BAB II

LANDASAN TEORI

2.1 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan; lamaran; penggunaan. Secara istilah aplikasi adalah: program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

2.1.1 Konsep Dasar Aplikasi

Menurut Dhanta (2009:32), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas - tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*. Sedangkan menurut Anisyah (2000:30), aplikasi adalah penerapan, penggunaan atau penambahan.

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

2.1.2 Klasifikasi Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (*enterprise*)
   - 1.1 Perangkat lunak akuntansi
   - 1.2 Bagian pendukung (*back office*)
   - 1.3 Perangkat lunak bisnis
   - 1.4 Manajemen sumber daya manusia

2.Perangkat lunak infrastruktur perusahaan

2.1 Perangkat lunak alur kerja bisnis

2.2 Sistem manajemen basis data

2.3 Manajemen aset *digital*

2.4 Manajemen dokumen

3.Perangkat lunak informasi kerja

3.1 Manajemen waktu dan sumber daya

3.2 Manajemen data

3.3 Dokumentasi

3.4 Perangkat lunak analitik

3.5 Perangkat lunak kolaborasi

4.Perangkat lunak media dan hiburan

4.1 Media *digital*

4.2 Perangkat lunak hiburan

5.Perangkat lunak pendidikan

5.1 Manajemen ruang kelas

5.2 Manajemen survei

5.3 Manajemen pelatihan

5.4 Perangkat lunak kesiapan penjualan

6.Perangkat lunak pengembangan media

6.1 Perangkat lunak seni grafis

6.2 Penyuntingan media

7.Perangkat lunak rekayasa produk

7.1 Rekayasa perangkat keras

7.2 Rekayasa perangkat lunak

2.1.3 Jenis Aplikasi

Aplikasi perangkat lunak akan sangat membantu pekerjaan kita sehari-hari, dan sudah banyak dipasaran. Ada beberapa jenis aplikasi perangkat lunak yaitu yang menangani masalah: keuangan, basis data, pengolah kata, grafik, *spreadsheet*, gambar, *multimedia*, internet, dan lain-lain. Semua aplikasi perangkat lunak tersebut saat ini sudah bisa dioperasikan di bawah *Operating System Windows*. Aplikasi perangkat lunak tersebut digolongkan menjadi beberapa bagian yaitu:

1. Perangkat Lunak Sistem (*System Software*)

   Perangkat lunak system merupakan sekumpulan program yang ditulis untuk melayani program-program yang lain.

2. Perangkat Lunak Real-Time (*Real Time Software*)

   Program - program yang memonitor / menganalisis / mengontrol kejadian-kejadian yang sesungguhnya (sedang berlangsung) pada dunia disebut dengan perangkat lunak *real-time*.

3. Perangkat Lunak Bisnis (*Business Software*)

   Pemrosesan informasi bisnis merupakan area aplikasi yang paling luas. Aplikasi dalam area ini akan menyusun kembali struktur data yang ada dengan cara tertentu untuk memperlancar operasi bisnis atau pengambilan keputusan manajemen.

4. Perangkat Lunak Teknik dan Ilmu Pengetahuan (*Engineering and Scientific Software*)

Perangkat lunak teknik dan ilmu pengetahuan ditandai dengan algoritma number crunching. Perangkat lunak ini memiliki jangkauan aplikasi mulai dari astronomi sampai vulkanologi, dari analisis otomotif sampai dinamika orbit pesawat luar angkasa, dan dari biologi *molecular* sampai pada pabrik yang sudah diotomatisasi.

5. Perangkat Lunak Embedded (*Embedded Sofware*)

Embedded software ada dalam *real-only memory* dan dipakai untuk mengontrol hasil serta *system* untuk keperluan konsumen dan pasar *industry*.

6. Perangkat Lunak Komputer Personal (*Personal Computer Software*)

Pasar perangkat lunak komputer *personal* telah berkembang selama *decade* terakhir. Pengolahan kata, *spreadsheet*, grafik komputer, *multimedia*, hiburan, manajemen *database*, aplikasi keuangan bisnis dan *personal*, jaringan eksternal atau akses *database* merupakan beberapa contoh yang dapat disebutkan.

7. Perangkat Lunak kecerdasan Buatan (*Artifical Intelligence Sofware*{AI})

Perangkat lunak kecerdasan buatan (*Artifical Intelligence Sofware*{AI}) menggunakan algoritma non-numerik untuk memecahkan masalah kompleks yang tidak sesuai untuk perhitungan atau analisis secara langsung. Area kecerdasan buatan yang aktif adalah sistem pakar (*expert system*), sering disebut dengan sistem berbasis ilmu pengetahuan.

