

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 mengikuti 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.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Sistem adalah suatu hal yang terpenting dalam membuat perancangan sistem informasi. Sistem informasi dapat terbentuk karena didorong oleh kebutuhan akan informasi yang terus meningkat yang dibutuhkan oleh pengambil keputusan.

2.1.1 Defenisi Sistem

Terdapat dua kelompok pendekatan dalam mendefenisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya (Jogiyanto, 2005).

Pendekatan sistem yang menekankan pada prosedurya mendefinisikan sistem sebagai berikut: "Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu".

Pendekatan sistem yang menekankan pada kumponen atau elemennya mendefinisikan sistem sebagai berikut "Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu".

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, dalam bukunya yang berjudul analisis dan desain sistem informasi menjelaskan karakteristik sistem antara lain sebagai berikut (Jogiyanto, 2005):

1) Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berintegrasi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.



2) Batasan Sistem

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3) Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4) Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5) Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem diproses dan akhirnya dikeluarkan berupa informasi yang dibutuhkan.

6) Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi informasi yang berguna.

7) Pengolahan Sistem

Pengolahan sistem merupakan suatu bagian yang mengolah masukan (*input*) dan memprosesnya agar menjadi *output* informasi yang berguna.

8) Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Jika suatu tidak mempunyai sasaran maka operasi sistem tidak akan berguna. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem

dan keluaran yang dihasilkan sistem. Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujannya.

2.1.3 Klasifikasi Sistem

Sistem juga dapat diklasifikasikan dari beberapa sudut pandang, klasifikasi sistem diantaranya adalah sebagai berikut (Jogiyanto, 2005):

1) Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik

2) Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia, sistem ini melibatkan interaksi antara manusia dengan mesin.

3) Sistem Tertentu dan Sistem Tak Tentu

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi antara bagian-bagiannya dapat dideteksi dengan pasti. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4) Sistem Tertutup dan Sistem Terbuka

Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.2 Konsep Dasar Informasi

Informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan, maka informasi menjadi tidak diperlukan. Sumber informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata dan merupakan suatu bentuk yang masih mentah sehingga perlu diolah lebih lanjut untuk menghasilkan informasi. (Tata Sutabri, 2005).

2.2.1 Defenisi Informasi

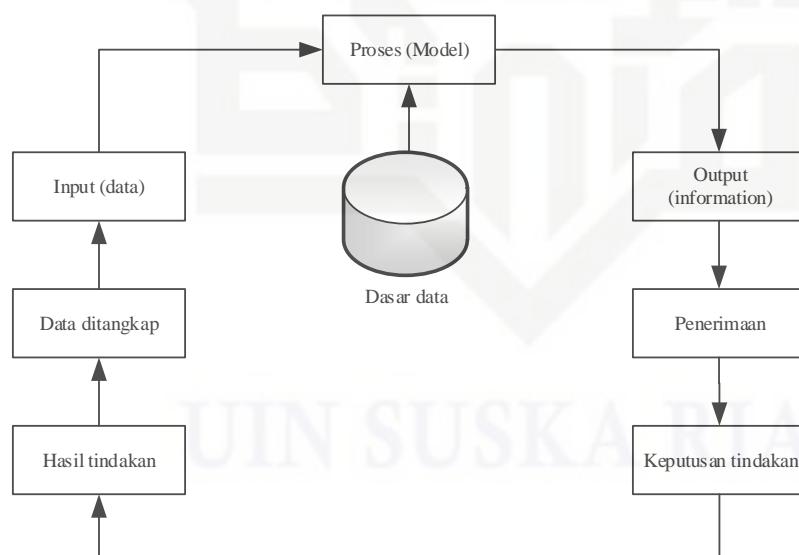
Informasi adalah data yang telah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 2005).

2.2.2 Nilai Informasi

Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya (Jogiyanto, 2005).

2.2.3 Siklus Hidup Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.



Gambar 2.1 Siklus Hidup Informasi

(Sumber: Jogiyanto, 2005)

2.2.4 Kualitas Informasi

Baik buruknya kualitas suatu informasi dipengaruhi oleh beberapa hal, antara lain (Jogiyanto, 2005):

- 1) Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
- 2) Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi harus sesuai dengan dengan keadaan saat itu, karena informasi yang sudah usang tidak akan mempunyai nilai lagi. Informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya informasi disebabkan harus cepatnya informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.
- 3) Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan lainnya berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi relevan untuk akuntan.

