



BAB 2

LANDASAN TEORI

2.1 Aplikasi

Menurut Koyuko, Sinsuw, dan Najoan (2016) aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya. *suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut. dan juga aplikasi-aplikasi tersebut mempunyai bisa berinteraksi antar aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolahan kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.

2.2 Monitoring

Monitoring dapat dikatakan suatu kegiatan yang merangkup dalam proses pengumpulan, meninjau, melaporkan, dan tindakan dari suatu informasi kegiatan yang sedang dikerjakan (Koyuko dkk., 2016). Pada dasarnya, *monitoring* dilakukan untuk pemeriksaan antara kinerja dan target yang telah di tentukan dalam sesuatu kegiatan. Adapun alat bantu yang digunakan dalam melakukan sistem *monitoring* yaitu *observasi* atau wawancara, dokumentasi dan menggunakan aplikasi visual. *Monitoring* mempunyai dua kegunaan yang berhubungan, yaitu *compliance monitoring* dan *performance monitoring*. *Compliance monitoring* berguna untuk memastikan proses telah sesuai dengan yang diharapkan/ direncanakan. Sedangkan *performance monitoring* berguna untuk mengetahui kemajuan di dalam organisasi apakah telah mencapai target yang di harapkan.

Monitoring adalah suatu proses kegiatan pengukuran, pencatatan, pengumpulan, memproses dan meng-komunikasi-kan informasi untuk membantu dalam pengambilan keputusan proyek Primazni (2016). Sistem *monitoring* adalah suatu kegiatan mengumpulkan data dari berbagai sumber daya. Data yang dikumpulkan merupakan data yang *real time*.

Menurut Dewi dkk. (2015) Tujuan *Monitoring*:

1. Meninjau apakah kegiatan yang telah dilakukan sudah berjalan dengan rencana yang ditentukan.
2. Mengidentifikasi masalah yang datang sehingga bisa langsung diatasi agar



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.

tujuan yang dicapai sesuai.

3. Mengevaluasi apakah sistem kegiatan dan manajemen yang digunakan sudah benar untuk mencapai tujuan.
4. Untuk mengetahui hubungan antara kegiatan dengan tujuan untuk mendapat hasil kemajuan yang telah dicapai.
5. Menyesuaikan kegiatan yang berubah dengan lingkungan, agar tidak terjadinya penyimpangan.

2.3 Android

Menurut Rismayani dan Hasyrif (2016), *android* adalah suatu system operasi yang digunakan untuk perangkat *mobile* yang berbasis linux yang menyediakan platform terbuka untuk pengembang membuat aplikasi sendiri yang pakai oleh berbagai macam piranti bergerak. *Android* rilis pertama kali pada 5 November 2007, *android* bersama dengan *Open Handset Alliance* menyatakan dukungan untuk pengembangan standar terbuka pada perangkat seluler. *Google* menulis *code android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. kemudian tanggal 9 Desember 2008 terdapat 14 anggota baru yang bergabung pada proyek *Android* diantaranya *Sony Ericsson*, *Toshiba Corp*, *Asustek Computer Inc* dan *Vodafone Group Inc*.

Sedangkan menurut Kasenda dkk. (2016) *android* adalah sistem operasi berbasis *linux* yang dipakai untuk *smartphone* dan komputer tablet. *Android* memberikan tempat terbuka untuk pengembang untuk membuat aplikasi sendiri tanpa batas. Beberapa fitur yang ada di *android: Framework*, grafik 2 dimensi, 3 dimensi, kamera, dan beberapa media. *Android* juga terpecah menjadi beberapa lapisan: *Linux kernel*, *Libraries*, *Android runtime*, dan *Application framework*.

2.4 Metode Waterfall

Menurut Aprisa dan Monalisa (2015), Beberapa tahapan metode *Waterfall* sebagai berikut:

1. *Requirement Analysis*

Pada fase ini semua kebutuhan sistem dibuat, termasuk didalamnya kegunaan sistem, yang diharapkan *user* dan batasan sistem. Informasi ini dapat diperoleh melalui *interview*, *observasi* maupun berdiskusi.

2. *System Design*

Tahap selanjutnya *system design*, tahap ini akan menghasilkan arsitektur sistem secara keseluruhan. Bertujuan untuk memberikan gambaran atau alur sistem yang akan di buat, dan juga membantu dalam men-spesifikasi-kan kebutuhan *hardware* dan *system*.



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. *Implementation*

Selanjutnya tahap *implemention* yaitu dilakukan tahapan pemrograman. Pada tahapan pembuatan, *software* dibagi menjadi beberapa komponen-komponen kecil yang nantinya akan digabungkan pada tahap selanjutnya. Setelah membuat komponen-komponen kecil tersebut dilakukan pemeriksaan apakah fungsinya sudah sesuai yang diinginkan.