2.1.4 Karakteristik Aplikasi

Penelitian dan pemahaman tentang karakteristik perangkat lunak sangatlah penting, untuk memperoleh pemahaman tentang perangkat lunak yang pada dasarnya berbeda dengan hal - hal lain yang dibangun oleh manusia. Perangkat lunak lebih merupakan elemen logika dan bukan merupakan elemen *system* fisik. Menurut Pressman (2001), ada beberapa karakteristik perangkat lunak dibandingkan dengan perangkat keras yang perlu diketahui, yaitu:

1. Perangkat lunak itu dibangun atau direkayasa, tidak dihasilkan dengan menggunakan mesin/tangan seperti perangkat keras.
2. Perangkat lunak itu tidak akan usang atau lusuh sejalan dengan waktu.
3. Perangkat lunak akan terus menerus dibuat menurut pesanan, sadangkan perangkat keras akan disusun dengan menggunakan komponen.

2.2 Pelayanan

Menurut Kotler (2002:83) definisi pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun. Produksinya dapat dikaitkan atau tidak dikaitkan pada satu produk fisik. Pelayanan merupakan perilaku produsen dalam rangka memenuhi kebutuhan dan keinginan konsumen demi tercapainya kepuasan pada konsumen itu sendiri. Kotler juga mengatakan bahwa perilaku tersebut dapat terjadi pada saat, sebelum dan sesudah terjadinya transaksi.

Serta Menurut Moenir (2002:27) pelayanan hakekatnya adalah serangkaian kegiatan, karena itu pelayanan juga merupakann sebuah proses, pelayanan berjalan secara rutin dan berkesinambungan meliputi seluruh kehidupan dalam masyarakat.

2.3 Listrik

Listrik adalah aliran electron - elektron dari atom ke atom pada sebuah penghantar. Semua atom memiliki partikel yang disebut elektron terletak pada orbitnya mengelilingi proton. Atom yang paling sederhana adalah atom *Hydrogen* (Atom Air), yaitu hanya mempunyai satu elektron yang mengelilingi satu proton. Di dalam kelistrikan akan dihasilkan listrik statis yang dibangkitkan dengan menggosokkan sebatang gelas, kemudian banyak teori yang tumbuh dan sekarang teori itu diterima dan disebut ”teori elektron” yang timbul sekitar tahun 1900. Diakhir abad kedelapan belas ketika pertama kali sumber listrik ditemukan oleh Volta Galvani sehingga mungkin untuk dipelajari efek kelistrikannya diatur oleh hukum tertentu sehingga mungkin untuk dihitung efeknya.

Arus listrik dapat disamakan dengan cairan di dalam sebuah pipa bila disambungkan sebuah penghantar ke pole-pole sumber arus. Arus listrik berarti arus dari listrik yang mengalir melalui penghantar dan consumer - konsumer pada suatu rangkaian tertutup. Arus listrik menimbulkan efek di dalam penghantar dan pada konsumer.

2.3.1 Listrik Prabayar

Sistem listrik prabayar, Pelanggan harus mengeluarkan uang atau membayar dulu energi listrik yang akan dikonsumsinya. Besar energi listrik yang telah dibeli oleh Pelanggan dimasukkan ke dalam Meter Prabayar (MPB) yang terpasang dilokasi Pelanggan melalui sistem '*token*' atau *stroom.*

MPB menyediakan informasi jumlah energi listrik (kWh) yang masih bisa dikonsumsi. Persediaan kWh tersebut bisa ditambah berapa saja dan kapan saja sesuai kebutuhan dan keinginan Pelanggan. Dengan demikian, Pelanggan bisa lebih mudah mengoptimalkan konsumsi listrik dengan mengatur sendiri jadwal dan jumlah pembelian listrik. Dengan sistem prabayar, Pelanggan tidak perlu berurusan dengan pencatatan meter yang bisasanya dilakukan setiap bulan, dan tidak perlu terikat dengan jadwal pembayaran listrik bulanan.

2.4 SMS (*Short Message Service*)

SMS (*Short Message Service*) ialah layanan pesan singkat yang terdapat pada telepon seluler. SMS adalah sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam *alphanumeric* antara terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti *email, paging, voice mail,*dan lain-lain.

Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah melakukan pengiriman pesan singkat dari satu terminal pelanggan ke terminal lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center* (SMSC) disebut juga *Message Center*. SMSC merupakan sebuah perangkat yang merupakan tugas *store and*

forward trafik *short message.* Di dalamnya termasuk penentuan atau pencarian rute tujuan akhir dari *short message.*

2.4.1 SMS *Gateway*

SMS *Gateway* adalah suatu platform yang menyediakan mekanisme untuk EUA menghantar dan menerima SMS dari peralatan *mobile* (HP, PDA phone, dll) melalui *SMS Gateway shortcode* (sbg contoh 9221).

SMS *Gateway* membolehkan UEA untuk berkomunikasi dengan Telco SMSC (telkomsel, indosat, dll) atau SMS *platform* untuk menghantar dan menerima pesan SMS dengan sangat mudah, Karena SMS *Gateway* akan melakukan semua proses dan koneksi dengan Telco. SMS *Gateway* juga menyediakan UEA dengan *interface* yang mudah dan standar.

UEA dapat berupa berbagai aplikasi yang memerlukan penggunaan SMS. Seperti berbagai aplikasi *web* yang telah banyak menggunakan SMS (*free* sms, pendaftaran, konfirmasi melalui SMS, aplikasi perkantoran, dsb), CMS, acara pengundian di televisi, dll.

UEA melakukan komunikasi dengan SMS *Gateway* melalui Internet menggunakan *standard* HTTP *GET* atau HTTPS (untuk komunikasi yang aman). Telco SMSC akan menghantar pesan (SMS) tersebut kepada perusahaan SMS *Gateway* (sesuai dengan nomor yang telah disewa) dengan menggunakan protokol yang khusus. Dan berdasarkan *keyword* yang telah dituliskan pada SMS, maka sistem SMS *Gateway* akan menghantar SMS tersebut ke URL yang telah ditentukan. UEA dapat menghantar SMS *reply* kepada pelanggan melalui SMS

Gateway tersebut. Dan UEA dapat menentukan besarnya biaya (*charging*) yang akan dikenakan kepada pelanggan.

2.5 Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer, *software* dan perangkat jaringan lainnya yang bekerja bersama-sama untuk mencapai suatu tujuan yang sama. Tujuan dari jaringan komputer adalah:

1. Membagi sumber daya: contohnya berbagi pemakaian printer, CPU, memori, *harddisk.*
2. Komunikasi: contohnya surat elektronik, *instant messaging, chatting.*
3. Akses informasi: contohnya *web browsing*

Agar dapat mencapai tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan layanan (*service*). Pihak yang meminta atau menerima layanan disebut klien (*client*) dan yang memberikan/mengirim layanan disebut pelayan (*server*). Arsitektur ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

2.5.1 Klasifikasi Jaringan Komputer

1. Berdasarkan geografis :
   a. *Local Area Network* (LAN): suatu jaringan komputer yang menghubungkan suatu komputer dengan komputer lain dengan jarak yang terbatas.
   b. *Metropolitant Area Network* (MAN): prinsip sama dengan LAN, hanya saja jaraknya lebih luas, yaitu 10-50 km.

c. *Wide Area Network* (WAN): jaraknya antar kota, negara, dan benua. ini sama dengan internet.

2. Berdasarkan topologi jaringan

a. Topologi bus

Topologi bus merupakan topologi yang banyak digunakan pada masa penggunaan kabel sepaksi menjamur. Dengan menggunakan *T-Connector* (dengan terminator 50ohm pada ujung *network*), maka komputer atau perangkat jaringan lainnya bisa dengan mudah dihubungkan satu sama lain.

Kesulitan utama dari penggunaan kabel sepaksi adalah sulit untuk mengukur apakah kabel sepaksi yang digunakan benar-benar matching atau tidak. Karena kalau tidak sungguh-sungguh diukur secara benar akan merusak NIC (*network interface card*) yang digunakan dan kinerja jaringan menjadi terhambat, tidak mencapai kemampuan maksimalnya. Topologi ini juga sering digunakan pada jaringan dengan basis *fiber optic* (yang kemudian digabungkan dengan topologi star untuk menghubungkan dengan *client* atau *node*.).

