gambar tangan. Hal hal yang dapat kita lakukan dengan *mouse* antara lain:

- a. Memperbesar atau mengecilkan tampilan peta dengan cara klik kanan pada *mouse*, kemudian pilih *zoom in* untuk memperbesar tampilan peta dan klik *zoom out* untuk memperkecil tampilan peta. Selain itu dapat pula dengan menggunakan *mouse scroll wheel* untuk memperbesar dan memperkecil peta, yaitu dengan cara menggerakkan *scroll* ke atas untuk melakukan perbesaran tampilan dan gerakkan *scroll* ke bawah untuk memperkecil tampilan.
- b. Menggeser tampilan ke berbagai arah dengan cara klik kiri dan tahan pada *mouse* kemudian geser ke arah yang kita inginkan, baik itu ke kiri, kekanan, ke atas maupun ke bawah.

2. Navigasi dengan kontrol navigasi

Selain dengan menggunakan *mouse* untuk bernavigasi, anda dapat pula memanfaatkan fitur kontrol navigasi yang sudah disediakan oleh *Google Maps*. Berikut penjelasan mengenai penggunaan kontrol navigasi yang disediakan oleh *Google Maps* yang dapat dilihat pada Gambar 2.1



Gambar 2.1. Navigasi Kontrol pada *Google Maps*

Keterangan dari Gambar 2.1:

1. Untuk kembali ke hasil sebelumnya.
2. Untuk menggerakkan tampilan sesuai dengan arah tanda panah.
3. Untuk menunjukkan lokasi anda berada sekarang.
4. Klik geser ikon tersebut dan letakkan sesuai dengan lokasi yang anda tentukan, maka anda dapat menuju pencitraan tingkat jalan.
5. Untuk memperbesar tampilan peta.
6. Tarik *zoom slider* ke atas untuk memperbesar tampilan peta, dan tarik ke bawah untuk memperkecil tampilan peta.
7. Untuk memperkecil tampilan peta.

2.5. *Google Maps*

Menurut Huda (2014) *Google Maps* adalah jasa peta gratis dan *online* disediakan oleh *Google* yang dapat ditemukan di <http://maps.google.com>. Pada situs tersebut kita dapat melihat informasi geografis pada hampir semua wilayah di muka bumi. Layanan ini interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah tingkat *zoom*, serta mengubah tampilan peta. *Google Maps* juga menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia, serta menawarkan rute perjalanan.

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.

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

keoptimalan berjalan diatas perangkat *hardware* yang ada. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka.

2.7. Metode *Object-Oriented Analysis and Design* (OOAD)

OOAD adalah metode pengembangan sistem yang lebih menekankan objek dibandingkan dengan data atau proses. Metode OOAD melakukan pendekatan terhadap masalah dari perspektif objek, tidak pada perspektif fungsional seperti pada pemrograman terstruktur. Hal tersebut untuk memenuhi peningkatan kebutuhan akan pendekatan berorientasi objek pada aplikasi bisnis. Tahap perancangan dimulai dengan hasil keluaran yang dihasilkan tahapan analisis dan aktivitas yang dilakukan adalah secara perlahan bergeser tekanannya dari domain aplikasi atau persoalan atau masalah menuju ke domain komputasi.

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu:

1. *Encapsulation*

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.

2. *Inheritance*

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek.

3. *Polymorphism*

Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon *message* yang sama.

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.

2.8. Tools Yang Digunakan

2.8.1. Personal Home Page (PHP)

Personal Home Page (PHP) merupakan bahasa *scripting* yang *open source* dan digunakan untuk membuat situs web yang dinamis dan *powerful*. Berikut ini adalah beberapa modul yang tersedia untuk PHP:

1. PHP menyediakan kemampuan untuk melakukan koneksi terhadap berbagai macam *database* diantaranya adalah Adabas, dBase, Empree, FilePro, Informix, Interbase, mSQL, MySQL, Oracle, PostgreSQL, Solid, Sybase, Velocis, dan beberapa *database* jenis dbm milik Unix.
2. *eXtensible Markup Language* (XML) juga didukung oleh PHP. XML sendiri adalah format lain dari HTML yang memisahkan isi atau informasi yang disimpan oleh halaman *web* dari presentasi dari halaman tersebut.
3. PHP mendukung proses *upload file* (seperti yang disebutkan oleh RFC-1867) *binary* atau *file* teks.
4. PHP dapat membaca dan men-*setting cookie* pada HTTP.
5. PHP mempunyai *regular expression* untuk memanipulasi *string* yang kompleks.