4. *Integration & Testing*

Integrattion & Testing yaitu proses penggabungan komponen-komponen kecil yang telah dibuat untuk dilakukan pengujian agar dapat mengetahui apakah sistem yang dibuat sesuai dengan desain.

5. *Operation & Maintenance*

Tahap ini adalah tahapan terakhir dalam model *waterfall*. *Software* yang telah dibuat, dijalankan serta dilakukan *maintenance*. *Maintenance* termasuk dalam memperbaiki bagian yang salah yang tidak ditemui pada langkah testing sebelumnya.

2.5 *Object Oriented Analysis & Design (OOAD)*

OOAD adalah cara untuk menganalisis proses pemeriksaan requirement dari sudut pandang kelas-kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem (Primazni, 2016). Ada beberapa entitas mewakili setiap obyek, yang terdapat dalam system menjadi model dan kelas, elemen data, dan perilaku. Analisis berorientasi objek (OOA) menggunakan objek *modeling* teknik untuk menganalisis fungsional persyaratan untuk sistem.

2.6 *Unifiel Modeling Language (UML)*

UML yaitu alat bantu dalam pengembangan sistem berorientasi objek. karena UML memberikan bahasa pemodelan visual yang bisa dimanfaatkan pengembangan sistem untuk membuat *blue print* dalam tujuan pengembang dalam bentuk baku, mudah dimengerti serta mekanisme yang mudah untuk berbagi dan menghubungkan rancangan mereka yang lain (Nugroho, 2005).

Dalam proyek pengembangan sistem, fokus utama dalam analisis dan perancangan adalah model. Dengan model kita bisa menjelaskan sesuatu karena, model sangat efektif dan efesien, model bisa dipakai untuk simulasi dalam mendalami lebih detail tentang sesuatu, model juga bisa dikembangkan, konsep model bisa mewakili sesuatu yang nyata maupun tidak nyata.



Hak Cipta Dilindungi Undang-Undang

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

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

2.6.1 Use Case Diagram

Use case diagram menggambarkan hubungan antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* dipakai untuk mengetahui kegunaan yang ada di dalam sebuah sistem informasi dan siapa yang berhak menggunakan fungsi tersebut. Tujuan pemodelan *use case* (Nugroho, 2005) adalah:

- Menentukan dan menjelaskan kebutuhan-kebutuhan fungsional sistem.
- Menjelaskan gambaran yang jelas kegiatan apa yang dilakukan, sehingga model *use case* digunakan diseluruh proses pengembangan, untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya, yang mengacu pada sistem yang harus memberikan fungsionalitas yang dimodelkan pada *use case*.
- Menyediakan tempat untuk pengujian sistem, Dan menguji apakah sistem telah melakukan fungsi yang diminta.
- Menyediakan kemampuan mencari kebutuhan fungsionalitas menjadi kelas-kelas dan operasi-operasi aktual di sistem.

Ketntuan dalam pemberian nama *use case* diagram adalah nama didefenisikan sesederhana dan mudah dipahami, dua hal dasar dalam pendefenisian *use case* yaitu apa yang disebut aktor dan *use case* (Shalahuddin dan Rosa, 2013).

- Aktor adalah pengguna, proses, atau sistem lain yang berhubungan dengan sistem yang akan dibuat dan juga diluar sistem yang akan dibuat, simbol dari aktor adalah gambar orang tapi aktor belum tentu orang.
- Use case* adalah fungsionalitas yang diberikan sistem sebgaia unit-unit yang saling bertukar pesan antara unit atau aktor.

Simbol-simbol *use case* dapat dilihat pada Tabel 2.1.

Tabel 2.1. *Use case* diagram (Shalahuddin dan Rosa, 2013)



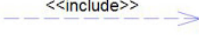

Simbol	Nama	Keterangan
	Use Case	Fungsionalitas diberikan sistem sebagai unit bertukar pesan antar unit atau aktor, biasanya dinyatakan digunakan kata kerja di awal frase nama <i>use case</i> .
	Aktor	Orang, proses, atau sistem lain berhubungan dengan sistem dibuat diluar sistem dibuat itu sendiri, jadi meskipun simbol aktor ialah gambar orang, tapi aktor belum tentu merupakan orang.

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

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

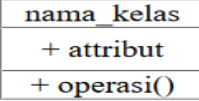

Tabel 2.1 *Use case diagram* (Shalahuddin dan Rosa, 2013) (Tabel lanjutan...)

