

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan Sistem

Perancangan sistem merupakan tahapan untuk menggambarkan model baru atau dikembangkan yang akan penulis buat.(Sandy Ferdinandus, dkk, 2013).

II.1.2. Tahap Perancangan

1. Tahap Pengembangan Sistem

Perancangan sistem dalam penelitian ini akan menggunakan metodologi pengembangan perangkat lunak *Rapin Application Development* (RAD).

2. Rencana Kebutuhan (*Requirement Planning*)

Rencana kebutuhan merupakan tahapan awal dan penting karena pada tahap ini dilakukan perancangan terhadap sistem, tujuan pembuatan sistem, fungsi-fungsi yang diinginkan serta pengkelompokan terhadap fitur-fitur yang harus ada dan fitur-fitur tambahan.

3. Proses Desain

Proses desain merupakan proses yang dilakukan setelah perancangan kebutuhan dilakukan, ini dikarenakan dalam melakukan desain terhadap suatu sistem pasti seorang analisis harus mengetahui spesifikasi atau kebutuhan dari sistem itu sendiri.(Talitha Almira, dkk, 2011)

II.2. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.(Asbon Hendra, 2012)

II.2.1. Karakteristik Sistem

Ada beberapa karakteristik yang membentuk sebuah system yaitu:

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen tersebut dapat berupa suatu bentuk subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan, karena dengan batas system ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu system menunjukkan ruang lingkup (*scope*) dari system tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu diluar batas sistem yang memengaruhi operasi dari suatu sistem. Lingkungan luar system ini dapat bersifat menguntungkan atau merugikan.Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pegaruhnya,

sedangkan lingkungan luar yang merugikan harus dimusnahkan atau dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain, *output* dari suatu subsistem akan menjadi *input* dari subsistem yang lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa perawatan (*maintenance input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem, meliputi *output* yang berguna.

7. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan menjadi keluaran yang diinginkan.

8. Tujuan Sistem (*Goal*)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang memengaruhi *input* yang dibutuhkan dan *output* yang dihasilkan. Dengan kata lain, suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai

sasaran atau tujuannya. Jika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

II.2.2. Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu sistem dapat diklasifikasikan dari beberapa sudut antara lain :

a. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem Fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya.

c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat, seperti sistem tata surya, sistem galaxy, sistem reproduksi, dan lain-lain.

d. Sistem Buatan Manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *Human Machine System*.

e. Sistem Tertentu (*Deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan.

f. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

g. Sistem Tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan system luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, system tersebut ada, tetapi kenyataannya tidak ada system yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup).

h. Sistem Terbuka (*Open System*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan system terotomasi, yang merupakan bagian dari system buatan manusia dan berinteraksi dengan control oleh satu atau lebih computer sebagai bagian dari system yang digunakan dalam masyarakat modern.

II.3. Sistem Pengelolaan

Sistem pengelolaan Surat Masuk dan Surat Keluar Terkomputerisasi adalah sebagai sarana informasi untuk staf tata usaha bagian persuratan agar dapat mempermudah dalam pengelolaan surat.(Sukadi Bibit, 2012)

II.4. Surat

Surat merupakan sarana komunikasi dalam bentuk tulisan, apapun medianya. Tata persuratan merupakan tahap penciptaan dalam daur ulang hidup Arsip. Kegiatan tata persuratan menyangkut materi yang lingkupnya esensial dalam komunikasi kedinasan meliputi penentuan jenis surat, sifat, format surat yang menampung bentuk redaksional serta penggunaan sarana pengamanan surat, serta kewenangan penandatanganan.(Dessi Tri Santi, 2013)

II.5. Dinas Komunikasi Dan Informatika Kota Medan

Dinas Komunikasi dan Informatika (DISKOMINFO) merupakan instansi yang bergerak di bidang komunikasi dan informatika yang meliputi telematika, serta pengolahan data elektronik. Dinas Komunikasi dan Informatika dalam pengolahan data elektronik telah menggunakan teknologi telekomunikasi berupa Jaringan Komputer dalam mengkomunikasikan antar OPD (Organisasi Perangkat Daerah) satu dengan lainnya yang terdapat di Propinsi Jawa Barat. Dengan menggunakan Jaringan Komputer, tugas-tugas komputasi dari berbagai daerah Jawa Barat dapat saling berkomunikasi dalam melaksanakan tugas-tugas tersebut.

Terdapat sedikitnya tiga kriteria paling penting dalam jaringan komputer yaitu performance, reliability, dan security. Untuk mengetahui apakah jaringan DISKOMINFO telah sesuai dengan kriteria jaringan, maka penulis melakukan analisis terhadap jaringan komputer DISKOMINFO Jabar.(Winarto Sugeng, dkk, 2012)

II.6. *SQL Server 2008*

Microsoft SQL Server 2008 R2 adalah generasi berikutnya dari *platform* Microsoft, dengan fitur baru yang memberikan kinerja yang lebih cepat, dan memberikan wawasan bisnis yang kuat. Microsoft SQL Server 2008 R2 telah membuat dampak pada organisasi di seluruh dunia dengan kemampuan inovasi, peningkatan efisiensi serta kolaborasi antara *database administrator* (DBA) dan pengembangan aplikasi, dan skala untuk mengakomodasi beban kerja data. (Talitha Almira, dkk, 2012)

II.7. *Visual Studio 2010*

Visual Studio adalah *Integrated Development (IDE)* dari Microsoft yang digunakan untuk membuat, menjalankan, dan memperbaiki program(aplikasi). Visual Studio dapat digunakan untuk membuat aplikasi dengan berbagai Bahasa yang didukung .NET seperti C++, C# dan Visual Basic.

II.8. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan


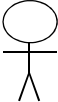
sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.


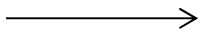
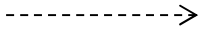
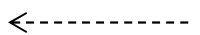
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*




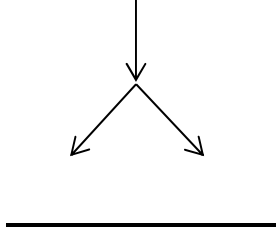
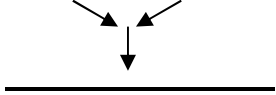
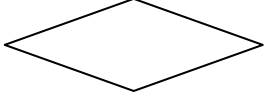

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus</p>

	ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran.
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

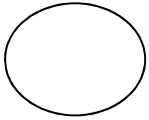
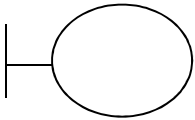
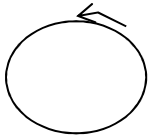
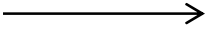
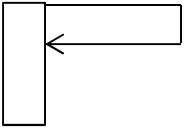


Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini:

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4