

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Perancangan

Perancangan sistem adalah penentuan proses data yang diperlukan oleh sistem baru. Jika sistem itu berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang digunakan. tahap-tahap perancangan sistem informasi adalah sebagai berikut :

- a. Menyiapkan rancangan sistem yang terinci
- b. Mengidentifikasi berbagai alternatif konfigurasi sistem
- c. Mengevaluasi berbagai alternatif konfigurasi sistem
- d. Memilih konfigurasi terbaik e. Menyiapkan usulan penerapan
- e. Menyetujui atau menolak penerapan sistem

System design is the specification or construction of a technical , computer based solution for the business requirements identified in a system analysis”. Yang diterjemahkan sebagai berikut: “Perancangan sistem adalah spesifikasi atau perwujudan dari solusi teknis berbasiskan komputer untuk kebutuhan bisnis yang diidentifikasi di sistem analisis”. Menurut Romney dan Steinbart (2006, p792): “System design is the process of preparing detail specifications for development of a new information system”. Yang diterjemahkan sebagai berikut: “Perancangan sistem adalah suatu proses detail spesifikasi untuk mengembangkan sebuah sistem informasi yang baru”. Berdasarkan pendapat-pendapat diatas, dapat disimpulkan bahwa perancangan sistem adalah proses

mengimplementasikan hasil-hasil dari analisis sistem ke dalam suatu rancangan sistem yang baru (Henny Hendarti ; 2009 : 158).

Model perancangan sesungguhnya adalah modal objek yang mendeskripsikan realisasi fisik *use case* dengan cara berfokus pada bagaimana spesifikasi-spesifikasi kebutuhan fungsional dan *non-fungsional*, bersama dengan batasan-batasan lain yang berhubungan dengan lingkungan implemenatasi, memiliki imbas langsung pada pertimbangan-pertimbangan pada aktivitas-aktivitas yang dilakukan pada tahap implementasi. Tambahannya, model perancangan sesungguhnya secara langsung bertindak sebagai abstraksi implementasi sistem/perangkat lunak dan dengan sendirinya model perancangan suatu saat nanti akan menjadi asupan bagi aktivitas-aktivitas selanjutnya yang kelak akan terdefinisi pada tahap implementasi (Adi Nugroho ; 2010 : 212).

II.2. Pengertian Aplikasi

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011: 3).

II.3. Kriptografi

Kriptografi berasal dari bahasa Yunani, “kryptós” yang berarti tersembunyi dan “gráphein” yang berarti tulisan. Sehingga kata kriptografi dapat diartikan menjadi “tulisan tersembunyi”. Menurut Request for Comments (RFC), Kriptografi adalah ilmu matematika yang berhubungan dengan transformasi data agar arti dari data tersebut menjadi sulit untuk dipahami (untuk menyembunyikan maknanya), mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga dapat diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang mudah dipahami. Sehingga, kriptografi juga dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas.

Ilmu kriptografi adalah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan. Kriptografi sudah dipakai sejak jaman Julius Caesar dimana akan mengirimkan pesan kepada panglimanya tetapi tidak mempercayai kurir pembawa pesan tersebut.

Kriptografi mempunyai 2 (dua) bagian yang penting, yaitu enkripsi dan dekripsi. Enkripsi adalah proses dari penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Dekripsi sendiri berarti merubah pesan yang sudah disandikan menjadi pesan aslinya. Pesan asli biasanya disebut plaintext, sedangkan pesan yang sudah disandikan disebut ciphertext (Aghus Sofwan ; 2014 : 2).

a. Plaintext

Plaintext merupakan pesan asli yang belum disandikan atau informasi yang ingin dikirimkan atau dijaga keamanannya.

b. Ciphertext

Ciphertext merupakan pesan yang telah disandikan sehingga siap untuk dikirimkan.

c. Enkripsi

Enkripsi merupakan proses yang dilakukan untuk menyandikan Plaintext menjadi Ciphertext dengan tujuan pesan tersebut tidak dapat dibaca oleh pihak yang tidak berwenang.

d. Dekripsi

Dekripsi merupakan proses yang dilakukan untuk memperoleh kembali Plaintext dari Ciphertext.

e. Kriptosistem

Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi (Aghus Sofwan ; 2014 : 3).

II.4. Pengertian Email

Email adalah surat elektronik yang dikirim dengan menggunakan internet, seperti layaknya surat biasa e-mail dapat ditujukan ke perorangan dan kelompok. Email bisa menjangkau seluruh dunia dengan karena didukung jaringan gbal. Dengan email maka surat menyurat dapat dilakukan dengan cepat tanpa harus menunggu tukang pos datang mengirim surat. Pengiriman email ke seluruh dunia tidak dibedakan biayanya baik jarak dekat atau jauh semuanya sama.