2.3 Sistem Informasi

Sistem informasi didefinisikan oleh Robert A.Leitch dan K.Roscoe Davis adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi. bersifat material dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang digunakan (Jogiyanto, 2005).

2.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yang terdiri dari beberapa blok adalah sebagai berikut (Tata Sutabri, 2005).

1) Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi.

2) Blok Model

Terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang disimpan di basis data.

3) Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas.

4) Blok Teknologi

Teknologi merupakan tool box dalam sistem informasi.

5) Blok Basis Data

Merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras dan perangkat lunak untuk memanipulasinya.

6) Blok Kendali

Pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau terlanjur terjadi kesalahan dapat langsung diatasi.

2.4 Metode Analisis dan Perancangan Sistem

2.4.1 Analisis Sistem

Analisis sistem (*Sistem Analisys*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Jogiyanto, 2005).

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 mengikuti 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.

Didalam tahap analis sistem terdapat langkah-langkah dasar yang harus dilakukan sebagai berikut (Jogiyanto, 2005):

- 1) *Identify*, yaitu mengidentifikasi masalah, mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analis sistem, tugas-tugas yang harus dilakukan seperti:
 - a. Mengidentifikasi penyebab masalah, tugas mengidentifikasi penyebab masalah dapat dimulai dengan mengkaji ulang terlebih dahulu subyek-subyek permasalahan yang telah diuraikan oleh manajemen atau yang telah ditemukan oleh analis sistem ditahap perencanaan sistem.
 - b. Mengidentifikasi titik keputusan, setelah penyebab terjadinya masalah dapat diidentifikasi, seianjutnya juga harus diidentifikasi titik keputusan penyebab masalah tersebut.
 - c. Mengidentifikasi personil-personil kunci. identifikasi personil-personil kunci ini dapat dilakukan dengan mengacu pada bagan alir dokumen yang ada diperusahaan serta dokumen deskripsi jabatan.
- 2) *Understand*, yaitu memahami kerja dari sistem yang ada, langkah kedua dari tahap analis sistem adalah memahami dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.
- 3) *Analyze*, yaitu menganalisis hasil penelitian.
 - a. Menganalisis kelemahan sistem, analis sistem perlu menganalisis masalah yang terjadi untuk dapat menemukan jawaban apa penyebab sebenarnya dari masalah yang timbul tersebut.
 - b. Menganalisis kebutuhan informasi pemakai/manajemen,
- 4) *Report*, yaitu membuat laporan hasil analisis, laporan hasil analis ini diserahkan kepada *steering committmen* yang nantinya akan diteruskan kemanajemen. Tujuan utama dari penyerahan laporan ini kepada manajemen adalah:
 - a. Laporan bahwa analisis telah selesai dilakukan.

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 mengikuti 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.

- b. Meluruskan kesalahan pengertian mengenai apa yang telah ditemukan dan dianalisis oleh analis sistem tctapi tidak sesuai menurut manajemen.
- c. Meminta pendapat-pendapat saran-saran dari pihak manajemen.
- d. Meminta persetujuan dari pihak manajemen untuk melakukan tindakan selanjutnya.

2.4.2 Perancangan Sistem

Perancangan adalah kemampuan untuk membuat beberapa alternatif pemecahan masalah (Bin Ladjamudin, 2005). Perancangan sistem adalah tahap yang dilakukan setelah melakukan analisis sistem, pendefinisian kebutuhan-kebutuhan sistem yang akan dibangun, dan persiapan untuk merancang bangun implementasi sistem dengan menggambarkan sistem yang akan dibangun. Perancangan sistem dimulai dengan memahami sistem yang sedang berjalan dan kriteria-kriteria sistem yang akan dibangun biasanya menggunakan pemodelan secara terstruktur yang digambarkan oleh grafik atau diagram (Jogiyanto, 2005).

