

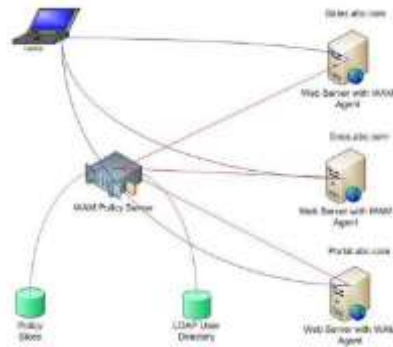
BAB II

TINJAUAN PUSTAKA

II.1. Sistem *Single Sign-On / Single Sign Out*

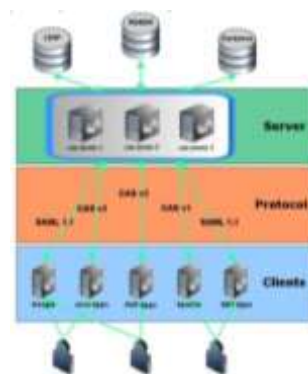
Pengguna di era ini, akan duduk di tempat kerjanya untuk *login* ke desktop, *login* ke emailnya, dan mungkin *login* ke banyak halaman web internal, *login* dan *login* lagi. Pengguna harus menghafalkan *username* dan *password* dari semua aplikasi tersebut, karena bila terjadi kelupaan, dapat menjadi mimpi buruk bagi pengguna. Akibat dari itu muncullah fungsi reset *password* atau *lost password*. Masalah banyaknya akun seringkali diatasi dengan diimplementasikan fasilitas direktori terpusat, seperti LDAP atau *Active Directory* (AD). Dengan adanya ini menyederhanakan banyaknya *username* dan *password* pada sistem berbasis web. Akan tetapi pengguna tetap harus *login* atau *logout* pada setiap aplikasi. Untuk mengatasi hal ini digunakanlah Sistem *Web Access Management* (WAM) atau yang biasa dikenal dengan nama Sistem *Single Sign-On/Single SignOut* (SSO). (Muhammad Yanuar Ary Saputro;2013:2)

Dengan adanya SSO, kita hanya perlu satu kali *login* atau *logout* untuk semua aplikasi berbasis web. Pengimplementasikan WAM atau SSO, memerlukan beberapa komponen yang perlu diperhatikan, yaitu *policy server*, *policy store* dan *web agent* yang mana terletak pada server web untuk mengontrol hak akses.



Gambar II.1. Sistem Single Sign On
 (Sumber : Muhammad Yanuar Ary Saputro;2013:2)

Arsitektur CAS dapat digambarkan dengan sistem komponen termasuk server, klien yang berkomunikasi melalui beberapa jenis protocol. CAS server dan klien CAS merupakan dua komponen sistem arsitektur CAS yang mengkomunikasi melalui berbagai protocol (Muhammad Yanuar Ary Saputro;2013:3)



Gambar II.2. Arsitektur CAS
 (Sumber : Addison dkk;2012:12)

Otentifikasi yang terpusat pada mesin yang unik, inilah yang disebut CAS Server. Mesin ini memiliki 2 peran utama, yaitu :

1. Proses otentikasi pengguna
2. Mengirim sertifikasi pengguna yang telah terotentifikasi (ke klien CAS)

Aplikasi CAS Server merupakan aplikasi Java *servlet* yang mana dibangun dengan Spring Framework yang memiliki fungsi utama untuk otentikasi pengguna

dan memberikan akses ke klien CAS dengan memvalidasi tiket. *Session SSO* dibuat saat tiket divalidasi ke pengguna pada kesuksesan *login*. *Service Ticket* (ST) diberikan ke pengguna melalui browser yang diteruskan menggunakan TGC sebagai *token*. (Muhammad Yanuar Ary Saputro;2013:3)

II.2. Pengenalan Jaringan

Jaringan komputer adalah sistem yang terdiri dari komputer-komputer, serta piranti-piranti yang saling terhubung sebagai satu kesatuan. Dengan dihubungkan pirang-piranti tersebut, alhasil dapat saling berbagi sumber daya antar satu piranti dengan piranti lainnya (Wahana Komputer;2012;2).

