BAB II

TINJAUAN PUSTAKA

II.1. Rancang Bangun

Rancang Bangun (desain) adalah tahap dari setelah analisis dari siklus Pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan Fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasikan dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

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. Definisi Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama sama menggunakan hardware/software yang terhubung dengan jaringan. Tiap komputer, printer atau periferal yang terhubung dengan jaringan disebut node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node (Marco Van Basten ; 2009 : 5).

II.3.1. Sejarah dan perkembangan Jaringan Komputer Internet

1. Bulan September 1940 : George Stibitz dari Dartmouth College di New hamspire menggunakan mesin teletip untuk mengirimkan beberapa masalah ke Complex Number Calculator di kota New York. George pun menerima balasan dengan alat yang sama. Kesuksesan inilah yang menarik perhatian ARPA (Advanced Research Projects Agency);
2. Tahun 1962 : Seorang bernama J.C.R. Liackliader mengembangkan sebuah grup jaringan yang disebut “Intergalactic Network”, sebagai cikal bakal pengembangan ARPAnet.
3. Tahun 1964 : Seorang Peneliti di Darthmouth mengembangkan sistem tersebut pada komputer. Pada tahun yang sama, MIT (Massachusetts Institute of Technology) membentuk grup penelitian yang didukung oleh General Electric dan Bell Labs, dengan komputer bernama PDP-8 untuk mengarahkan dan mengatur koneksi telepon.

4. Paul Baran mengusulkan model jaringan yang terdiri dari datagram dan paketpaket yang dapat digunakan dalam sistem jaringan komputer .
5. 1969 : Empat Universitas di Amerika yaitu : universitas California di Los Angeles, Universitas SRI di Stanford, Universitas California di santa barbara, dan Universitas Utah mereka saling terhubung dengan jaringan 50 kb. Inilah yang disebut proyek ARPAnet, cikal bakal terjadinya Internet. Oleh sebab itu ARPANET dipecah manjadi dua, yaitu MILNET untuk keperluan militer dan ARPANET baru yang lebih kecil untuk keperluan non-militer seperti, universitas-universitas. Dan gabungan kedua jaringan tersebut, akhirnya dikenal dengan nama DARPA Internet, yang kemudian disederhanakan menjadi Internet

Jaringan komputer pada hakekatnya adalah dua komputer atau lebih yang terhubung satu dengan yang lainnya. Perangkat yang dihubungkan tidak terbatas pada komputer saja, melainkan termasuk printer dan perangkat-perangkat kertas yang lain. Sebagai penghubung, dapat digunakan kabel, misalnya gelombang radio dan sinar inframerah (Marco Van Basten ; 2009 : 5).

Sebuah jaringan biasanya terdiri dari 2 atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya CDROM, Printer, pertukaran file, atau memungkinkan untuk saling berkomunikasi secara elektronik (Marco Van Basten ; 2009 : 5).

Tujuan dari jaringan komputer adalah :

1. Membagi sumber daya, misalnya printer, CPU, memory ataupun harddisk.
2. Komunikasi, misalnya *e-mail, instant messaging, chatting.*
3. Akses Iinformasi misalnya *web browsing.*

Dalam sebuah jaringan / *network*, antara satu komputer dengan computer lainnya dihubungkan dengan menggunakan kabel ataupun nirkabel. Pada awal perkembangannya, jaringan / *network* kerap kali menggunakan media kabel, namum seiring dengan perkembangan dunia teknologi informasi yang kian pesat penggunaan media nirkabel / *wireless* kini sudah banyak diterapkan. Hal ini dikarenakan semakin banyak *user* yang menggunakan *laptop* / *notebook*, sehingga *user* dapat mengakses ke dalam jaringan secara mobilitas.

Dan berdasarkan arah transmisinya, komunikasi data dibedakan menjadi :

1. *Simplex*

   Pada *simplex*, signal hanya ditransmit satu arah saja dimana satu stasiun sebagai pemancar dan yang lainnya sebagai penerima.Pada sistem ini aliran data hanya dapat terjadi ke satu arah saja

2. *Half-duplex*

   Dalam operasi ini, kedua stasiun mungkin melakukan pengiriman, tapi tidak bisa bersamaan melainkan secara beroperasi bergantian. Pada sistem ini aliran informasi dapat terjadi kedua arah tapi tidak bersamaan.

3. *Full-duplex*

Dalam operasi *full-duplex*, kedua stasiun mungkin mentransmisi secara serentak. Pada sistem ini aliran dapat terjadi kedua arah pada saat yang bersamaan. Sistem ini dapat terjadi hanya menggunakan sebuah saluran komunikasi data ataudengan menggunakan dua saluran komunikasi data.