Hal-hal yang harus diperhatikan dalam merancang dan membangun sebuah sistem (Jogiyanto, 2005):

- 1) Kebutuhan perusahaan, organisasi, atau lembaga, yaitu dengan cara memahami bidang yang akan dikembangkan, sasaran yang dibidik serta media yang akan digunakan.
- 2) Kebutuhan operator, yaitu kebutuhan operator untuk memperoleh sistem yang mudah dipahami dan dioperasikan serta tampilan yang interaktif.
- 3) Kebutuhan pemakai, yaitu keinginan-keinginan dari si pemakai sistem informasi, seperti jaminan keamanan, standarsasi tampilan, kecepatan akses, dan kemudahan dalam pengoperasian.
- 4) Kebutuhan teknis, yaitu meliputi arsitektur dan Konfigurasi sistem. Secara teknis peralatan dan teknologi yang digunakan, termasuk pertimbangan penggunaan peralatan yang tidak standar, seperti peralatan multimedia, kebutuhan *interface*, *database*, dan perangkat lunaknya.

2.5 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan paradigma pemrograman yang berorientasi kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek (Sutopo, 2002 dikutip oleh kasini, 2015)

Pada objek oriented terdapat beberapa model pendekatan, yaitu *Object oriented programing* (OOP) dan *object oriented analysis and design* (OOAD). *Object oriented programing* (OOP) atau pemrograman berorientasi objek adalah konsep pemrograman yang difokuskan pada penciptaan kelas yang merupakan abstraks/blueprint/prototype dari suatu objek. Kelas ini harus mengandung sifat (data) dan tingkah laku (method) umum yang dimiliki oleh objek-objek yang kelak akan dibuat (diinstansiasi). Data dan method merupakan anggota dari suatu kelas. Sedangkan *object oriented analysis and design* (OOAD) adalah metode analisis yang memeriksa *requirement* dari sudut pandang kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem. OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Dari beberapa model *Object Oriented* diatas, maka dalam penelitian ini, saya akan menggunakan model *object oriented analysis and design* (OOAD).

2.5.1 Konsep OOAD

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD). OOA adalah metode analisis yang memeriksa *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem (Sutopo, 2002 dikutip oleh kasini, 2015).

Terdapat beberapa konsep dalam OOAD, yaitu (Sutopo, 2002 dikutip oleh kasini, 2015)



- 1) *Object* (objek)
 - a. *Object* adalah benda secara fisik dan konseptual yang ada di sekitar kita sebuah objek memiliki keadaan sesaat yang disebut *state*.
 - b. *State* dari sebuah objek adalah kondisi dari objek atau himpunan keadaan yang menggambarkan objek tersebut. *State* dinyatakan dengan nilai dari atribut objeknya.
 - c. Atribut adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kandisi sesaat, koneksi dengan objek lain dan identitas.
 - d. *Behaviour* (perilaku objek) mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. *Behaviour* ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek tersebut, yang dicerminkan oleh *interface*, *service*, dan *method* dari objek tersebut.
 - e. *Interface* adalah pintu untuk mengakses *service* dari objek
 - f. *Service* adalah fungsi yang dapat dikerjakan oleh sebuah objek
 - g. *Method* adalah mekanisme *internal* objek yang mencerminkan perilaku objek tersebut.
- 2) *Class* (kelas)

Class adalah himpunan objek yang sejenis yaitu mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. *Class* adalah abstraksi dari objek dalam dunia nyata. *Class* menetapkan spesifikasi perilaku dan atribut dari objek tersebut.

- 3) *Blackbox* (kotak hitam)

Sebuah objek adalah kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi *object oriented* hanya *developer* yang dapat memahami detail proses yang ada didalam kotak tersebut, sedangkan user tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.



- a. *Encapsulation*, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui *interface*. Untuk berkomunikasi dengan objek digunakan *message*.
 - b. *Message* adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan *result* dari aksi tersebut kepada objek pengirim (*sender*)
- 4) Asosiasi dan *Agregasi*
- a. Asosiasi adalah hubungan yang mempunyai makna antara sejumlah objek. Asosiasi digambarkan dengan sebuah garis penghubung diantara objeknya. contohnya: Asosiasi karyawan dengan unit kerja. Setiap karyawan bekerja di satu unit kerja, sedangkan unit kerja dapat memiliki beberapa karyawan.
 - b. Agregasi adalah bentuk khusus sebuah asosiasi yang menggambarkan seluruh bagian pada suatu objek merupakan bagian dari objek yang lain. Contohnya: Kopling dan piston adalah bagian dari mesin, sedangkan mesin, roda, *body* merupakan bagian dari sebuah mobil.