2.8.2. MySQL

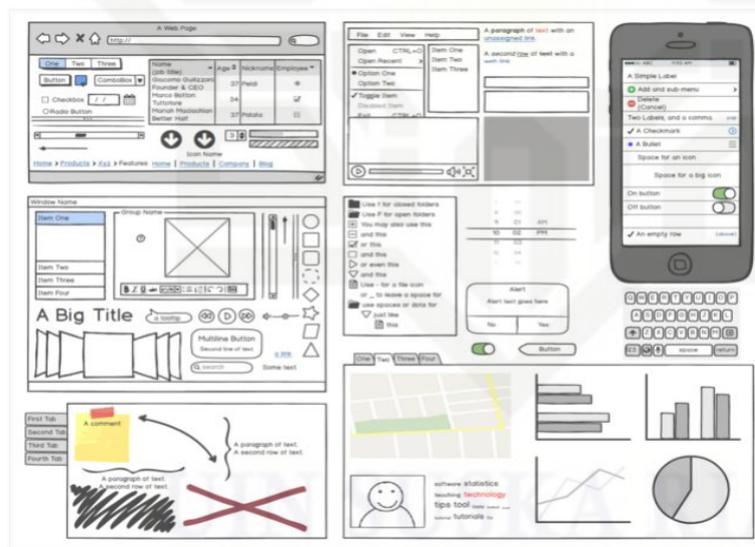
Menurut (Anhar, 2010) *My Structure Query Language* (MySQL) adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lainnya. MySQL berfungsi untuk mengolah *database* menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemograman PHP juga sangat mendukung/*support* dengan *database* MySQL.

MySQL adalah sebuah sistem manajemen basis data relasi (*relational database management system*) yang bersifat terbuka (*open source*). Terbuka maksudnya adalah MySQL boleh di *download* oleh siapa saja, baik versi kode program aslinya maupun versi binernya dan bisa digunakan secara (*relative*) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

2.8.3. Balsamiq Mockup 3

Balsamiq Mockup adalah program aplikasi yang digunakan dalam pembuatan tampilan *user interface* sebuah aplikasi. *Software* ini sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan kita buat. *Software* ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

Alih-alih menggambar sketsa (*wireframe*) atau *prototype* rancangan desain website di atas kertas Balsamiq Mockups membantu seorang web desainer membuat tampilan web dalam bentuk gambar di komputer. Tujuannya selain agar membuat tampilan (desain) website menarik juga dapat menyesuaikan dengan kebutuhan *customer* (pelanggan). Dengan alat pembuat *Mockup* maka seorang web desainer dapat menganalisa tata letak, desain dan fungsi. Berikut ini adalah contoh gambar tampilan dari aplikasi Balsamiq yang dapat dilihat pada Gambar 2.2 di bawah ini.



Gambar 2.2. Contoh Tampilan Balsamiq

Kelebihan Balsamiq Mockup dibanding *software* pembuat *Mockup* lainnya adalah aplikasi ini berbasis *cloud*, disertai aplikasi *desktop* yang memungkinkan kita dengan cepat dan mudah membuat rancangan website. Dengan konten yang terbuat seperti dari gambaran tangan, akan membuat kita fokus pada pemecahan masalah *user interface* yang lebih besar, daripada pada

- 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.