Dengan semakin berkembangnya kebutuhan pengolahan data dan informasi, di dalam sebuah perusahaan dibutuhkan beberapa komputer yang digunakan oleh banyak orang yang bekerja dalam sebuah tim. Untuk saling bertukar data dan informasi, maka komputer-komputer yang digunakan akan terhubung antara satu dengan yang lainnya. Kumpulan komputer yang saling terhubung disebut sebagai jaringan komputer. Keuntungan yang didapat dengan menggunakan jaringan komputer diantaranya adalah :

1. Dapat mengakses data di komputer lain dari komputer yang Anda gunakan.
2. Data yang digunakan dapat disimpan atau dicopy ke beberapa komputer, sehingga bila salah satu komputer rusak, maka salinan di komputer yang lain masih dapat Anda gunakan.
3. Penggunaan *Printer*, *Scanner*, CD/DVD ROM dan perangkat lainnya dapat digunakan bersama-sama dengan penggunaan lain.

4. Administrator jaringan dapat mengontrol data-data penting agar dapat diakses oleh pengguna yang berhak saja. Sehingga keamanan data akan lebih terjamin.
5. Penghematan biaya dapat dilakukan, karena sebuah perangkat dapat digunakan secara bersama-sama (Wahana Komputer;2012;3).

II.2.1. Jenis Jaringan

Berdasarkan jangkauan area atau lokasi, jaringan dibedakan menjadi 3 jenis yaitu :

1. *Local Area Network* (LAN) merupakan jaringan yang menghubungkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. LAN dapat menggunakan media komunikasi seperti kabel dan *wireless*.
2. *Wide Area Network* (WAN) merupakan jaringan antara LAN satu dengan LAN lain yang dipisahkan oleh lokasi yang cukup jauh. Contoh penggunaan WAN adalah hubungan antara kantor pusat dengan kantor cabang yang ada di daerah-daerah.
3. *Metropolitan Area Network* (MAN) merupakan jaringan yang lebih besar dari jaringan LAN tetapi lebih kecil dari jaringan WAN. Jaringan MAN dan jaringan WAN sama-sama menghubungkan beberapa LAN yang membedakan hanya lingkup areanya yang berbeda (Wahana Komputer;2012;6)

II.2.2. Jaringan *Client Server*

Jaringan *client server* menghubungkan komputer *server* dengan komputer *client/workstation*. Komputer *Server* adalah komputer yang menyediakan fasilitas bagi komputer-komputer *client/workstation* yang terhubung dalam jaringan.

Sedangkan komputer adalah komputer yang menggunakan fasilitas yang disediakan oleh komputer *server*. Komputer *server* pada sebuah jaringan tipe *client server* disebut dengan *dedicated server*, karena komputer yang digunakan hanya sebagai penyedia fasilitas untuk komputer *client/workstation*. Komputer *server* tidak dapat berperan sebagai komputer *client/workstation*. Keunggulan tipe jaringan *client server* adalah (Wahana Komputer;2012;15) :

1. Terdapat Administrator jaringan yang mengelola sistem keamanan, sehingga sistem keamanan dan administrasi jaringan akan lebih terkontrol.
2. Komputer *server* difungsikan sebagai pusat data, komputer *client* dapat mengakses data yang ada dari komputer *client* manapun. Apabila terdapat komputer *client* yang rusak, pengguna masih dapat mengakses data dari komputer *client* yang lain.
3. Pengaksesan data lebih tinggi karena penyediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer *server*. Dan komputer *server* tidak terbebani dengan tugas lain sebagai workstation.
4. Pada tipe jaringan *client server*, sistem backup data lebih baik, karena backup data dapat dilakukan terpusat di komputer server. Apabila data pada komputer *client/workstation* mengalami masalah atau kerusakan masih tersedia backup pada komputer server (Wahana Komputer;2012;16).

Kelemahan tipe jaringan *client server* adalah :

1. Biaya mahal, karena membutuhkan komputer yang memiliki kemampuan tinggi yang difungsikan sebagai komputer server.

2. Kelancaran jaringan tergantung pada komputer server. Bila komputer server mengalami gangguan maka jaringan akan terganggu (Wahana Komputer;2012;17).