2.5.2 Teknik Pemodelan Dalam OOAD

Berikut adaiah pemodelan dalam *object oriented analysis design* (Sutopo, 2002 dikutip oleh kasini, 2015)

Model Objek:

- 1) Model objek menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya
- 2) Model objek berisi diagram objek. Diagram objek adalah graph dimana nodenya adalah kelas yang mempunyai relasi antar kelas.

Model Dinamik:

- 1) Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
- 2) Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.

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 mengikuti 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) Model dinamik berisi state diagram. State diagram adalah graph dimana nodenya adalah *state* dan *arc* adalah tarsnsisi antara state yang disebabkan oleh *event*.

Model Fungsional

- 1) Model fungsional menggambarkan transformasi nilai data di dalam sistem.
- 2) Model fungsional berisi data *flow* diagram. DFD adalah suatu *graph* dimana nodenya menyatakan proses dan arcnya adalah aliran data.

2.5.3 Karakteristik Objek

Berikut karakteristik dari objek (Sutopo, 2002 dikutip oleh kasini, 2015):

1) Objek

- a. Identitas berarti bahwa data diukur mempunyai nilai tertentu yang membedakan entitas disebut objek.
- b. Objek dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti kebijakan penjadualan dalam *multiprocessig* pada sistem operasi.
- c. Setiap objek mempunyai sifat yang melekat pada identitasnya.
- d. Dua objek dapat berbeda walaupun bila semua nilai atribut
- e. Atributnya identik



Mobil



Singa

NoPeg	Nama
96001	Susan
96002	David
97001	Shila

Tabel

Gambar 2.2 Macam-macam objek

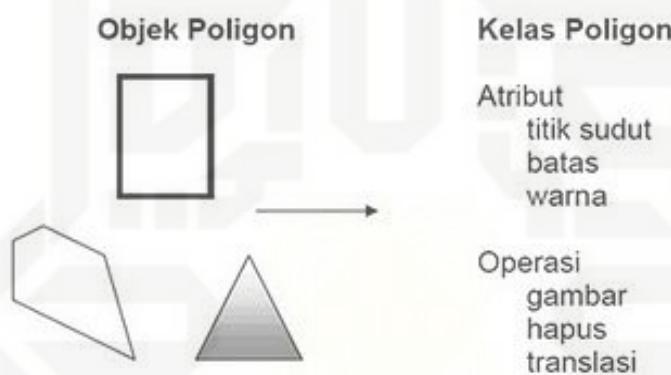
(Sumber: Sutopo, 2002)

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 mengikuti 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) Kelas Objek

- 1) Kelas merupakan gambaran sekumpulan objek yang terbagi dalam atribut, operasi, metode, hubungan, dan makna yang sama,
- 2) Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup.
- 3) Kelas objek merupakan wadah bagi objek. Dapat digunakan untuk menciptakan objek.
- 4) Objek mewakili fakta/keterangan dari sebuah kelas.



Gambar 2.3 Kelas dan Objek

(Sumber: Sutopo, 2002)

2.5.4 Karakteristik Metodologi Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama, yaitu (Nugroho, 2005 dikutip oleh kasini, 2015):

1) *Encapsulation (Pengkapsulan)*

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. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

2) *Inheritance (Pewarisan)*

Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung.

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 mengikuti 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.

Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya, karena telah mewarisi sifat induknya. Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secam umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya, kelas objek dapat didefinisikan atribut dan *service* dari kelas objek lainnya. *Inheritance* menggambarkan generalisasi sebuah kelas. Sifat yang dimiliknya oleh kelas induknya tidak perlu diulang dalam setiap subkelas. Sebagai contoh, Sedan dan Sepeda motor adalah subkelas dari kendaraan bermotor. Kedua subkelas mewarisi sifat yang dimiliki oleh kendaraan bermotor, yaitu:

- a. Mempunyai mesin
- b. Dapat berjalan

Kedua subkelas mempunyai sifat masing-masing yang berbeda, misalnya jumlah roda, dan kemampuan untuk berjalan mundur yang tidak dimiliki oleh sepeda motor. Beberapa faktor dari super kelas yang bersifat umum dan dimasukkan kedalam kelas induknya serta mewariskan sifat tersebut pada kelas turunannya, sehingga mengurangi pengulangan yang terjadi dalam desain dan pemrograman. Hal ini merupakan keuntungan dari sistem berorientasi objek.

