

BAB II

LANDASAN TEORI

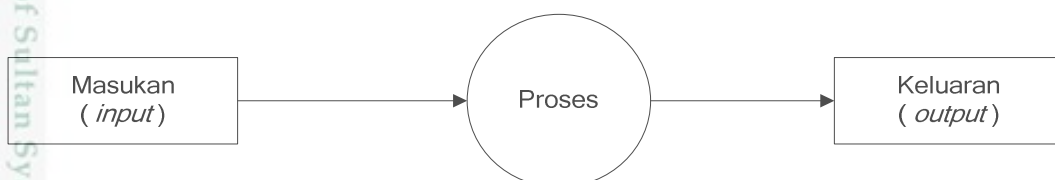
2.1 Rancang Bangun

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2003). Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

2.2 Pengertian Sistem

Istilah sistem secara umum dapat didefinisikan sebagai kumpulan hal atau elemen yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan. Sistem mempunyai karakteristik atau sifat – sifat tertentu, yaitu: Komponen Sistem, Batasan Sistem, Lingkungan Luar Sistem, Penghubung Sistem, Masukan Sistem, Keluaran Sistem, Pengolahan Sistem dan Sasaran Sistem.

Menurut Jogiyanto (2005) Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Proses *input dan output* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Proses I/O

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.

Sistem mempunyai 8 karakteristik sebagai berikut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.

2. Batas Sistem

Batas sistem (*Boundry*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau lingkungan luarnya.

3. Lingkungan luar Sistem

Lingkungan luar (*Environments*) dari suatu sistem adalah apapun diluar dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem

Penghubung (*Interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

5. Masukan Sistem

Masukan (*Input*) adalah suatu energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*Maintenance Input*) dan masukan signal (*Signal Input*). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal Input adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem

Keluaran (*Output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

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.

2.3 Konsep Dasar Informasi

Menurut Kadir (2003) Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya dalam pengambilan keputusan saat ini atau saat mendatang.

Informasi dapat diperoleh dari sistem informasi (*information system*) atau disebut juga dengan proses sistem (*information processing system*) atau *Informating generation system*. Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diberikan.

Komponen- komponen Sistem Informasi ada 5, yaitu:

1. Blok Masukan, yaitu berupa input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, berupa dokumen dasar.
2. Blok Model, blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di data base dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok Keluaran, produk sistem informasi adalah keluaran yang merupakan sistem informasi berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem
4. Blok Teknologi, teknologi merupakan alat atau *tool-box* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara keseluruhan.
5. Blok Kendali, beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang merusak seperti bencana alam, api, debu, virus, serta kecurangan dapat dicegah dan langsung diatasi.

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.

2.4 PHP (*Hypertext Processor*)

2.4.1 Pengertian PHP

PHP merupakan bahasa server-side yang cukup handal, yang akan disatukan dengan HTML dan berada di server. Artinya, sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server sebelum dikirim ke komputer klien. Pada awal tahun 1995, Rasmus Lerdorf membuat produk bernama PHP/FI (*Personal Home Page/Form Interpreter*). Produk yang merupakan cikal bakal PHP ini ditulis menggunakan bahasa C, dan memiliki kemampuan untuk berkomunikasi dengan database serta membuat halaman dinamis (Sidiq , 2012).

Seluruh aplikasi yang berbasis web dapat dibuat menggunakan PHP. Salah satu kelebihan PHP adalah kemampuan untuk dapat melakukan koneksi dengan berbagai database, seperti MySQL, PostgreSQL, dan *Access*. Selain itu PHP juga bersifat open source, untuk dapat menggunakannya kita tidak perlu membayar. Variabel PHP digunakan untuk menyimpan data yang nilainya dapat berubah-ubah. Dalam bahasa PHP, variabel dimulai dengan tanda "\$".