Pada topologi bus dua ujung jaringan harus diakhiri dengan sebuah terminator. *Barel connector* dapat digunakan untuk memperluasnya. Jaringan hanya terdiri dari satu saluran kabel yang menggunakan kabel BNC. Komputer yang ingin terhubung ke jaringan dapat mengkaitkan dirinya dengan men tap Ethernetnya sepanjang kabel.

Instalasi jaringan Bus sangat sederhana, murah dan maksimal terdiri atas 5-7 komputer. Kesulitan yang sering dihadapi adalah kemungkinan terjadinya tabrakan data karena mekanisme jaringan relatif sederhana dan jika salah satu *node* putus maka akan mengganggu kinerja dan trafik seluruh jaringan.

b. Topologi Bintang

Topologi bintang merupakan bentuk topologi jaringan yang berupa konvergensi dari node tengah ke setiap *node* atau pengguna. Topologi jaringan bintang termasuk topologi jaringan dengan biaya menengah.

c. Topologi cincin

Topologi cincin adalah topologi jaringan berbentuk rangkaian titik yang masing-masing terhubung ke dua titik lainnya, sedemikian sehingga membentuk jalur melingkar membentuk cincin. Pada topologi cincin, komunikasi data dapat terganggu jika satu titik mengalami gangguan. Jaringan FDDI mengantisipasi kelemahan ini dengan mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan. Topologi ring digunakan dalam jaringan yang memiliki performance tinggi, jaringan yang membutuhkan *bandwidth* untuk fitur yang time-sensitive seperti *video* dan *audio*, atau ketika *performance* dibutuhkan saat komputer yang terhubung ke jaringan dalam jumlah yang banyak.

d. Topologi mesh

Topologi jala atau Topologi mesh adalah suatu bentuk hubungan antar perangkat dimana setiap perangkat terhubung secara langsung ke perangkat lainnya yang ada di dalam jaringan. Akibatnya, dalam topologi mesh setiap perangkat dapat berkomunikasi langsung dengan perangkat yang dituju (*dedicated links*).

Dengan demikian maksimal banyaknya koneksi antar perangkat pada jaringan bertopologi mesh ini dapat dihitung yaitu sebanyak n(n-1)/2. Selain itu karena setiap perangkat dapat terhubung dengan perangkat lainnya yang ada di dalam jaringan maka setiap perangkat harus memiliki sebanyak n-1 *Port Input* atau *Output* (I/O *ports*).

Berdasarkan pemahaman di atas, dapat dicontohkan bahwa apabila sebanyak 5 (lima) komputer akan dihubungkan dalam bentuk topologi mesh maka agar seluruh koneksi antar komputer dapat berfungsi optimal, diperlukan kabel koneksi sebanyak 5(5-1)/2 = 10 kabel koneksi, dan masing-masing komputer harus memiliki port I/O sebanyak 5-1 = 4 port.

e. Topologi pohon

Topologi Pohon adalah kombinasi karakteristik antara topologi bintang dan topologi bus. Topologi ini terdiri atas kumpulan topologi bintang yang dihubungkan dalam satu topologi bus sebagai jalur tulang punggung atau *backbone*. Komputer-komputer dihubungkan ke hub, sedangkan hub lain di hubungkan sebagai jalur tulang punggung.

Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer.

f. Topologi linier

Jaringan komputer dengan topologi runtut (*linear topology*) biasa disebut dengan topologi bus beruntut, tata letak ini termasuk tata letak umum. Satu kabel utama menghubungkan tiap titik sambungan (komputer) yang dihubungkan dengan penyambung yang disebut dengan Penyambung-T dan pada ujungnya harus diakhiri dengan sebuah penamat (*terminator*). Penyambung yang digunakan berjenis BNC (*British Naval Connector*: Penyambung Bahari Britania), sebenarnya BNC adalah nama penyambung bukan nama kabelnya, kabel yang digunakan adalah RG 58 (Kabel Sepaksi Thinnet). Pemasangan dari topologi bus beruntut ini sangat sederhana dan murah tetapi sebanyaknya hanya dapat terdiri dari 5-7 komputer.

3. Berdasarkan arsitektur jaringan computer.

a. Jaringan *peer to peer*

Pada jaringan *peer to peer*, semua komputer memiliki posisi setara atau sejajar, dalam hierarki yang sama. Setiap komputer dapat menjadi klien terhadap komputer *peer* lainnya, setiap komputer dapat pula berbagi sumber daya dengan komputer yang berada dalam jaringan *peer-to-peer* ini.