3) ***Polymorphism (Perbedaan Bentuk)***

Polimorfisme yaitu konsep yang menyatakan bahwa suatu yang sama dapat mempunyai bentuk dan perilaku berbeda. *Polimorfisme* mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan objek.

2.6 UML

Pada perkembangan perangkat lunak, diperlukan bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau stuktural, kemudian juga ada *State Transition Diagram* (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak (Rosa dan Salahuddin, 2014).

2.6.1 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada *depelover* pengembang bahasa pemrograman berorientasi objek selanjutnya (Rosa dan Salahuddin, 2014).

2.6.2 Diagram-Diagram UML yang Digunakan

- 1) *Class Diagram*
- 2) *Use case Diagram*
- 3) *Activity Diagram*
- 4) *Sequence diagram*

Tabel 2.1 Tipe Diagram UML

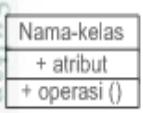
Diagram	Tujuan
<i>Class Diagram</i>	Memodelkan kosakata di sistem, distribusi dan tanggung jawab tipe primitive, kolaborasi, skema <i>database</i> logik
<i>Use case Diagram</i>	Menunjukkan sekupulan kasus fungsional dan aktor dan hubungannya.
<i>Activity Diagram</i>	Pandangan operasi, bagaimana objek-objek bekerja, aksi-aksi yang mempengaruhi objek, pandangan <i>use case workflow</i> .
<i>Sequence diagram</i>	Berfungsi untuk <i>overview</i> prilaku sistem, menunjukkan objek-objek yang diperlukan, mendokumentasikan skenario dari suatu diagram <i>use case</i> , memeriksa jalur-jalur pengakesan.

(Sumber: Rosa dan Salahuddin. 2014).

1. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Salahuddin, 2014). Dapat dilihat pada tabel 2.2

Tabel 2.2 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem

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 mengikuti 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 2.2: Simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
 Antarmuka/ / interface nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
 Asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan <i>multiplicity</i>
 Asosiasi berarah / directed association	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
 Generalisasi	Relasi antar kelas dengan makna generalisasi-spesifikasi (umum-khusus)
 Kebergantungan/dependency	Relasi antar kelas dengan makna ketergantungan antar kelas
 Aggregasi / aggregation	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber: Rosa dan Salahuddin, 2014).

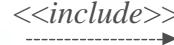
2. *Use Case Diagram*

Use case adalah pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa dan Salahuddin, 2014).

Tabel 2.3 Simbol *Use Case Diagram*

Simbol	Deskripsi
 Use case nama use case	Fungsionalisasi yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya di nyatakan dengan menggunakan kata kerja di awal <i>phrase name use case</i> .
Aktor / actor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di

Tabel 2.3: Simbol *Use Case Diagram* (Lanjutan)

Simbol	Deskripsi
 Nama aktor	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
<i>Asosiasi / association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<i>Ekstensi / 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 walaupun tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang di tambahkan misal arah panah mengarah pada <i>use case</i> yang di tambahkan
<i>Generalisasi</i> 	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> di tambahkan memerlukan <i>use case</i> ini menjalankan fungsinya atau syarat di jalankan <i>use case</i> ini

(Sumber: Rosa dan Salahuddin, 2014).

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 mengikuti 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. Activity Diagram

Activity diagram menggambarkan berbagai aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem (Rosa dan Salahuddin, 2014). Diagram aktivitas juga banyak di gunakan untuk mendefenisikan hal-hal sebagai berikut:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang di gambarkan merupakan proses bisnis sistem yang di defenisikan
- Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana setiap aktivitas di anggap memiliki sebuah rancangan antarmuka tampilan
- Rancangan pengujian dimana setiap aktivitas di anggap memerlukan sebuah pengujian yang di perlukan didefinisikan kasus ujiannya
- Rancangan menu yang ditampilkan perangkat lunak

Tabel 2.4 Simbol - Simbol *Activity Diagram*

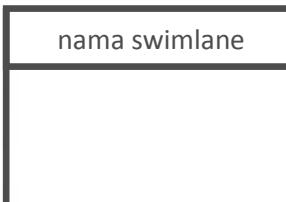
Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memilih sebuah status awal.
Aktivitas	Aktivitas yang di lakukan sistem aktivitas biasanya di awali dengan kata kerja
Percabangan / <i>decision</i>	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari
Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih satu aktivitas di gabung menjadi satu
Status akhir	Status akhir yang di lakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir

- Hak Cipta Dilindungi Undang-Undang**
- 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.
 - Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

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 mengikuti 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 2.4 Simbol - Simbol *Activity Diagram* (Lanjutan)

Simbol	Deskripsi
 Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Sumber: Rosa dan Salahuddin, 2014).