PHP *Hypertext Preprocessor* adalah skrip yang berjalan dalam server side yang ditambahkan dalam HTML. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman HTML tidak lagi bersifat statis, namun menjadi dinamis. Sifat server side ini membuat pengerjaan skrip tersebut dikerjakan di server sedangkan yang dikirimkan kepada browser adalah hasil proses dari skrip tersebut yang sudah berbentuk HTML.

2.4.2 Sintaks PHP

Sintaks Program/Script ditulis dalam apitan tanda khusus PHP. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok script PHP:

1. `<?php.....?>`
2. `<script
language="PHP">.....</script>`
3. `<?.....?>`
4. `<%.....%>`.

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.

2.5 MySQL

MySQL dikembangkan oleh sebuah perusahaan *Swedia* bernama *MySQL AB* yang pada saat itu bernama *TcX Data Konsult AB* sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak 1979. Awalnya *TcX* membuat *MySQL* dengan tujuan mengembangkan aplikasi web untuk klien. *TcX* merupakan perusahaan pengembang *software* dan konsultan database. Saat ini *MySQL* sudah diakuisisi oleh *Oracle Crop*. *MySQL* adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang database sebagai sumber dan pengelolaan datanya. Kepopuleran *MySQL* antara lain karena *MySQL* menggunakan SQL sebagai bahasa dasar untuk mengakses database-nya sehingga mudah untuk digunakan. *MySQL* juga bersifat *open source* dan *free* pada berbagai platform kecuali pada *windows* yang bersifat *shareware*. *MySQL* didistribusikan dengan lisensi *open source GPL (General Public License)*.

Dalam perintah *MySQL* terdapat beberapa perintah, sebagai berikut:

1. Membuat Tabel

Tabel adalah objek utama yang harus ada pada sebuah data base, karena di dalam tabel semua data akan disimpan. Posisi tabel terletak pada sebuah data base, sehingga pembuatan tabel merupakan sebuah urutan setelah membuat database. Untuk membuat tabel, perintah yang digunakan adalah:
`CREATE TABLE nama_tabel (table_Field);`

2. Menghapus Tabel

Menghapus tabel sama dengan melakukan penghapusan database. Untuk menghapus tabel, perintah yang digunakan adalah:
`DROP TABLE nama_tabel;`

3. Menambah Kolom

Untuk menambah sebuah kolom kedalam tabel maka perintah yang digunakan adalah:
`ALTER TABLE nama_tabel;`
`ADD kolom_baru type (panjang);`

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.

4. Menghapus Kolom

Untuk menghapus sebuah kolom kedalam tabel maka perintah yang digunakan adalah:

```
ALTER TABLE nama_tabel;
```

```
DROP nama_kolom;
```

5. Memasukkan Data Tunggal

Memasukkan data tunggal merupakan kemampuan *MySQL* untuk dapat melakukan input data secara satu per satu. Perintah yang digunakan adalah :

```
INSERT INTO nama_tabel kolom_a,kolom_b,.....,kolom_n)
```

```
VALUES ('isi kolom_a','isi kolom_b'.....','isi kolom_n)
```

6. Memasukkan Data Secara Bersamaan

Untuk mempercepat proses pemasukkan data pada tabel,maka *MySQL* mempunyai kemampuan untuk dapat melakukan pemasukkan data secara bersamaan. Perintah yang digunakan adalah:

```
INSERT INTO nama_tabel kolom_a,kolom_b,.....,kolom_n)
```

```
VALUES
```

```
('isi kolom_a','isi kolom_b'.....','isi kolom_n)
```

```
('isi kolom_a','isi kolom_b'.....','isi kolom_n)
```

```
('isi kolom_a','isi kolom_b'.....','isi kolom_n)
```

7. Menampilkan Data

Perintah yang digunakan untuk menampilkan data adalah *SELECT*, bentuk umumnya adalah :

- a. Menampilkan semua kolom

```
SELECT*FROM nama_tabel;
```

- b. Menampilkan beberapa kolom

```
SELECT*FROM nama_tabel='nama_field yang ingin di tampilkan
```