Sumber daya diletakkan secara desentralisasi pada setiap anggota jaringan, dan tidak memerlukan *administrator* jaringan. Aliran informasi bisa mengalir di antara dua komputer secara langsung, di mana pun. Namun, jaringan ini tidak sepenuhnya bebas tanpa kontrol, masih bisa digunakan *password* untuk memproteksi *file* dan folder, dapat juga diatur agar orang-orang tertentu tidak bisa menggunakan *periferal* tertentu. Karena kemudahan pemasangan, pemeliharaan, serta biaya, jaringan ini lebih populer untuk jaringan dengan jumlah komputer yang sedikit (sekitar 2 sampai 20 komputer).

b. Jaringan *client* atau *server*

Pada jaringan *client* atau *server*, perangkat lunak yang mengontrol keseluruhan kerja jaringan berada pada *server*. Jaringan ini dapat menghubungkan ratusan komputer dengan tingkat keamanan yang tidak dimungkinkan dalam jaringan *peer-to-peer*. Jaringan ini bisa diatur sehingga setiap klien harus log on ke *server* sebelum bisa memanfaatkan sumber daya yang terhubung ke *server*. *Server* lalu mengotentikasi klien dan memverifikasi bahwa komputer yang digunakan klien tersebut memiliki izin untuk *log on* ke jaringan, dengan memeriksa username dan *password* klien tersebut terhadap database pada *server*.

c. Jaringan *hybrid*

Jaringan ini merupakan gabungan dari sifat pada jaringan *peer to peer* dan *client* atau *server*. *Workgroup* yang terdiri dari beberapa komputer yang saling terhubung dapat mengelola sumber daya tanpa membutuhkan

otorisasi dari *administrator* jaringan atau *server*. Pada jenis jaringan ini, terdapat pula sifat dari jaringan *client* atau *server* sedemikian sehingga tingkat keamanan dapat lebih terjaga dan adanya *server* yang mempunyai suatu fungsi layanan tertentu, seperti sebagai *file server*, *print server*, *database server*, *mail server*, dan lainnya. Sifat jaringan *peer to peer* digunakan untuk hubungan antara setiap komputer yang terhubung dalam jaringan komputer yang ada, sehingga komunikasi data terjadi antar komputer dengan hierarki yang sama karena setiap komputer dapat berfungsi sebagai *server* maupun klien. Sedangkan, sifat jaringan *client* atau *server* digunakan untuk memfasilitasi setiap komputer dengan hubungan internet. Jadi, komputer *server* dihubungkan ke *Internet Service Provider* (ISP), sehingga komputer klien yang terhubung dalam jaringan juga dapat melakukan akses internet.

2.6 Diagram *Use case*

Use Case merupakan sebuah teknik yang digunakan dalam pengembangan sebuah *software* atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, *Use Case* menjelaskan interaksi yang terjadi antara 'aktor' – inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah *Use Case* direpresentasikan dengan urutan langkah yang sederhana.

Aktor adalah sesuatu atau seseorang yang ada di luar sistem dan ikut berperan serta dalam aktifitas sistem. Aktor bisa berupa: *End User*, perangkat *hardware*, bahkan sistem yang lain. Setiap *use case* merupakan sebuah seri yang

lengkap dari sebuah *event* kejadian, dilihat dari sudut pandang aktor. fokus dari sebuah *use case* adalah menjelaskan bagaimana mencapai sebuah tujuan atau *goal*. Dalam pengembangan *software*, dibutuhkan banyak *use case* untuk mendefinisikan *scope* dari sebuah *system*. derajat formalitas dari sebuah sistem yang dikembangkan menentukan *level detail* yang dibutuhkan dari sebuah *use case*.

Agar kita mengerti mengenai *Use Case* diagram, maka pertama-tama kita harus mengerti dulu elemen-elemen yang ada di dalamnya, elemen-elemennya yaitu :

1. *Actor*:

Elemen aktor diumpamakan adalah sebuah peran dari '*user*' sebuah sistem. aktor ini bisa berupa 'human *user*' atau bahkan bisa sebuah sistem yang lain. dalam diagram *use case*, elemen aktor dinotasikan dengan notasi. karakteristik dari sebuah aktor antara lain:

1. Aktor merupakan elemen eksternal dari sistem
2. Aktor berinteraksi dengan sistem, aktor boleh menggunakan fungsionalitas dan menerima informasi yang disediakan oleh sistem dan aktor juga dapat memberikan informasi kepada sistem.
3. Aktor dapat berbentuk *class*, maka aktor bisa memiliki *instance* atau objek yang merepresentasikan aktor yang lebih spesifik.