II.3. IP Address

IP (*Internet Protocol*) address merupakan alamat yang diberikan kepada komputer-komputer yang terhubung dalam suatu jaringan. *IP Address* terdiri dari dua bagian, yaitu : *Network ID* dan *Host ID*. *Network ID* menentukan alamat dalam jaringan (*network address*), sedangkan *Host ID* menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin dengan mesin yang lain. Ibarat sebuah alamat rumah, *network ID* seperti alamat rumah dan *Host ID* seperti nomor rumah (Madcoms;2013;10).

IP Address berdasarkan perkembangan dibagi menjadi dua jenis :

1. IPv4 (*internet protocol* versi 4), merupakan *IP Address* yang terdiri dari 32 bit yang dibagi menjadi 4 segmen berukuran 8 *bit*.
2. IPv6 (*internet protocol* versi 6), merupakan *IP Address* yang terdiri dari 128 bit yang digunakan untuk mengatasi permintaan *IP Address* yang semakin meningkat.

IP Address terdiri dari 32 *bit* angka *biner* yang dituliskan dalam bentuk empat kelompok dan masing-masing kelompok terdiri dari delapan (*oktat*) *bit* yang dipisahkan oleh tanda titik.

Contoh : 11000000.101010000.00000000.01100100

IP Address dapat juga ditulis dalam bentuk angka decimal dalam empat kelompok, dari angka 0-255. *IP Address* dibedakan menjadi 3 kelas, yaitu kelas

A, kelas B, dan kelas C. Tujuan membedakan kelas IP adalah untuk menentukan jumlah komputer yang bisa terhubung dalam sebuah jaringan (Madcoms;2013;15).

1. IP kelas A terdiri dari 8 *bit* pertama digunakan untuk *network ID*, dan 24 *bit* berikutnya merupakan *host id*. IP kelas A memiliki 126 *network*, yakni dari nomor 1.xxx.xxx.xxx sampai dengan 126.xxx.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP kelas A :

ONNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 0
- b. *Network ID* adalah 8 *bit* dan *Host ID* adalah 24 *bit*
- c. *Bit* pertama diisi antara 0 sampai dengan 127
- d. Range IP antara 1.xxx.xxx.xxx – 126.xxx.xxx.xxx
- e. Jumlah *Network* adalah 126 (untuk 0 dan 127 dicadangkan)
- f. Jumlah *Host* adalah 16.777.214
- g. Sebagai contoh *IP address* 10.11.22.33, maka *Network ID* adalah 10 dan *Host ID* adalah 11.22.33.

Jadi IP di atas mempunyai *host* dengan nomor 11.22.33 pada jaringan 10

2. IP kelas B terdiri dari 16 *bit* pertama digunakan untuk *Network ID*, dan 16 *bit* berikutnya merupakan *Host ID*. IP kelas B memiliki 16.384 *Network*, yakni dari nomor 128.0.xxx.xxx sampai dengan 191.255.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP Kelas B

10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 10
- b. *Network ID* adalah 16 *bit* dan *Host ID* adalah 16 *bit*
- c. *Bit* pertama diisi antara 128 sampai dengan 191
- d. Range IP antara 128.0.xxx.xxx – 191.255.xxx.xxx
- e. Jumlah *Network* adalah 16.384 (64 x 256)
- f. Jumlah *Host* adalah 65.532
- g. Sebagai contoh *IP address* 130.1.2.3, maka *Network ID* adalah 130.1 dan *Host ID* adalah 2,3

Jadi IP di atas mempunyai host dengann nomor 2,3 pada jaringan 130.1

3. IP kelas C terdiri dari 24 *bit* pertama digunakan untuk *Network ID*, dan 8 *bit* berikutnya merupakan *Host ID*. IP kelas C memiliki 2.097.152 *Network*, yakni dari nomor 192.0.0.xxx sampai dengan 223.255.255.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP kelas C

110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 110
- b. *Network ID* adalah 24 *bit* dan *Host ID* adalah 8 *bit*
- c. *Bit* pertama diisi antara 192 sampai dengan 223
- d. Range IP antara 192.0.0.xxx – 223.255.255.xxx

- e. Jumlah *Network* adalah 2.097.152 (32 x 256 x 256)
- f. Jumlah *Host* adalah 254

Sebagai contoh *IP address* 192.168.0.100, maka *Network ID* adalah 192.168.0 dan *Host ID* adalah 100. Jadi IP di atas mempunyai host dengan nomor 100 pada jaringan 192.168.0 (Madcoms;2013;20)