- c. Menampilkan semua kolom

```
SELECT*FROM nama_tabel;
```


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.

b. *Inheritance* (Pewarisan)

Inheritance adalah mekanisme untuk mengekspresikan keserupaan di antara kelas-kelas, menyederhanakan pendefinisian kelas yang serupa dengan kelas yang telah didefinisikan sebelumnya. Pewarisan menyangkut generalisasi dan spesialisasi, menyatakan atribut dan layanan bersama (*commonality*) secara eksplisit di hirarki kelas.

c. *Polymorphism* (Polimorfisme)

Polymorphism merupakan konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku yang berbeda. Pada *polymorphism* memungkinkan kita menganggap sekelompok objek dengan tipe-tipe berbeda seperti dimiliki oleh suatu tipe yang sama. Pemakai objek-objek itu tidak perlu mengetahui mengenai perbedaan-perbedaan itu. Ketidakpedulian perbedaan-perbedaan ini membantu kita mengelola kompleksitas. Kompleksitas disebabkan adanya perbedaan-perbedaan pada kelompok objek itu menjadi tidak relevan untuk diperhatikan atau dipedulikan.

2.8 UML (*Unified Modelling Language*)

Menurut Sholiq (2006), Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbaugh, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Jacobson menulis tentang pendefinisian persyaratan-persyaratan sistem yang disebut *use case*. Juga mengembangkan sebuah metode untuk perancangan sistem yang disebut *Object Oriented Software Engineering (OOSE)* yang berfokus pada analisis. Booch, Rumbaugh dan Jacobson biasa disebut dengan tiga sekawan (*tree amigos*). Semuanya bekerja di *Rational Software Corporation* dan berfokus pada standarisasi dan perbaikan ulang UML. Simbol UML mirip dengan Booch, notasi *OMT*, dan juga ada kemiripan dengan notasi lainnya.

Penggabungan beberapa metode menjadi UML dimulai 1993. Pada akhir tahun 1995 *Unified Method* versi 0.8 diperkenalkan. *Unified Method* diperbaiki dan diubah menjadi UML pada tahun 1996, UML 1.0 disahkan dan diberikan

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.

pada *Object Technology Group (OTG)* pada tahun 1997, dan pada tahun itu juga beberapa perusahaan pengembang utama perangkat lunak mulai mengadopsinya. *UML* menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada delapan diagram yang disediakan dalam *UML* antara lain:

1. Diagram *use case (use case diagram)*.
2. Diagram aktivitas (*activity diagram*).
3. Diagram sekuensial (*sequence diagram*).
4. Diagram kolaborasi (*collaboration diagram*).
5. Diagram kelas (*class diagram*).
6. Diagram statechart (*statechart diagram*).
7. Diagram komponen (*component diagram*).
8. Diagram deployment (*deployment diagram*).

Adapun tujuan utama perancangan *UML* yaitu:

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

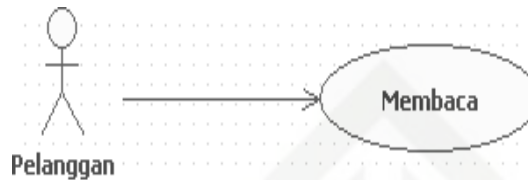
2.9 Diagram UML yang di gunakan

2.9.1 Use Case Diagram

Use case Diagram digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. *Use case Diagram* menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Pemodelan ini biasa dilakukan lewat proses berulang interaksi antara pengembang dan pemakai untuk memperoleh spesifikasi kebutuhan yang sama-sama disepakati.

Menurut Nugroho (2005) Diagram *use case* berguna dalam tiga hal yaitu:

- a. Menjelaskan fasilitas yang ada (*requirements*).
- b. Komunikasi dengan klien.
- c. Membuat test dari kasus-kasus secara umum.