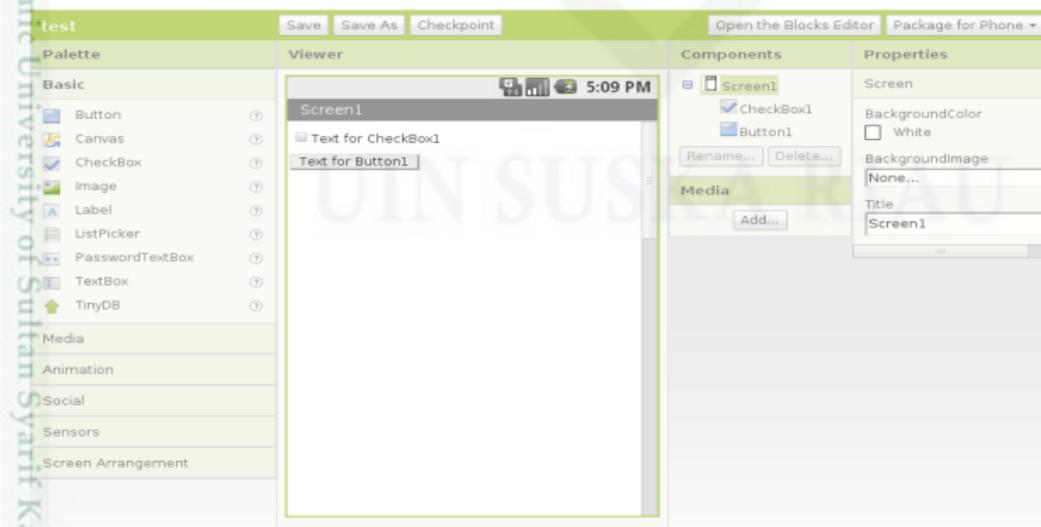
perincian website. Di websitenya sendiri ada dua pilihan untuk para pengguna, ada versi *trial for dekstop* dan ada juga yang bisa kita *download* untuk versi *dekstop*. Namun ada juga yang disediakan dalam versi berbayar. Aplikasi ini bisa digunakan untuk sistem operasi Windows, Mac OS, dan Linux.

2.8.4. MIT App Inventor 2

App Inventor untuk Android adalah sebuah aplikasi *web open-source* asli yang disediakan oleh Google, dan sekarang dikelola oleh *Massachusetts Institute of Technology* (MIT).

MIT App Inventor adalah sebuah inovasi pengantar bagi pemula untuk membuat *program* dan menciptakan aplikasi yang mengubah bahasa yang rumit dari *coding* berbasis *text* ke dalam bentuk *visual*, dengan *men-drag-and-drop* bangunan berupa blok-blok. Merupakan antar muka dengan grafik sederhana yang bahkan memberikan kemampuan bagi pemula untuk membuat suatu aplikasi yang berfungsi penuh dalam waktu satu jam atau kurang dari satu jam. (appinventor.mit.edu, 2012)

Misi dari MIT App Inventor yaitu berusaha untuk mendemokrasiikan pengembangan perangkat lunak dengan memberdayakan semua orang, terutama remaja, untuk berpindah dari konsumen teknologi menjadi pencipta teknologi. (appinventor.mit.edu, 2012)



Gambar 2.3. MIT App Inventor 2

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.

App Inventor meliputi:

1. Halaman Desainer, di mana komponen program yang ditentukan. Ini termasuk komponen yang terlihat, seperti tombol dan gambar, yang ditempatkan pada layar simulasi, dan komponen *non*-terlihat, seperti sensor dan koneksi *web*.
2. Sebuah *editor* blok, di mana logika program dibuat. Sebuah kompiler berdasarkan kerangka sesuai dengan skema bahasa pemrograman. Diagram blok yang berbentuk *puzzle* memudahkan pengguna untuk menyusun algoritma pemrograman.

2.8.5. Unified Modelling Language (UML)

Menurut Sholiq (2006) 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.