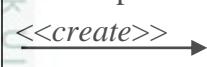
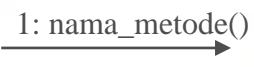
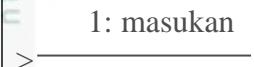
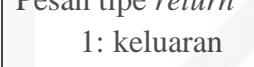
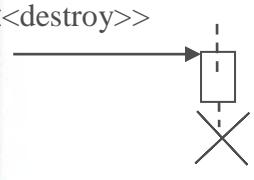
4. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek (Rosa dan Salahuddin, 2014).

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

Simbol	Deskripsi
 Aktor Nama aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orangbiasanya di nyatakan menggunakan kata benda di awal <i>phrase</i> nama aktor
 Garis hidup / <i>lifeline</i>	Menyatakan hidup suatu objek
 Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan

Tabel 2.5 Simbol-Simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
 Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
 Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
 Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri
 Pesan tipe <i>send</i>	Menyatakan suatu objek mengirim data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
 Pesan tipe <i>return</i>	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
 Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber: Rosa dan Salahuddin, 2014)

2.7 Konsep Dasar Pemrograman PHP

PHP singkatan dari PHP *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan web yang disisipkan pada dokumen HTML.

2.7.1 Sejarah Singkat PHP

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa

saja pengunjung pada *homepage*-nya. Rasmus Lerdorf adalah salah seorang pendukung *open source*. Oleh karena itu, ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudian menambah kemampuan PHP 1.0 dan meluncurkan PHP 2.0

Pada tahun 1996, PHP telah banyak digunakan dalam website, di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Guttmann, Sblg Bakken, Shane Caravco, dan Jim Winstead bekerja sama untuk menyempurnakan PHP 2.0 akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0 tidak berhenti sampai disitu, kemampuan PHP terus ditambah, dan saat buku ini disusun, versi terbaru yang telah dikeluarkan adalah PHP 5.0.x (kasiman, 2006)

2.7.2 Kelebihan-Kelebihan PHP

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan halaman web dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih dari pada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain Linux, unix (termasuk variannya *HP-UX*, *Solaris*, dan *Open BSD*), *Microsoft Windows*, *Mac OS X*, *RISC OS*. PHP juga mendukung banyak *web server*, seperti *Apache*, *Microsoft Internet Information Server* (MIIS), *Personal Web Server* (PWS), *Netscape and Planet servers*, *Oreilly Website Pro server*, *audium*, *Xitami*, *OmniHTTPd*. dan masih banyak lagi lainnya, bahkan PHP dapat bekerja sebagai suatu *CGI processor*.

PHP tidak terbatas pada hasil keluaran HTML (*Hyper Text Mark Up Languages*). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file *PDF*, dan *movies Flash*. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya.

Salah satu fitur yang dapat diandalkan oleh PHP adalah lingkungannya yaitu hadap banyak *database*. (Kasiman, 2006)

1. *Adabas D*
2. *Dbase*
3. Direct MS-SQL
4. *Empress*
5. *Interbase*
6. *MSQL*
7. *MySQL*
8. *Oracale*
9. *Unix DBD*
10. *Velocis*

2.8 MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL memiliki dua bentuk lisensi, yaitu *freeSoftware* dan *Shareware* (Bunafit Nugroho, 2005).

2.8.1 Kelebihan MySQL

MySQL memiliki beberapa kelebihan dibandingkan database lain diantaranya:

1. *MySQL* sebagai *database management sistem* (DBMS)
2. *MySQL* sebagai *relation database management Sistem* (RDBMS)
3. *MySQL* sebuah software database yang opensource bersifat *free* bebas digunakan oleh siapa saja
4. *MySQL* merupakan sebuah *database server*
5. *MySQL* merupakan sebuah *database client*
6. *MySQL* mampu menerima *query* yang bertumpuk dalam satu permintaan
7. *MySQL* sebuah database yang mampu menyimpan data berkapasitas sangat besar
8. *MySQL* didukung oleh *driver ODBC MySQL* dapat diakses menggunakan aplikasi apa saja.