1. *IP Address Private*, merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan lokal (LAN). *IP Address Private* digunakan untuk komunikasi pada jaringan yang tidak terhubung langsung dengan Internet. *IP Address Private* hanya dapat dipakai untuk komunikasi pada jaringan *intranet* dan tidak dapat digunakan pada jaringan *internet*.
 - a. *IP Address Private* kelas A, memiliki range *IP Address* antara 10.0.0.1 – 10.255.255.254
 - b. *IP Address Private* kelas B, memiliki range *IP Address* antara 172.16.0.1 – 172.31.255.254
 - c. *IP Address Private* kelas C, memiliki range *IP Address* antara 192.168.0.1 – 192.168.255.254
4. *IP Address Public*, merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan *Internet*.
5. *Subnet Mask*, merupakan angka biner 32 *bit* yang digunakan untuk membedakan *Network ID* dan *Host ID*. *Subnet Mask* menunjukkan letak suatu *host*, apakah berada di jaringan lokal atau berada di jaringan luar. Pada *subnet mask*, *bit* yang berhubungan dengan *Network ID* diset 1 sedangkan *bit* yang berhubungan dengan *Host ID* diset 0

a. *IP Address* kelas A menggunakan *subnet mask*

$$11111111.00000000.00000000.00000000 = 255.0.0.0$$

b. *IP Address* kelas B menggunakan *subnet mask*

$$11111111.11111111.00000000.00000000 = 255.255.0.0$$

c. *IP Address* kelas C menggunakan *subnet mask*

$$11111111.11111111.11111111.00000000 = 255.255.255.0$$

II.4. Kabel *Twisted Pair*

Jenis kabel *twisted pair* ini paling murah dan sederhana dalam penggunaannya karena tidak memerlukan keahlian dan peralatan khusus dalam pemasangannya. Kabel ini biasanya dipakai pada jaringan telepon. Terdiri dari dua kabel tembaga atau lebih diisolasi dan dipilin serta biasanya beberapa pasang kabel dijadikan satu tanpa saling mengganggu. Terdapat dua jenis kabel *twisted pair*, yaitu *shielded twisted pair (STP)* dan *unshielded twisted pair (UTP)*. (Kodrat; 2012 : 2).



Gambar II.3. Kabel UTP dan Konektor RJ-45

(Sumber : Kodrat ; 2012 : 2)

Kabel *shielded twisted pair (STP)* digunakan apabila jaringan yang akan dibangun berada atau melewati kondisi lingkungan yang buruk (*extreme*), namun apabila kondisi ruangan dalam keadaan normal sebaiknya cukup menggunakan kabel jenis *unshielded twisted pair (UTP)*. Kabel UTP yang digunakan sebaiknya

berjenis Category 5 yang mendukung transmisi data sampai dengan kecepatan 100 Mbps. Pada praktikum ini digunakan kabel UTP category 5. (Kodrat ; 2012 : 3).

Jika ingin membuat jaringan dengan 2 komputer dan tidak ada rencana untuk menambahkan komputer lagi ke dalam jaringan di masa depan, cara paling mudah adalah dengan menggunakan satu kabel UTP dan sepasang RJ-45 yang di konfigurasi secara *cross*.

II.5. Pengertian PHP

PHP singkatan dari PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web *server side* yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server (server side HTML embended scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamsi. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua script PHP dieksekusi pada *server* di mana script tersebut dijalankan. (Anhar ; 2012 : 3).

II.5.1. Alasan Mempelajari PHP

Beberapa alasan bagi yang mempelajari PHP adalah karena : (Anhar ; 2012 : 3).

1. Kesederhanaan.

User yang baru belajar pemrograman, alasan ini pasti merupakan alasan utama untuk mulai belajar PHP. Karena kesederhanaan tersebut, maka menjadi merasa mudah untuk belajar PHP. User yang sedikit tahu atau bahkan sama

sekali tidak mengerti tentang pemrograman PHP bisa dengan cepat belajar dan mencoba membuat aplikasi web PHP. Selain itu, PHP memiliki banyak sekali fungsi built in untuk menangani kebutuhan standar pembuatan aplikasi web.