Tujuan utama perancangan UML adalah:

1. Menyediakan bahasa pemodelan Visual yang Ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
2. Mendukung spesifikasi independen bahasa pemograman dan proses pengembangan tertentu.
3. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
4. Mendorong pertumbuhan pasar kaku berorientasi objek.
5. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

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

a. *Use case Diagram*

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tujuan utama pemodelan *use case* adalah:

- 1) Memutuskan dan mendeskripsikan kebutuhan-kebutuhan fungsional sistem.
- 2) Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *use case* digunakan diseluruh proses pengembangan untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya yang mengacu sistem harus memberikan fungsionalitas yang dimodelkan para *use case*.
- 3) Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.
- 4) Menyediakan kemampuan melacak kebutuhan fungsionalitas menjadi kelas-kelas dan operasi-operasi *actual* di sistem. Untuk menyederhanakan perubahan dan ekstensi ke sistem dengan mengubah model *use case* dan kemudian melacak *use case* yang dipengaruhi ke perancangan dan implementasi sistem.

Syarat penamaan *Use case* adalah nama didefenisikan sesederhana 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 di buat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum tentu orang.
- 2) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Hak Cipta Dilindungi Undang-Undang

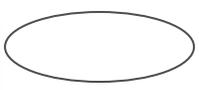
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan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

Dalam *use case* juga dikenal dengan hubungan antar *use case* yang merupakan generalisasi antara *use case* yaitu:

- 1) *Include*, yaitu perilaku *use case* merupakan bagian dari *use case* yang lain.
- 2) *Extend*, yaitu perilaku *use case* memperluas perilaku *use case* yang lain.

Berikut adalah tabel dari *use case diagram* yang dapat dilihat pada Tabel 2.1.

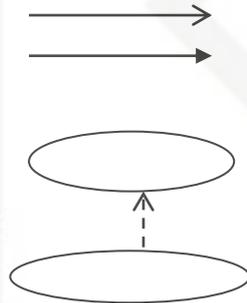
Tabel 2.1. Simbol-Simbol *Use Case Diagram*

Simbol	Nama	Deskripsi
	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang tertukar pesan antar unit atau aktor biasanya dinyatakan dengan menggunakan kata kerja di awal di awal <i>frase</i> nama <i>use case</i> .
	Aktor/ <i>Actor</i>	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 dari orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor.
	Asosiasi/ <i>association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

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 2.1. Simbol-simbol *Use Case Diagram* (Lanjutan)

Simbol	Nama	Deskripsi
	Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> .
	Generalisasi/ <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum.
	Menggunakan/ <i>include/users</i>	<i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan.

b. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu di perhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem.

Diagram aktivitas juga banyak di gunakan untuk mendefenisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefenisikan.

2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan tampilan dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu di defenisikan kasus ujinya.

Berikut adalah tabel dari *activity diagram* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2. Simbol-Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

c. *Class Diagram*

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Elemen-elemen esensi di diagram kelas adalah sebagai berikut:

1. Kelas

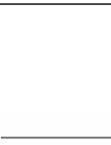
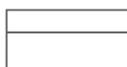
Kelas merupakan elemen terpenting di sistem berorientasi objek. Kelas memiliki sejumlah fitur, seseorang dapat memodelkan multipisitas, ketampakan, penenda, *polymorphism*, dan karekteristik lain.

- a. Nama, nama harus unik, karena akan menjadi identifier di program.
- b. Atribut adalah *property* bernama dikelas yang mendiskripsikan *range* nilai yang dipunyai instan kelas.

- c. Operasi adalah layanan yang dapat diminta pada sembarang objek kelas itu untuk mempengaruhi perilaku sistem.
 - d. Tanggung jawab adalah kontrak atau kewajiban kelas.
 - e. Lingkup menspesifikasikan apakah fitur (atribut atau operasi) muncul di masing-masing instan dan kelas.
2. Antarmuka (*interfaces*)
 Merupakan koleksi operasi yang menspesifikasikan layanan dari suatu kelas atau komponen. Antarmuka mendiskripikan perilaku tampak secara eksternal dari elemen.
3. Kolaborasi
 Merupakan pendefinisian suatu interaksi dan sekelompok peran dan elemen-elemen lain yang bekerja bersama untuk menyediakan suatu perilaku kooperatif yang lebih besar dari penjumlahan seluruh elemen.
4. Hubungan (*releationship*) seperti kebergantungan, generalisasi, dan asosiasi. Hubungan kelas di diagram kelas beraneka ragam, yaitu:
- a. Asosiasi
 - b. Generalisasi
 - c. *Dependency*

Berikut adalah tabel dari *class diagram* yang dapat dilihat pada Tabel 2.3.