Fasilitas e-mail dengan bantuan netscape cukup baik dan sangat mudah cara pemakaiannya. Anda tinggal klik ikon anvelope atau cukup menekan tombol to mail apabila anda akan menulis dan mengirim e-mail yang masuk saat itu.

a. E-mail Gratis

Dengan menjamunya warnet (warung Internet) di hampir semua di kota-kota besar diseluruh Indonesia, kita bisa memiliki e-mail sendiri tanpa harus mendaftar menjadi anggota salah satu provider. Email gratis adalah alamat email yang dapat diperoleh dengan gratis pada internet. Untuk dapat berinternet dan memiliki alamat email tidak perlu mendaftar ke sebuah ISP cukup menyewa di warnet.

b. Registrasi E-mail Gratis

Sebelum masuk ke salah satu program browser untuk membuka situs internet, terlebih dahulu harus mengkoneksikan komputer dengan provider. Atau jika tidak berlangganan internet anda bisa datang ke Warnet terdekat atau lab komputer yang menyediakan fasilitas internet. Jalankan salah satu program browser, seperti internet explorer, pointCash, Netscape Communicator setelah berada salah satu browser tersebut, pada kota location/address tuliskan salah satu alamat situs internet misalnya <http://www.yahoo.com> (Sunarto ; 2014 :).

II.5. RC4

Algoritma RC4 RC4 merupakan stream cipher yang didesain oleh Rivest untuk RSA Data Security (sekarang RSA Security) pada 1987. RC4 menggunakan panjang variabel kunci dari 1 s.d 256 byte untuk menginisialisasi state tabel. State table digunakan untuk pengurutan menghasilkan byte pseudorandom yang kemudian menjadi stream pseudo-random. Setelah di-XOR dengan plaintext sehingga didapatkan ciphertext. Tiap elemen pada state table di swap sedikitnya sekali. Kunci RC4 sering dibatasi sampai 40 bit, tetapi dimungkinkan untuk menggunakan kunci 128 bit. RC4 memiliki kemampuan penggunaan kunci antara 1 sampai 2048 bit (Yuri Ariyanto ; 2013 : 2).

II.6. Javascript

Javascript bukan bahasa berorientasi objek, melainkan bahasa berbasis objek. Bahasa berorientasi objek harus mendukung tiga konsep dasar, yaitu pengkapsulan (*encapsulation*), perwarisan (*inheritance*) dan polimorfisme (*polymorphism*). Javascript hanya mendukung pengkapsulan, itupun tidak 100% benar.

Program *JavaScript* dituliskan pada file *HTML* (.html atau .htm) dengan menggunakan tag container `<SCRIPT>`. Dengan kata lain, anda tidak perlu menuliskan program *JavaScript* pada file terpisah (meskipun anda bisa juga melakukannya). Ingat bahwa yang dimaksud dengan tag container adalah tag yang diawali dengan `<NAMA_TAG>` dan diakhiri dengan `</NAMA_TAG>`. Beberapa contoh tag container adalah `<HTML></HTML>`, `<HEAD></Body>`, dsb (Antony Pranata ; 2001 : 11).

II.7. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.7.1. Bentuk-bentuk Normalisasi

1. Bentuk normal tahap pertama (1st Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional.

Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Yuniar Supardi, 2010 : 448).

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

Menurut Windu Gata (2013 : 4) 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.

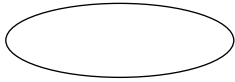
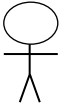
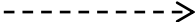
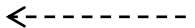
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Use case Diagram

Use case diagram merupakan pemodelan untuk melakukan (*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, yaitu :

Tabel II.2. Simbol Use Case

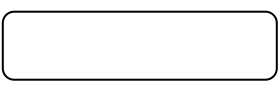
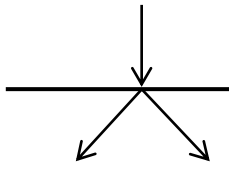
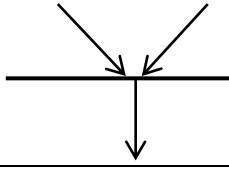
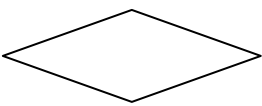
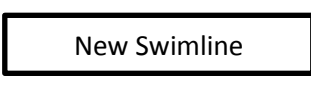
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau 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.

(Sumber : Windu Gata, 2013 : 4)

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*, yaitu :

Tabel II.3. Simbol Activity Diagram

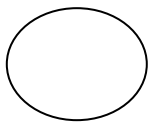
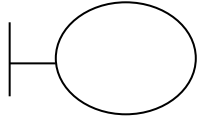
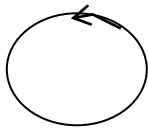
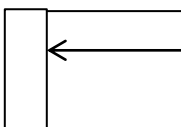
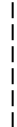
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau 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.

(Sumber : Windu Gata, 2013 : 6)

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*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<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.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata, 2013 : 7)

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.

Tabel II.5. *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

(Sumber : Windu Gata, 2013 : 9)