9. *MySQL database* menggunakan enkripsi *password*.
10. *MySQL* server database yang multi *user*
11. *MySQL* dapat menciptakan lebih dari 16 kunci portabel
12. *MySQL* mendukung *field* yang dijadikan sebagai kunci primer dan kunci *uniq*
13. *MySQL* didukung oleh sebuah *component C* dan perl API

2.9 Model Proses Pengembangan Perangkat Lunak

2.9.1 Waterfall

Metodologi pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan yang akan digunakan untuk mengembangkan suatu sistem informasi (Jogiyanto, 2005 dikutip oleh kasini, 2015).

Metodologi yang penulis gunakan adalah metodologi *waterfall*, Metode air terjun atau *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara skuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut adalah tahapan dari model air terjun:

1) Survei sistem

Pada tahap ini akan dilakukan beberapa kegiatan yang meliputi pendefinisian dari permasalahan yang ada untuk menentukan ruang lingkup, menentukan metodologi yang dipergunakan, serta membuat jadwal kegiatan dengan menggunakan beberapa teknik pengumpulan data seperti wawancara dapat dilihat pada lampiran-B dan observasi dapat dilihat pada lampiran-A.

2) Analisa sistem

Pada tahap ini dilakukan analisis terhadap sistem yang telah ada dengan mengidentifikasi permasalahan, penentuan tujuan dari perbaikan sebuah sistem, dan mengidentifikasi kebutuhan pengguna sistem.

3) Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari

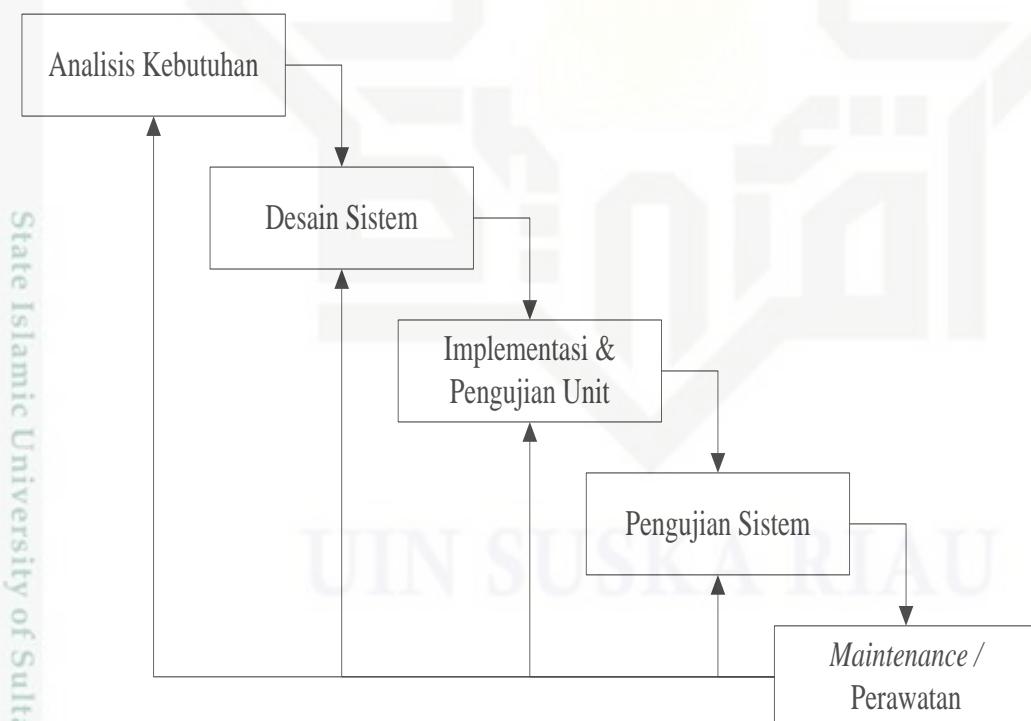
tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap *desain*.

4) Pengujian

Pengujian fokus pada perangkat lunak secara segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5) Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi untuk tidak untuk membuat perangkat lunak.



Gambar 2.4 Metodologi Waterfall

(Sumber: Jogiyanto, 2005)