Tabel 2.3. Simbol-Simbol *Class Diagram*

Simbol	Nama	Deskripsi
	<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>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

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. Simbol-Simbol *Class Diagram* (Lanjutan)

Simbol	Nama	Deskripsi
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

d. *Sequence Diagram*

Diagram *sequence* merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

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.

Berikut adalah tabel dari *sequence diagram* yang dapat dilihat pada Tabel 2.4.
Tabel 2.4. Simbol-Simbol *Sequence Diagram*.

Simbol	Nama	Deskripsi
 Nama aktor	Aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang; biasanya di nyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.
	Garis hidup / <i>lifeline</i>	Menyatakan hidup suatu objek.
 Nama objek : nama kelas	Objek	Menyatakan objek yang berinteraksi pesan.
	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
 <<create>>	Pesan tipe <i>create</i>	Objek yang lain, arah panah mengarah pada objek yang di buat.

2.9. Website

Website adalah suatu jaringan dari dokumen-dokumen elektronik yang disebut halaman web, yang isinya dapat berupa teks, grafis, dan bahkan format suara dan format video. Dokumen-dokumen tersebut terintegrasi dengan *hyperlinks*. *Hyperlinks* memungkinkan *user* untuk mendapatkan informasi yang diinginkan dari satu halaman ke halaman lainnya menggunakan *link* tersebut dengan mudah.

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.

sebuah transaksi jual-beli. Pada saat reservasi berlangsung biasanya ditandai dengan adanya proses tukar menukar informasi antara konsumen dan produsen agar kesepahaman mengenai produk dapat terwujud.

Alasan *reservasi* menjadi sebuah media yang sangat efektif baik bagi produsen maupun bagi konsumen adalah produsen akan dapat melakukan evaluasi terhadap produk yang akan mereka jual melalui tingkat tinggi rendahnya jumlah *reservasi* jauh sebelum produk tersebut dijual (barang) ataupun diselenggarakan (jasa), dimana hasil evaluasi tersebut akan membantu produsen untuk menentukan langkah pemasaran yang akan diambil terhadap produk yang akan dijual tersebut.

2.12. Pengumpulan Data

2.12.1. Observasi

Observasi berasal dari bahasa latin yang berarti “melihat” dan ”memperhatikan”. Istilah observasi diarahkan pada kegiatan memperhatikan secara akurat, mencatat fenomena yang muncul dan mempertimbangkan hubungan antar aspek dalam fenomena tersebut. Observasi merupakan metode pengumpulan data esensial dalam penelitian baik secara umum pada konteks maupun dengan fokus-fokus khusus.

Tujuan observasi adalah mendeskripsikan setting yang dipelajari, aktivitas-aktivitas berlangsung, orang yang terlibat dalam aktivitas dan makna kejadian yang diamati tersebut. Deskripsi harus akurat, aktual sekaligus teliti tanpa harus dipenuhi berbagai catatan panjang lebar yang tidak relevan. Peneliti yang baik akan melaporkan hasil observasinya secara deskriptif tidak interpretatif. Pengamat tidak mencatat kesimpulan atau interpretasi, melainkan data konkrit berkenaan dengan fenomena yang diamati.

Hasil dari Observasi yang dilakukan pada penelitian ini dapat dilihat pada Lampiran B di Laporan ini.

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.

2.12.2. Wawancara

Wawancara adalah percakapan dengan maksud tertentu. Percakapan itu dilakukan oleh dua pihak, yaitu pewawancara (*interviewer*) yang mengajukan pertanyaan dan terwawancara (*interviewee*) yang memberikan jawaban atas pertanyaan itu. Wawancara kualitatif dilakukan bila peneliti bermaksud untuk memperoleh pengetahuan tentang makna yang berkenaan dengan topik yang diteliti. Wawancara terstruktur artinya peneliti telah mengetahui dengan pasti apa informasi yang ingin digali dari responden sehingga daftar pertanyaannya sudah dibuat secara sistematis. Peneliti juga dapat menggunakan alat bantu *tape recorder*, kamera photo, dan material lain yang dapat membantu kelancaran wawancara.