2. *Use Case*:

Class Use Case digunakan untuk merepresentasikan unit fungsionalitas atau pelayanan yang diberikan oleh sebuah sistem / bagian sistem, *Use Case* dinotasikan dengan simbol *elips* atau *oval*. dalam diagram, *Use Case* digambar

didalam simbol kotak yang merepresentasikan sistem. *Use Case* memiliki karakteristik:

1. *Use Case* merupakan interaksi atau ‘dialog’ antara sistem dan aktor, termasuk pertukaran pesan dan aksi yang dilakukan oleh sistem.
2. Sebuah *use case* diinisiasikan oleh aktor dan bisa melibatkan lebih dari satu aktor.
3. *Use Case* memiliki *instance* atau objek yang disebut dengan ‘skenario’ yang merepresentasikan interaksi yang spesifik.

2.7 *Diagram Activity*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (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 *actor*.

2.7.1 Kegunaan *Diagram Activity*

Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan

1. urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
2. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

2.7.2 Langkah-langkah Penggambaran

Diagram aktivitas dibaca dari atas ke bawah, mungkin bercabang untuk menunjukkan kondisi, keputusan atau memiliki kegiatan parallel didalamnya.

1. Buat simbol status awal ketika mengawali diagram.
2. Gambarkan aksi pertama dan seterusnya sesuai aliran kegiatan sistem. Gunakan sebuah *fork* ketika berbagai aktivitas terjadi secara bersamaan. Setelah penggabungan seluruh kegiatan paralel, harus digabungkan dengan simbol *join.*
3. Cabang keputusan digunakanuntuk menunjukkan suatu kegiatan yang memenuhi kondisi tertentu. Seluruh pancabangan diakhiri tanda penggabungan (mengganakan tanda *decision*) sebagai akhir perilaku.
4. Akhiri diagram dengan simbol status akhir

2.8 Diagram *Sequence*

Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek - objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.

2.9 Diagram Kelas

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas - kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. atribut merupakan variabel - variabel yang dimiliki oleh suatu kelas
2. operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

2.9.1 Pendefinisian Kelas pada Diagram Kelas

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai
3. Kelas yang diambil dari pendefinisian *use case*

   Kelas yang menangani fungsi - fungsi yang harus ada diambil dari pendefinisian *use case*
4. Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis - jenis kelas di atas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi - fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas

utilitas seperti Koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

2.10 Diagram Objek

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kilas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem objeknya tidak pernah dipakai. Hubungan *link* pada diagram objek merupakan hubungan memakai dan dipakai dimana dua buah objek akan dihubungkan oleh *link* jika ada objek yang dipakai oleh objek lainnya. Sebuah diagram objek merupakan gambaran objek - ojek dalam sebuah sisem pada satu titik waktu. Karena lebih menonjolkan perintah - perintah dari pada kelas. Diagram objek sering disebut juga sebagai sebuah diagram perintah. Elemen - elemen sebuah diagram objek adalah spesifikasi perintah.

2.11 Diagram Komponen

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Pengertian komponen sendiri dalam UML adalah hal - hal fisik dari sistem yang akan dimodelkan da nada ketika sistem dieksekusi. Diantara contoh komponen dasar pada sebuah sistem yaitu :

1) Komponen *user interface* yang menangani tampilan

2) Komponen *business processing* yang menangani fungsi - fungsi proses bisnis
3) Komponen data yang menangani fungsi - fungsi proses bisnis
4) Komponen *security* yang menangani keamanan sistem

2.12 Diagram *Deployment*

Diagram *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node* dan *hardware.*
2. Sistem *client/server*
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

Diagram *deployment* mewakili pandangan pengembangan sistem sehingga hanya akan ada satu diagram *deployment* untuk satu sistem. Diagram *deployment* terdiri dari node dan node merupakan perangkat keras fisik yang digunakan untuk menyebarkan aplikasi. Diagram *deployment* banyak digunakan oleh *system engineer*. Tiap node pada diagram *deployment* mewakili satu unit komputasi sistem yang dalam banyak hal merupakan bagian dari perangkat keras. Diagram *deployment* umumnya memiliki *node* dan hubungan kebergantungan. Memungkinkan juga dalam diagram *deployment* terdapat komponen.