2. Dalam sisi pemahaman.

PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

3. PHP adalah bahasa open source yang dapat digunakan di berbagai sistem operasi seperti linux, unix, macintosh, dan windows. PHP dapat dijalankan secara *runtime* melalui *console* serta dapat menjalankan perintah-perintah sistem. Open source artinya code-code PHP terbuka untuk umum dan tidak harus membayar biaya pembelian atas keaslian *license* yang biasanya cukup mahal.

4. *Web Server* yang mendukung PHP dapat ditemukan dimana-man, mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.

5. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti support langsung ke berbagai macam *database* yang populer, misalnya Oracle, PostgreSQL, MySQL dan lain-lain. (Anhar ; 2012 : 4).

II.5.2. PHP dan Internet

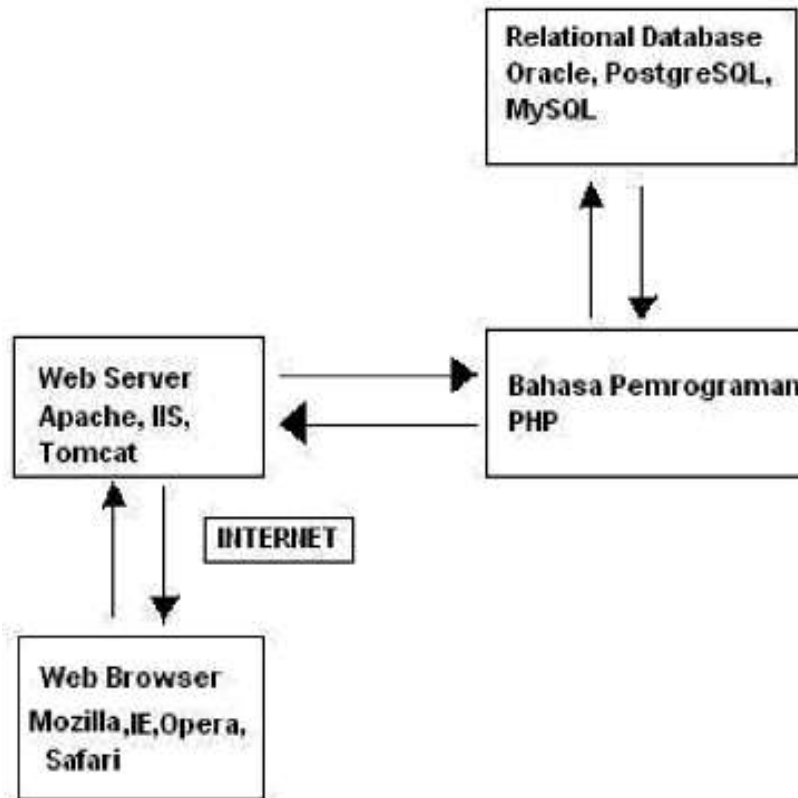
Sebenarnya saat berinternet menggunakan *browser* seperti *Mozilla*, *Internet Explorer*, **Opera**, dan *Safari*, halaman yang muncul di depan layar komputer adalah halaman yang diperoleh dari proses pemanggilan dengan menuliskan alamat sesuai nama *file* yang terdapat pada *web browser*.



Gambar II.9. Mengetikkan Alamat URL
(Sumber: Anhar, 2012: 4)

Gambar II.7. hanya menerangkan tentang alamat URL yang dimasukkan ke bagian aplikasi *mozilla firefox* pada peralatan di dalam aplikasi tersebut yang bernama *tool address*. Tampilan hasil dari URL tersebut adalah aplikasi web facebook dengan alamat URL <http://www.facebook.com>.

Web server adalah aplikasi yang berfungsi untuk melayani permintaan pemanggilan alamat dari pengguna melalui web browser, di mana *web server* mengirimkan kembali informasi yang diminta tersebut melalui HTTP (*Hipertext Transfer Protocol*) untuk ditampilkan ke layar monitor komputer. Agar dapat mengubah isi dari website yang dibuat, harus membutuhkan program PHP. *Script-script* PHP tersebut yang berfungsi membuat halaman website menjadi dinamis. Dinamis artinya pengunjung web dapat memberikan komentar saran. *Website* yang dibuat menjadi lebih hidup karena ada komunikasi antara pengunjung dan sebagai webmasternya. (Anhar, 2012: 5)



Gambar II.10. Bagan dari Aplikasi Web
(Sumber: Anhar, 2012: 5)

Dari gambar diatas terlihat adanya hubungan antara web browser, web server, bahasa pemrograman PHP, dan Database. Secara teknis, Database menyimpan data yang diolah oleh bahasa pemrograman PHP yang ada di web server kemudian data tersebut ditampilkan dengan web browser dengan cara menuliskan alamat URL sesuai nama file yang terdapat pada web server.