Gambar 2.2 Use Case Diagram

3 Class Diagram

Class Diagram yaitu sebuah spesifikasi yang jika diinstalasi menghasilkan obyek dan inti dari pengembangan desain berorientasi objek. *Class diagram* menggambarkan properti dan metode atau fungsi. *Class* memiliki tiga area yaitu:

1. Nama dan *stereotype*.
2. *Atribut*.
3. *Metoda*.

Atribut dan *metoda* dapat memiliki salah satu sifat berikut:

1. *Private*, tidak dapat di panggil dari luar *class*.
2. *Protected*, hanya dapat di panggil oleh *class* yang bersangkutan dan anak pewarisnya.
3. *Public*, dapat di panggil oleh siapa saja.








Hubungan antar *class* ada empat yaitu:

1. *Asosiasi*, yaitu hubungan *statis* antar *class*. Panah navigasi menunjukkan arah *query* antar *class*.
2. *Agregasi*, yaitu hubungan yang menyatakan bagian (“terdiri atas...”)
3. *Pewarisan*, yaitu hubungan *hirarki* antar *class*. Artinya sebuah *class* dapat diturunkan menjadi *class* lain dan mewarisi semua *atribut* serta *metode* dari *class* induknya.

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.

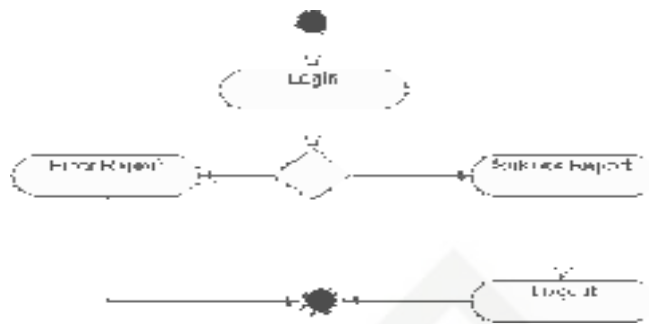
4 Hubungan dinamis, rangkaian pesan yang di-*passing* dari *class* ke *class* lain.
Hubungan dinamis digambarkan menggunakan *sequence diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Boundary Lifeline</i>	Menggambarkan hubungan suatu elemen yang berbeda, secara khas merupakan penghubung <i>actor</i> dengan layar.
2		<i>Entity Lifeline</i>	Menggambarkan suatu tempat atau mekanisme yang menangkap pengetahuan atau informasi dalam suatu sistem
3		<i>Control Lifeline</i>	Menggambarkan suatu pengendalian yang mengorganisir dan menjadwalkan aktivitas elemen-elemen.
4		<i>Message</i>	Perilaku sistem yang menandai adanya suatu alur informasi atau transisi kendali antar elemen.
5		<i>Actor</i>	Menunjukkan seorang pemakai sistem yang memulai alur peristiwa/kejadian.
6		<i>Activation bar</i>	Menggambarkan lamanya suatu pesan diproses.
7		<i>Note</i>	Menunjukkan catatan untuk komentar dari suatu pesan antar elemen.

Gambar 2.3 Simbol *Class Diagram*

4 *Activity Diagram*

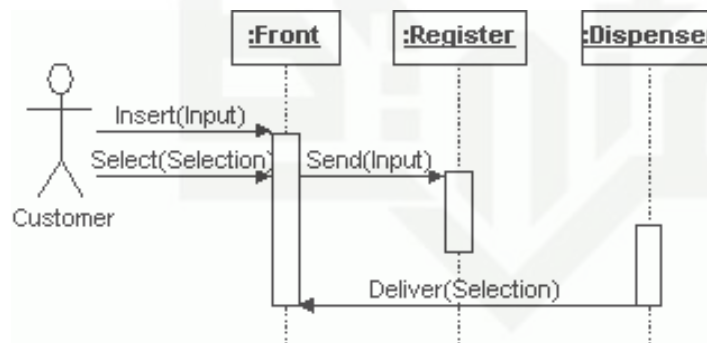
Pada dasarnya *Activity Diagram* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.