II.4. Cloud Storage

Cloud Storage Services pribadi adalah salah satu contoh dari cloud computing. Cloud storage memungkinkan untuk melakukan sinkronisasi folder lokal dengan server di awan (internet). Layanan penyimpanan awan telah mendapatkan popularitas, dengan perusahaan yang menawarkan sejumlah besar penyimpanan jarak jauh (remote storage) dengan harga murah atau bahkan gratis. Semakin banyak orang tertarik dengan tawaran ini, seperti menyimpan file pribadi, sinkronisasi perangkat dan berbagi konten dengan kesederhanaan besar. Ketertarikan publik yang tinggi mendorong berbagai penyedia untuk memasuki pasar penyimpanan awan. Layanan seperti Dropbox, SkyDrive dan Google Drive menjadi meresap dalam rutinitas masyarakat. Aplikasi seperti ini penggunaannya cukup meningkat dan menghasilkan porsi yang signifikan dari lalu lintas Internet.

Salah satu layanan penyimpanan awan tersebut di atas adalah Google Drive. Google Drive adalah layanan populer, menyediakan pengguna dengan biaya-efektif, dan dalam beberapa kasus bebas biaya, kemampuan untuk mengakses, menyimpan, berkolaborasi, dan menyebarkan data (Quick & Choo, 2014). Google drive merupakan raksasa dalam bidang penyimpanan awan yang menjadi pesaing besar bagi perusahaan penyimpanan awan lainnya seperti Drop Box, SugarSync, Insync, LogMeIn Cubby, Apple iCloud, SkyDrive, Mozy Stash, SpiderOak, AVG LiveKive, Wuala by LaCie, Box, Syncplicity (Azuar Juliandi ; 2014 : 3)

II.5. Algoritma El-Gamal

Algoritma ElGamal diciptakan oleh Taher ElGamal pada tahun 1984. Algoritma ini pada mulanya digunakan untuk kepentingan *digital signature*, namun kemudian dimodifikasi sehingga algoritma ElGamal bisa digunakan untuk enkripsi dan dekripsi. ElGamal digunakan di dalam perangkat lunak sekuriti yang dikembangkan oleh GNU, program PGP dan pada sistem sekuriti

lainnya. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.

Logaritma ini disebut logaritma diskret karena nilainya berhingga dan bergantung pada bilangan prima yang digunakan. Karena bilangan prima yang digunakan adalah bilangan prima besar, maka sangat sulit bahkan tidak mungkin menurunkan kunci privat dari kunci publik yang diketahui walaupun serangan dilakukan dengan menggunakan sumberdaya komputer yang sangat besar.

1. Kelebihan Algoritma *ElGamal*

Algoritma ElGamal dikenal sebagai kriptografi *digital signature* karena algoritma ini berfungsi dengan baik untuk mengirimkan sebuah tanda tangan digital pada sebuah pesan. Kelebihan dari algoritma ElGamal yaitu:

1) Plainteks yang sama dapat diubah menjadi chiperteks yang berbeda, karena bilangan bulat pada algoritma Elgamal dapat dipilih secara acak untuk menentukan kunci.
2) Pada algoritma ElGamal tidak hanya kunci privat yang perlu dijamin kerahasiannya, tetapi autentikasi kunci publik juga harus tetap dijaga.
3) Kunci publik dan kunci privat pada algoritma ElGamal tidak perlu diubah dalam periode waktu yang panjang.

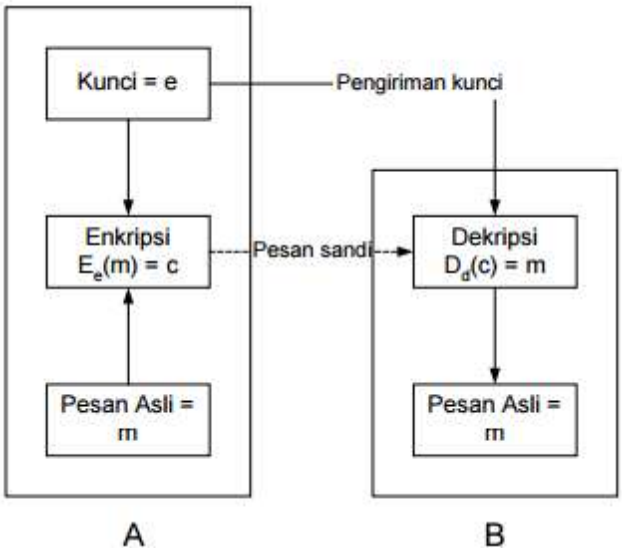
Algoritma ElGamal bisa dimanfaatkan untuk mengirimkan sebuah pesan rahasia, yaitu dengan menentukan kunci dari sebuah kriptografi simetris (Anandia Zelvina ; 2012 : 59).