II.6. MySQL

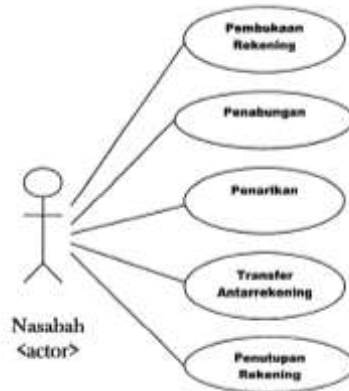
MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain.

MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu MySQL AB. MySQL AB memegang hak cipta kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larson, dan Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau *open source* sehingga bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau support secara gratis. Pemrograman PHP juga sangat mendukung / support dengan *database* MySQL sehingga apabila mempelajarinya dengan sungguh-sungguh dapat mengaplikasikan PHP dan MySQL dalam membuat aplikasi *website* maupun dalam membuat *website*. (Anhar ; 2012 : 21).

II.7. UML

Pemodelan perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan, ia biasanya membuat denah-denang yang menggambarkan bentuk jadi dari bangunan. Sebagai seorang perancang sistem perangkat lunak juga bertindak dengan cara yang serupa, hanya saja yang dirancang bukan bangunan, melainkan sistem perangkat lunak. Menggambarkan komponen sistem perangkat lunak dalam bentuk-bentuk geometri tertentu dalam aplikasi (Adi Nugroho; 2009:6).

Membuat use case diagram yang komprehensif merupakan hal yang sangat penting dilakukan pada tahap analisis. Dengan menggunakan use case diagram, kita akan mendapatkan banyak informasi yang sangat penting yang berkaitan dengan aturan-aturan bisnis yang coba kita tangkap. Dalam hal ini, setiap objek yang berinteraksi dengan sistem perangkat lunak misalnya, orang, suatu perangkat keras, sistem lain, dan sebagainya merupakan actor untuk sistem perangkat lunak, sementara use case merupakan deskripsi lengkap tentang bagaimana sistem perangkat lunak berperilaku untuk para actornya. Dengan demikian, use case diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para actor dengan sistem perangkat lunak yang sedang kita kembangkan. Untuk lebih jelas dapat dilihat pada Gambar II.11.



Gambar II.11. Diagram Use Case
(Sumber : Adi Nugroho; 2012)

Actor pada dasarnya ditentukan berdasarkan perannya (role) pada program /aplikasi yang sedang kita kembangkan, bukan sebagai objek-objek secara mandiri. Sebagai contoh, jika mengambil kasus ATM (Anjungan Tunai Mandiri), seseorang (objek tunggal) mungkin bisa dikelompokkan sebagai actor Karyawan

Bank serta Nasabah jika orang tersebut merupakan karyawan bank yang bersangkutan sekaligus sebagai nasabah karena memiliki tabungan di bank tersebut. Sementara itu, Adi, Ana Geuis, dan beberapa orang lainnya dapat dikelompokkan menjadi *actor* nasabah jika mereka semua masing-masing memiliki tabungan di bank tersebut. Dalam hal ini, kita akan coba mengambil contoh *actor* nasabah untuk menentukan *use casenya*.

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek, karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*. UML juga memberikan standar penulisan sebuah sistem blueprint, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML terdiri dari beberapa diagram, yaitu *use case diagram*, *class diagram*, *state diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram*.

II.8. Perancangan

Untuk membuat tampilan yang menarik memang tidak mudah dilakukan. Seorang perancang tampilan selain harus mempunyai jiwa seni yang memadai, ia juga harus mengerti selera pengguna secara umum. Hal lain yang perlu disadari oleh seorang perancang tampilan adalah bahwa ia harus bisa meyakinkan

pemrogramnya bahwa apa yang ia bayangkan dapat diwujudkan dengan peranti bantu yang tersedia (Insap Santoso ; 2010 : 185).