Simbol	Nama	Keterangan
	Asosiasi/ <i>association</i>	Komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
	Ektensi/ <i>extend</i>	Menambah sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat dipakai tanpa <i>use case</i> tambahan.
	Include	<i>Include</i> mengecek apakah <i>use case</i> yang ditambahkan telah dijalankan.
	Generelisasi/ <i>generalization</i>	Hubungan umum ke khusus antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.


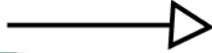


2.6.2 Class Diagram

Class diagram menggambarkan bentuk sistem dari sisi pendefenisian kelas-kelas yang dibuat untuk membuat sistem. *Class diagram* dibuat agar programmer membuat kelas-kelas sesuai rancangan supaya dokumentasi, perancangan dan *software* menjadi sesuai dengan sistem yang akan dibuat. (Shalahuddin dan Rosa, 2013). Simbol-simbol *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2. *Class diagram* (Shalahuddin dan Rosa, 2013)

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Tabel 2.2 *Class diagram* (Shalahuddin dan Rosa, 2013) (Tabel lanjutan...)

Simbol	Nama	Keterangan
	Asosiasi/ <i>association</i>	Hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generelisasi/ <i>generalization</i>	Hubungan antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Kebergantungan/ <i>depedency</i>	Hubungan antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Agregasi/ <i>aggregation</i>	Hubungan antar kelas dengan makna semua-bagian.

2.6.3 *Acitivity Diagram*

Activity diagram mendeskripsikan *workflow* (aliran kerja) atau kegiatan dari suatu sistem atau proses bisnis. *activity diagram* adalah diagram yang menggambarkan kegiatan sebuah sistem bukan yang dikerjakan aktor, tetapi kegiatan apa yang bisa dikerjakan oleh sistem.

Activity diagram digunakan untuk mendefinisikan hal-hal berikut (Shalahuddin dan Rosa, 2013):

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Pengelompokkan *interface* dari sistem (*user interface*) dimana setiap kegiatan dianggap memiliki sebuah rancangan *interface* tampilan.
3. Rancangan pengujian setiap kegiatan dianggap memerlukan sebuah pengujian yang perlu dijelaskan kasus pengujiannya.
4. Rancangan menu yang ditampilkan di *software*.



Hak Cipta Dilindungi Undang-Undang


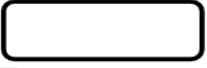
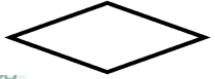


1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

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

Simbol-simbol *activity* diagram dapat dilihat pada Tabel 2.3.

Tabel 2.3. *Activity* diagram (Shalahuddin dan Rosa, 2013)

Simbol	Nama	Keterangan
	Status Awal	Status awal dari kegiatan sistem.
	Aktivitas	Kegiatan yang dilakukan sebuah sistem.
	Percabangan/ <i>decision</i>	Asosiasi pembagian jika ada pilihan aktivitas lebih dari satu.
	Penggabungan/ <i>join</i>	Asosiasi menggabungkan dari beberapa aktivitas.
	Status akhir	Status akhir yang dijalankan sistem.

2.6.4 Sequence Diagram

Menurut Rismayani dan Hasyrif (2016), *sequence* diagram (diagram urutan) adalah suatu diagram yang menampilkan hubungan antar objek di sistem yang disusun pada sebuah urutan. Hubungan antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/ message. *Sequence* Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu.

2.7 Front End dan Back End

2.7.1 Pengertian Front End

Menurut Arhandi (2016), *front end* adalah segala sesuatu yang menghubungkan antara user dengan sistem *back end*. Biasanya merupakan sebuah *user interface* dimana user akan berinteraksi dengan sistem. Pekerjaan yang sering muncul sebagai seorang *front end developer* adalah *desainer user interface* dan *desainer user experience*. Seorang front end developer hanya berfokus ke interface sistem, desain grafis dan membuat tampilan atau desain yang nyaman di gunakan



oleh pengguna. Seorang *front end developer* tidak bertugas untuk program atau aplikasi. Dan biasanya bahasa pemrograman yang digunakan dalam pengembangan *front end* adalah html.