Beberapa kelebihan wawancara tatap muka antara lain:

1. Bisa membangun hubungan dan memotivasi responden
2. Bisa mengklarifikasi pertanyaan, menjernihkan keraguan, menambah pertanyaan baru
3. Bisa membaca isyarat non verbal
4. Bisa memperoleh data yang banyak

Hasil dari Wawancara yang dilakukan pada penelitian ini dapat dilihat pada Lampiran A di Laporan ini.

2.13. *Black Box Testing*

Black box testing berkaitan dengan pengujian-pengujian yang dilakukan pada antarmuka perangkat lunak. Pengujian ini juga sering disebut pengujian fungsional karena penguji hanya melakukan pengujian pada perangkat lunak yang berkaitan dengan fungsionalitas dan bukan pada implementasi perangkat lunak. Pengujian ini berfungsi untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi kebutuhan. Berdasarkan penjelasan diatas dapat disimpulkan bahwa *black box testing* merupakan pengujian yang berfokus pada fungsional sistem serta berfungsi untuk mengetahui apakah fungsi *input* dan *output* sistem telah sesuai dengan kebutuhan (Ratnawati, 2016).

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

Berikut ini merupakan rumus perhitungan dari hasil pengujian dengan menggunakan metode *black box testing*.

$$\text{Persentase keberhasilan} = \frac{\text{Jawaban berhasil}}{\text{Jumlah pertanyaan}} \times 100 \dots\dots\dots(2.1)$$

2.14. Pengertian *User Acceptance Testing* (UAT)

Pengujian UAT atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa *software* yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna. Proses UAT didasarkan pada dokumen *requirement* yang disepakati bersama. Dokumen *requirement* adalah dokumen yang berisi lingkup pekerjaan *software* yang harus dikembangkan, dengan demikian maka dokumen ini semestinya menjadi acuan untuk pengujian.

Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil pekerjaan. Diperiksa apakah *item-item* yang ada dalam dokumen *requirement* sudah ada dalam *software* yang diuji atau tidak. Diuji apakah semua *item* yang telah ada telah dapat memenuhi kebutuhan penggunanya. Skenario UAT adalah suatu rencana yang disusun untuk dijalankan sesuai dengan urutan yang telah ditetapkan.

Suatu skenario akan mencakup perencanaan pelaksanaan dan proses pengujian dari awal pemasangan *software* sampai dengan akhir proses. Suatu skenario tidak boleh lepas dari dokumen kebutuhan. Hasil dari UAT adalah dokumen yang menunjukkan bukti pengujian, berdasarkan bukti pengujian inilah dapat diambil kesimpulan, apakah *software* yang diuji telah dapat diterima atau tidak.

Menurut Zarnelly (2014) *User Acceptance Test* yaitu pengujian yang dilakukan oleh pengguna dari sistem untuk memastikan fungsi- fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Proses UAT didasarkan pada dokumen *requirement* yang disepakati bersama. Dokumen *requirement* adalah dokumen yang berisi lingkup pekerjaan

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.

software yang harus dikembangkan, dengan demikian maka dokumen ini semestinya menjadi acuan untuk pengujian. *User Acceptance Test* adalah proses pengujian oleh *user* dan menghasilkan dokumen untuk dijadikan bukti bahwa aplikasi yang dikembangkan dapat diterima *user* dan hasil pengujiannya dianggap memenuhi kebutuhan pengguna.

Berikut ini merupakan rumus perhitungan dari hasil pengujian dengan menggunakan metode *User Acceptance Testing*.

$$\text{Persentase keberhasilan} = \frac{\text{Jawaban berhasil}}{\text{Jumlah pertanyaan}} \times 100 \dots\dots\dots(2.2)$$