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan-batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka. (Y. Yohakim Marwanta ; 2010 : 5).

Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah mendokumentasikan semua pekerjaan yang dilakukan. Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara jendela.
4. Menggunakan peranti bantu CASE (*Computer Aided Software Engineering*).

Cara kedua dan keempat tidak selalu dapat diterapkan, karena peranti tersebut biasanya harus dibeli dan seringkali cukup mahal. Cara ini kebanyakan diterapkan pada pembuatan antarmuka grafis untuk suatu jenis pekerjaan berskala besar.

II.8.1. Cara Pendekatan

Sebuah program aplikasi pastilah ditujukan kepada pengguna, yang utama, bukan perancangan program aplikasi tersebut. Program aplikasi pada dasarnya dapat dikelompokkan dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula dan program aplikasi yang

akan digunakan oleh pengguna umum, yang juga sering dikenal dengan sebutan public software. Karena perbedaan pada calon pengguna, maka perancang program antarmuka perlu memperhatikan hal ini (Insap Santoso ; 2009 : 186).

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan pendekatan perancangan berpusat ke pengguna (*user centered design approach*). Cara pendekatan ini berbeda pendekatan perancangan oleh pengguna (*user design approach*).

Pendekatan perancangan berpusat ke pengguna adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna di sini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancangan antarmuka sedang menggambar wajah antarmuka yang mereka inginkan. Dengan kata lain, perancangan dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya. Sementara perancangan menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh

pengguna, seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan (Insap Santoso ; 2010 : 187).

Pada perancangan oleh pengguna, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapat diimplementasikan dengan mudah, atau bahkan tidak dapat dikerjakan dengan menggunakan peranti bantu yang ada.

Perancang program aplikasi yang dimasukkan dalam kelompok kedua, atau *public software*, perlu menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kepandaian dan karakteristik yang sangat beragam. Di satu sisi keadaan ini dapat ia gunakan untuk memaksa pengguna menggunakan antarmuka yang ia buat, tetapi pada sisi lain pemaksaan itu akan berakibat bahwa program aplikasinya menjadi tidak banyak penggunanya. Satu kunci penting dalam pembuatan modul antarmuka untuk program-program aplikasi pada kelompok ini adalah dengan melakukan *customization*. Dengan *customization* pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.

Salah satu contoh dari adanya kemampuan yang dimiliki oleh sebuah program aplikasi atau sistem operasi yang dapat disesuaikan dengan karakteristik pengguna adalah pengaturan desktop pada OS X versi 10.5 favoritnya, sehingga pengguna dapat mengubahnya sesuai keinginan justru akan membuat mata

pengguna itu sakit, dikarenakan mata harus melakukan akomodasi maksimum terus menerus untuk menyesuaikan dengan warna tampilan yang ada.

Selain cara pendekatan yang dijelaskan di atas, Anda yang terbiasa menulis program-program aplikasi mungkin mempunyai cara khusus untuk berhadapan dengan pengguna. Tetapi perlu Anda ingat bahwa apapun cara yang Anda gunakan, Anda tetap harus mempunyai pedoman bahwa pada akhirnya program itu bukan untuk Anda sendiri, tetapi akan digunakan oleh orang lain. Dengan kata lain, jangan pernah mengabaikan pendapat (calon) pengguna program aplikasi Anda. (Insap Santoso ; 2010 : 188).

II.8.2. Prinsip Dan Petunjuk Perancangan

Antarmuka pengguna secara alamiah terbagi menjadi empat komponen model pengguna, bahasa perintah, umpan balik, dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain (Insap Santoso ; 2010 : 188).

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi. Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang dikerjakan oleh program, bahkan oleh pengguna yang sama sekali tidak mengetahui teknologi komputer. Dengan pertolongan model itu pengguna dapat mengantisipasi pengaruh suatu tindakan yang dilakukan dan dapat memilih strategi yang cocok untuk mengoperasikan program tersebut. Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.

Setelah pengguna mengetahui dan memahami model yang di inginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (command language), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasikan. (Insap Santoso ; 2010 : 189).