Gambar 2.4 Activity Diagram

5 Sequence Diagram

Sequence diagram merupakan salah satu diagram interaksi yang menjelaskan bagaimana suatu operasi itu dilakukan, pesan apa yang dikirim dan kapan pelaksanaannya. Diagram ini disusun berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2.5 Sequence Diagram

5.6 Model Pengembangan Sistem Waterfall

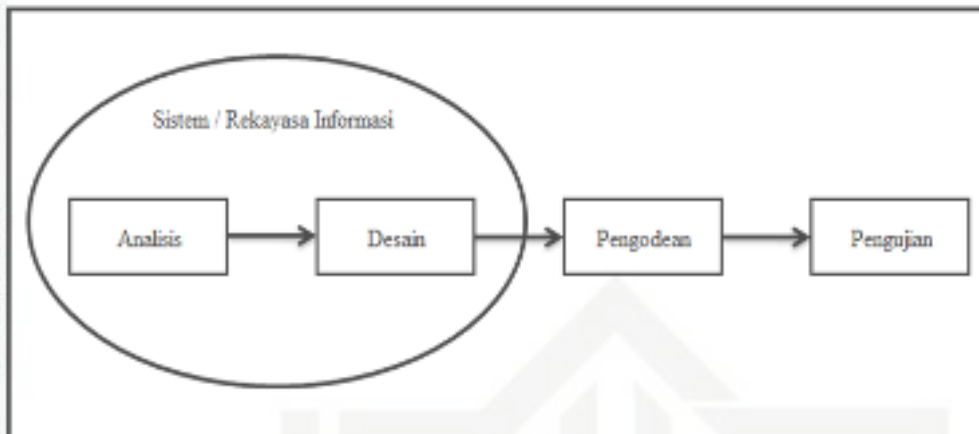
Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensi linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakann pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

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.



Gambar 2. 6. Ilustrasi Model *Waterfall*
(Rosa, 2014)

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi perangkat lunak, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

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 dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal 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 mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

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 tidak untuk membuat perangkat lunak baru.

2.11. Polres Kepulauan Meranti

Polres Kepulauan Meranti merupakan Polres terbungsu di wilayah hukum Polda Riau yang baru terbentuk selama 3 tahun lima bulan, tepatnya pada tanggal 17 Agustus 2013 dan diresmikan oleh Brigjen Pol Drs Condro Kirono selaku Kapolda Riau. Terbentuknya Polres Kepulauan Meranti karena telah terbentuknya Kabupaten Kepulauan Meranti yang meupakan pemekaran dari Kabupaten Bengkalis pada tanggal 19 desember 2009 lalu.

Dalam penugasannya diperbantukan oleh peronil polsek Tebing Tinggi serta dikirim sebanyak 172 orang personil. Setiap tahunnya personil Polres Kepulauan Meranti semakin bertambah, dari pertama terbentuknya Polres yang hanya di pimpin oleh Kapolres dan dbawahi Wakapolres serta personil Polsek yang diperbatukan dalam pelaksanaan tugas Polres Kepulauan Meranti, kemudian dikirim sebanyak 172 pesonil, hingga sampai 2017 tercatat sebanyak 350 lebih Personil Polres Kepulauan Meranti beserta Polsek jajaran.