II.5.1. Kriptografi Bilangan Prima

Kriptografi Simetris vs Kriptografi Kunci Publik Kriptografi adalah teknik untuk menyamarkan suatu pesan. Kriptografi meliputi enkripsi, yaitu transformasi data ke bentuk yang tidak mungkin dibaca pihak lain tanpa mengetahui

kuncinya, serta dekripsi, yang merupakan kebalikan dari enkripsi, yaitu mengembalikan data yang ditrans- formasi ke bentuk semula. Baik enkripsi maupun dekripsi selalu membutuhkan suatu informasi rahasia yang disebut kunci.

Berdasarkan sifat kuncinya, terdapat 2 jenis kriptografi yaitu kriptografi simetris (kunci rahasia) dan kriptografi dengan kunci publik. Dalam kriptografi simetris, kunci yang sama dipakai dalam enkripsi dan dekripsi sehingga baik pengirim maupun penerima informasi harus memiliki kunci yang sama untuk mengolahnya. Keadaan ini dapat digambarkan dalam gambar 1.



Gambar II.1 : Skema Enkripsi pada Kriptografi Simetris

II.6. Android Versi (*Ice Cream Sandwich*)

Android adalah system operasi disematkan pada gadget, baik itu hanphone, tablet, juga sekarang sudah merambah ke kamera digital dan jam tangan. Saat ini gadget berbasia Android, baik itu tablet atau handphone, begitu digandrungi. Selain harganya yang semakin terjangkau, juga banyak varian spesifikasi yang bias dipilh sesuai kebutuhan sdan kantong.

Untuk kebuthan yang lebih praktis, tablet dan handphone pintar ini bias menggantikan peran dari sebuah komputer jinjing, terutama untuk kebutuhan entertainment, seperti mendengarkan lagu, menonton video, mengirim email, juga kegitan hiburan online lainnya.

Perkembangan Android sangat cepat. Di awal tahun 2012 saja ada 200 juta pengguna aktif Android, dan Google Play mampu menampung400.000 aplikasi yang siap digunakan, dan total mencapai 10 triliun kali aplikasi yang sudah di download lewat Android Market, pertumbuhan yang luar biasa. Jumlah ini diyakini akan terus bertambah seiring waktu dan perkembangan teknologi. (Agus Wahadyo ; 2013 : 1).

II.7. 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.

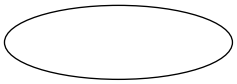
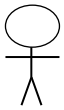
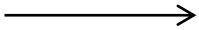
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Usecase* Diagram

   *Usecase*diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Usecase*mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* 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 *usecase* diagram, yaitu :

Tabel II.1. Simbol *UseCase*

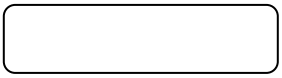
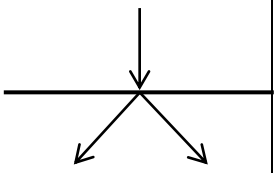
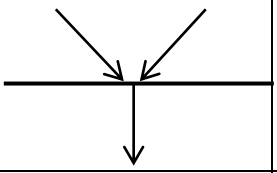
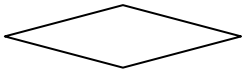
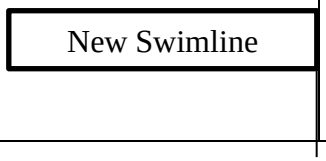
Gambar	Keterangan
	*Usecase* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *usecase*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 *usecase*, tetapi tidak memiliki control terhadap *usecase*.
	Asosiasi antara aktor dan *usecase*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *usecase* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *usecase* lain (*required*) atau pemanggilan *usecase* oleh *usecase* lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *usecase* lain jika kondisi atau syarat terpenuhi.

(Sumber : WinduGata ; 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.2. Simbol *Activity Diagram*

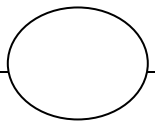
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*Endpoint*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork*(Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua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

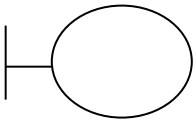
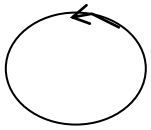
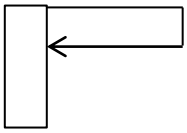
(Sumber : WinduGata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang

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

(Sumber : WinduGata ; 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 tanggng 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.4. *MultiplicityClass 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 : WinduGata ; 2013 : 9)