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna pengguna yang belum berpengalaman dalam menjalankan program sebuah aplikasi. Umpan balik dapat digunakan untuk member keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut.

Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini perancang harus menampilkan pesan-pesan tersebut seefektif mungkin sehingga mudah dipahami oleh pengguna. Setelah memahami beberapa prinsip dalam perancangan antarmuka pengguna. Pada bagian berikut ini akan diberikan petunjuk singkat tentang perancangan antarmuka yang akan Anda lakukan sebagai seorang perancang tampilan.

II.8.3. Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara atas ke bawah. Proses perancangannya dapat dikerjakan secara bertahap sampai rancangan yang diinginkan terbentuk, yaitu sebagai berikut (Insap Santoso ; 2010 : 190).

1. Pemilihan ragam dialog

Untuk suatu tugas tertentu, pilihlah ragam dialog yang menurut perkiraan cocok untuk tugas tersebut. Ragam dialog dapat dipilih dari sejumlah ragam dialog yang telah dijelaskan pada bab-bab sebelumnya. Pemilihan ragam dialog dipengaruhi oleh karakteristik populasi pengguna, tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah ragam tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung.

2. Perancangan Struktur Dialog

Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna sebaiknya banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik dari diskusi yang terjadi. Pada tahap ini suatu purwarupa dialog seringkali dibuat untuk memberik gambaran yang lebih jelas kepada calon pengguna.

3. Perancangan format pesan

Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus

dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan pengguna tombol.

II.9. Keamanan Data

Keamanan data merupakan bagian dari perkembangan teknologi informasi. Ketika berpikir bahwa data yang dimiliki merupakan data yang sangat penting, semua berusaha untuk melindunginya agar jangan sampai jatuh ke tangan orang yang tidak bertanggung jawab. Tetapi buat sebagian orang, mereka justru tidak mengetahui sepenting apakah data yang mereka miliki. Karena ketidaktahuan tersebut, mereka baru menyadari bahwa data yang mereka miliki sangat penting setelah mengalami kecurian data dan mengalami kerugian. Data di sini bisa bersifat umum tidak terbatas pada data digital saja, tetapi juga seperti data diri (ktp, ijasah, sertifikat, dan lain-lain). Data yang menyangkut informasi pribadi tidak seharusnya diumbar sembarang seperti pada situs blog yang tersedia, situs jejaring pertemanan, *email*, selebaran, fotokopi KTP di buang sembarangan dan lain-lain. (Andik Susilo ; 2010 : 59).

Masalah keamanan merupakan salah satu aspek terpenting dari sebuah sistem informasi. Masalah keamanan sering kurang mendapat perhatian dari para perancang dan pengelola sistem informasi. Masalah keamanan sering berada di urutan setelah tampilan, atau bahkan di urutan terakhir dalam daftar hal-hal yang

dianggap penting. Apabila mengganggu performansi sistem, masalah keamanan sering tidak dipedulikan, bahkan ditiadakan. (Doni Ariyus ; 2010 : 6).

Informasi menentukan hampir setiap elemen dari kehidupan manusia. Informasi sangat penting artinya bagi kehidupan karena tanpa informasi maka hampir semuanya tidak dapat dilakukan dengan baik. Contohnya, jika membeli tiket penerbangan dan membayarnya dengan menggunakan kartu kredit, informasi mengenai diri nantinya disimpan dan dikumpulkan serta digunakan oleh bank dan penerbangan. Demikian juga halnya saat membeli obat di apotik. Harus mendapat resep dari dokter dan memberikan resep tersebut ke pelayan apotik. Resep itu merupakan satu informasi yang disampaikan dokter ke pihak apotik tentang obat yang dibutuhkan.

Kemajuan sistem informasi memberikan banyak keuntungan bagi kehidupan manusia. Meski begitu, aspek negatifnya juga banyak, seperti kejahatan komputer yang mencakup pencurian, penipuan, pemerasan, kompetisi, dan banyak lainnya. Jatuhnya informasi ke pihak lain, misalnya lawan bisnis, dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi milik perusahaan yang hanya boleh diketahui oleh orang-orang tertentu di perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam pengembangan. Algoritma dan teknik yang digunakan untuk menghasilkan produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas tertentu.