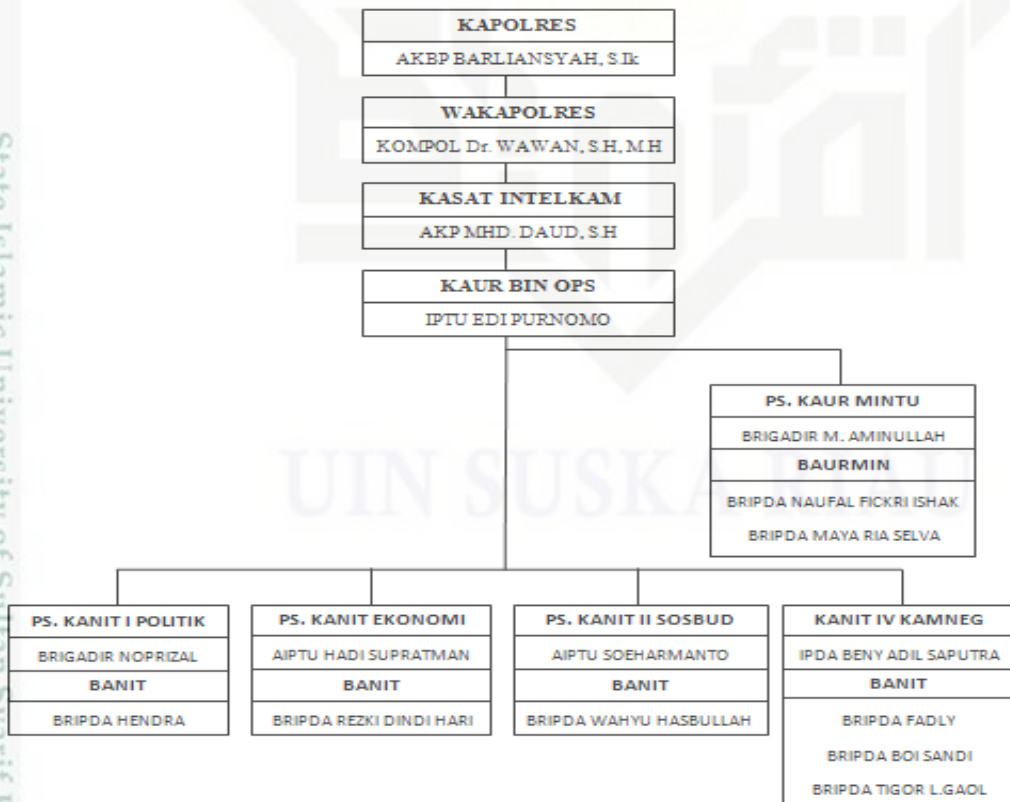
Polres dipimpin oleh seorang Ajun Komisaris Besar Polisi (AKBP) selaku Kapolres dan Komisaris Polisi (Kompol) selaku Wakapolres. Hingga saat ini Polres Kep. Meranti telah megalami 3 (tiga) kali pergantian kepemimpinan. Awal berdirinya Polres Kep. Meranti dipimpin oleh AKBP Zahwani Pandra Arsyad,S.H, M. Si dan Kompol STP Manulang, S.H periode agustus 2013 s/d juni 2016, dan dilanjutkan oleh AKBP Asep Iskandar, S.Ik dan Kompol R.A Samosir, S.E yang hanya berlangsung beberapa bulan saja. Mutasi selanjutnya

Kepemimpinan selaku Kapolres Kep. Meranti dipimpin oleh AKBP Barliansyah, S.Ik dan Kopol Dr. Wawan, S.H, M.H selaku Wakapolres hingga saat sekarang ini (Januari 2018).

2.11.1. Satuan Intelijen dan Keamanan (Sat Intelkam)

Sat Intelkam adalah unsur pelaksana utama Polres yang berada di bawah Kapolres, yang bertugas menyelenggarakan dan membina fungsi Intelejen bidang Keamanan, termasuk persandian, dan pemberi pelayanan dalam bentuk Surat izin dan Keterangan yang menyangkut Orang Asing, Senjata Api & Bahan Peledak, kegiatan social dan Politik masyarakat dan Surat Keterangan Catatan Kepolisian (SKCK) kepada warga masyarakat yang membutuhkan serta melakukan pengawasan dan pengamanan dan pelaksanaannya. Sat Intelkam dipimpin oleh Kepala Sat Intelkam, disingkat Kasat Intelkam yang bertanggung jawab kepada Kapolres dan dalam pelaksanaan tugas sehari-hari di bawah kendali Waka Polres.

2.11.2. Struktur Oraganisasi



Gambar 2. 7. Struktur Organisasi Sat Intelkam Polres Kep. Meranti