2.7.2 Pengertian Back End

Back end atau sering di sebut *server side* pada dasarnya adalah tempat dimana proses sebuah aplikasi atau sistem berjalan, proses di *back end* biasnya untuk menambahkan, menguubah atau menghapus data (Arhandi, 2016). *Back end* biasanya tidak langsung berinteraksi kepada *user*, yaitu seperti *database* dan *server*. Biasanya orang yang bekerja sebagai *back end developer* adalah *programmer* atau *developer* yang fokus pekerjaannya pada keamanan, desain sistem, dan manajemen pada sistem. *Back end developer* dibutuhkan dalam pengembangan sistem atau aplikasi dinamis yang memiliki data yang selalu berubah ubah, contoh *website* dinamis antara lain *facebook* dan *google*. Seorang *back end developer* biasanya haru menguasai bahasa pemrograman yang dapat digunakan untuk mengelola *database*, mengolah file dan I/O seperti PHP, ASP, dan NodeJs.

2.8 Rancang Bangun

Menurut Hasyim (2014) Perancangan adalah serangkaian mekanisme untuk menterjemahkan hasil analisis dan sebuah sistem ke dalam bahasa pemrograman untuk menjelaskan secara rinci dari komponen sistem diimplementasikan. Perancangan bertujuan untuk memberikan gambaran secara lengkap kepada progremmer sistem apa yang akan dibuat. Supaya sistem yang telah jadi mudah digunakan oleh *user*.

Pengertian pembangunan atau bangun sistem adalah aktifitas membuat sebuah sistem baru atau perbaikan sistem yang ada secara menyeluruh.

Menurut Hasyim (2014) jadi dapat disimpulkan bahwa Rancang Bangun adalah menggambarkan, merencanakan, dan membuat desain dari beberapa bagian yang terpisah dan meggambukannya menjadi satu bagian. Dengan begitu dapat diartikan rancang bangun adalah menjelaskan hasil analisis ke dalam bentuk sebuah *software* kemudian membuat sistem tersebut atau memperbaiki sistem yang ada.

2.9 Google maps API

Menurut Minarni dan Yusdi (2015), *Google Map* adalah peta yang diberikan oleh *Google* secara *free* dan online dan bisa dicari di internet dengan menggunakan URL <http://maps.google.com>. Layanan *Google Map* yaitu menampilkan jalan-jalan yang ada di dunia, yang berbentuk peta pada umumnya, *google Map* hanya menampilkan jalan, dan tidak ada nama gunung, sungai, maupun batas setiap dae-



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.

rah seperti pada peta umum. Tetapi, peta yang di *Google Map* sangat lengkap. Misalnya di kota-kota besar, *Google Map* menampilkan nama-nama jalan dan gedung. Dan tidak hanya itu, *Google Map* juga bisa menampilkan peta dengan tampilan foto satelit. beberapa fasilitas dari *Google Map* yaitu: Legenda, zoom in/ out, pan, dan bisa mengambil koordinat dari sebuah lokasi.

Google maps dapat diintegrasikan ke *Web* ataupun aplikasi, yaitu dengan cara menggunakan *Google maps API*. *Google Maps API* adalah aplikasi antar muka yang diakses melalui *java script* agar *Google Map* dapat diintegrasikan ke halaman web. *Google Maps Java Script API* versi dua, *Google* membuat *API Key* ketika ingin mengakses *Google Map* pada sebuah halaman web. *API Key* merupakan kode untuk mendapatkan izin menampilkan *Google Map* pada halaman web. Dalam versi yang ketiga *Google Maps API* tidak harus menggunakan *API Key*, tetapi pihak *google* menyarankan untuk tetap memakai *API Key* gunanya untuk mempermudah dalam mengontrol *Google Maps API*. *API Key* hanya bisa digunakan satu URL saja. Adapun syarat untuk bisa mendapatkan *API Key* adalah harus memiliki akun *google/ Gmail* untuk men-generate domain atau URL web pada link <https://code.google.com/apis/console/>. *API Key* bisa digunakan secara gratis tetapi memiliki batas pengunjung per hari yaitu 25.000. Jika melebihi 25.000 pengunjung per hari maka diperlukan biaya tambahan.

2.10 KWH Meter Listrik

Menurut Prasetyo dkk. (2016) Kwh-Meter listrik adalah suatu alat pencatat pemakaian daya listrik. Jumlah pemakaian daya listrik dapat dilihat dari angka yang tertera pada Kwh-Meter dengan perantara kepingan aluminium yang berputar.

Prinsip kerja KWH meter adalah Dalam alat ukur energi, kumparan-kumparan arus dan tegangan merupakan suatu belitan pada dua buah magnet. Kumparan arus akan membangkitkan flux magnet dengan nilai berbanding lurus dengan besar arus. Terjadinya perputaran dari piringan aluminium karena interaksi dari kedua medan magnet ini. Kemudian putaran piringan di transfer pada roda-roda pencatat. Pada transfer mati nilai putaran keping Aluminium ke roda-roda pencatat dilakukan kalibrasi untuk memperoleh nilai energi terukur dalam besaran Kwh (*Kilowatt hours*).

2.11 Blackbox Testing

Menurut Mustaqbal, Firdaus, dan Rahmadi (2016) *Blackbox Testing* berfungsi untuk mengathui fungsi dari sebuah *software*. penguji dapat mengartikan kumpulan kondisi inputan dan mngetes pada spesifikasi fungsi sistem. *Blackbox Testing* bukan alternatif dari *White Box Testing* tetapi pelengkap untuk menguji



Hak Cipta Dilindungi Undang-Undang

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

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

hal-hal yang tidak dicakup oleh *White Box Testing*. Ada beberapa teknik untuk melakukan *blackbox testing* di antaranya: *requirement testing*, *performance testing*, dan *behavior testing*. *Blackbox Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan *interface*.
3. Kesalahan pada struktur data dan akses basis data.
4. (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Testing bisa dilakukan dengan memberikan pertanyaan-pertanyaan tentang fungsional sistem melalui kuesioner dan perhitungannya dilakukan dengan Persamaan 2.1.

$$PresentasiBerhasil = \frac{JawabanBerhasil}{JumlahPertanyaan} \times 100\% \quad (2.1)$$

2.12 User Acceptance Test (UAT)

”Menurut Simarmata (2010), di *UAT testing software* langsung diberikan kepada *user* untuk mengetahui apakah software sudah sesuai dengan apa yang diinginkan pengguna dan berjalan sesuai yang diharapkan.

Sedangkan menurut Mutiara, Awaludin, Muslim, dan Oswari (2014) *user acceptance test* adalah kegiatan pengujian langsung oleh pengguna serta memberikan hasil sebuah dokumen untuk digunakan sebagai bukti apakah aplikasi yang dibuat dapat diterima pengguna dan hasil *testing* dirasa kriteria kebutuhan *user*. *User Acceptance Test* beris tentang pertanyaan yang ada hubungan dengan sistem yang akan dilakukan pengetesan. Perhitungan *User Acceptance Test* menggunakan formula yang dapat dilihat pada Persamaan 2.2.

$$PresentasiBerhasil = \frac{JawabanTidakSetuju}{TotalPertanyaan} \times 100\% \quad (2.2)$$

2.13 Riset Terdahulu

Riset terdahulu dapat dilihat pada Tabel 2.4.

Tabel 2.4. Riset terdahulu

No	Judul	Oleh	Hasil
1	Digital <i>Monitoring</i> Proyek Pergantian Kwh Meter Dan Segel Pada PT. Andika Energindo	Dewi dkk. (2015)	Hasil penelitiannya yaitu sistem untuk membuat surat penunjukan tugas ke teknisi dan meginput hasil kerja teknisi dalam melakukan penggantian KWH.
2	Perancangan Aplikasi Komputer Berbasis <i>Android</i> untuk Panduan Pengawasan pembangunan Kapal Baru oleh Owner Surveyor	Lasuardi dan Pribadi (2016)	menghasilkan sistem pengawasan pembangunan kapal baru yang dapat diimplementasikan dalam bentuk aplikasi komputer berbasis <i>android</i> .
3	Rancang Bangun Sistem Informasi Montoring Perkembangan Proyek Berbasis WEB (Studi Kasus: PT. Inti Pratama Semesta	Aprisa dan Monalisa (2015)	Dengan adanya sistem informasi <i>monitoring</i> perkembangan proyek maka dapat membantu admin dalam proses pengolahan data, mempermudah manager alam melakukan pemantauan perkembangan proyek setiap harinya tanpa harus datang ke lokasi pengerjaan proyek, dan mempermudah pimpinan dalam mengambil keputusan terhadap <i>monitoring</i> proses pengolahan data proyek.
4	Sistem Pelaporan Dan <i>Monitoring</i> Kegiatan Pusat Informasi Dan Konseling Remaja (Pik-R) (Studi Kasus: Dinas Bp2kb Kota Pariaman)	Primazni (2016)	Sistem Pelaporan Dan <i>Monitoring</i> Kegiatan Pusat Informasi Dan Konseling Remaja (PIK-R) Berbasis Web, yang nantinya sistem ini bisa meningkatkan keefisienan waktu pengumpulan, penyimpanan dan penataan data maupun informasi dengan baik, dan dapat <i>memonitoring</i> kegiatan yang telah ditetapkan batas waktunya, apakah seluruh kegiatan selesai tepat waktu atau tidak sehingga sistem akan memberikan peringatan apabila waktu akan berakhir.

Hak Cipta Diindungi 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.

State Islamic University of Sultan Syarif Kasim Riau