

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Authentic**

*Authentic* adalah proses dimana seorang *user* (melalui berbagai macam akses fisik berupa komputer, melalui jaringan, atau melalui remote *access* ) mendapatkan hak akses kepada suatu *entity* (dalam hal ini jaringan suatu *corporate*). Seorang user melakukan *logon* kedalam suatu infrastruktur jaringan dan sistem mengenali *USER ID* dan menerimanya untuk kemudian diberikan akses terhadap *resources* jaringan atau sitem aplikasi sesuai dengan authorisasi yang dia terima.

Ada banyak metode yang bisa digunakan untuk melakukan *user authentication* tetapi kebanyakan sistem operasi dan *server* mempunyai beberapa jenis sistem *authentication* dengan menggunakan *user-name* dan *password*. Kebanyakan sistem ini mempunyai beberapa jenis mekanisme untuk manajemen arsitektur *user-name* dan *password*, seperti *account expiration* (kadaluarsa *account*), *password expiration*, panjang *password* minimum, dan atau kualitas *password* (perpaduan capital, angka, simbol). Metode inilah yang banyak dipakai dalam penerapan akses kepada sebuah sistem di jaringan local maupun internet (Ki Grinsing ; 2009).

## II.2. Encryption

Enkripsi yaitu suatu proses pengaman suatu data yang disembunyikan atau proses konversi data ( *plaintext* ) menjadi bentuk yang tidak dapat dibaca/dimengerti. Enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, namun, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Di pertengahan tahun 1970 enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti Internet, *e-commerce*, jaringan telepon bergerak dan ATM pada bank. Enkripsi dapat digunakan untuk tujuan keamanan, tetapi teknik lain masih diperlukan untuk membuat komunikasi yang aman, terutama untuk memastikan integrasi dan autentikasi dari sebuah pesan. Untuk menampilkan enkripsi dan kebalikannya dekripsi, digunakan algoritma yang biasa disebut *Cipher* dengan menggunakan metode serangkaian langkah yang terdefinisi yang diikuti sebagai prosedur. Alternatif lain ialah *Encipherment*. Informasi yang asli disebut sebagai *plaintext*, dan bentuk yang sudah dienkripsi disebut sebagai *chiphertext*. Pesan *chiphertext* berisi seluruh informasi dari pesan *plaintext*, tetapi tidak dalam format yang didapat dibaca manusia ataupun komputer tanpa menggunakan mekanisme yang tepat untuk melakukan dekripsi. Sedangkan Dekripsi yaitu kebalikan dari proses enkripsi yaitu proses konversi data yang sudah dienkripsi ( *ciphertext* ) kembali menjadi data aslinya ( *Original Plaintext* ) sehingga dapat dibaca atau dimengerti kembali (Harriansya ; 2012)

### II.3. File Manager

Manajemen *file* merupakan proses aplikasi pilihan. Pilihan tersebut antara lain membuat *subfolder*, *copy file* memindahkan *file*, menghapus *file*, menampilkan data yang telah terhapus, dan mengoperasikan penggabungan komputer yang di sambungkan dengan sistem jaringan/*network Neighborhood* yang berupa kesatuan *file-file* program yang ada dalam *hard disk* komputer. Tujuannya agar dapat digunakan sesuai dengan pilihan yang dibutuhkan. Manajemen *file* juga diartikan sebagai pengorganisasian *file* data dan *file* program.

Media penyimpanan data pada manajemen *file* dibagi-bagi dalam kelompok yang dinamakan *folder*. Banyak *file* program yang dibutuhkan pada setiap program aplikasi yang meliputi *my computer*, *my document*, internet explorer, *recycle bin*, *new folder*, dan lain-lain. Untuk dapat melakukan pengelolaan *folder* atau *sub-folder* user tinggal melakukan *double click* pada icon *shortcut*.

Banyak *file* program yang dibutuhkan pada setiap program aplikasi. Disamping itu banyak program aplikasi yang ada di dalam suatu komputer beserta data yang sudah dibuat. Semua itu disimpan dalam satu *hard disk*. Untuk itu diperlukan pengaturan-pengaturan sehingga *user* tidak menemui kesulitan dalam mengelola data tersebut. Dalam pengorganisasian *file* data dan *file* program tersebut, media penyimpan data dibagi-bagi ke dalam kelompok yang dinamakan *folder*.

Dewasa ini pesatnya perkembangan teknologi khususnya dibidang internet membuat aplikasi *file manager* menjadi pilihan dalam memenej berkas atau *file*,

dengan semakin banyaknya penggunaan aplikasi melalui internet maka pengelola berkas secara *online* sangatlah penting. Sebagai salah satu solusi adalah dengan menggunakan *file manager* berbasis *web* yang dapat digunakan untuk mengelola berkas, *folder*, bahkan dapat mengeditnya secara *online* seperti dokumen *Microsoft Word*, *Microsoft Excel* dan lain-lain. *File manager* berbasis *web* dapat diakses menggunakan *browser-browser* yang telah ada saat ini.

#### **II.4. Jaringan Komputer**

Dalam bahasa yang populer dapat dijelaskan bahwa jaringan komputer adalah kumpulan beberapa komputer (dan perangkat lain seperti *printer*, *hub* dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Media perantara ini bisa berupa media kabel ataupun media tanpa kabel (*nirkabel*). Informasi berupa data akan mengalir dari satu komputer keperangkat lain, sehingga masing-masing komputer yang terhubung tersebut bisa saling bertukar data atau berbagi perangkat keras (Iwan Sofana ; 2008 : 3).

#### **II.5. Client Server**

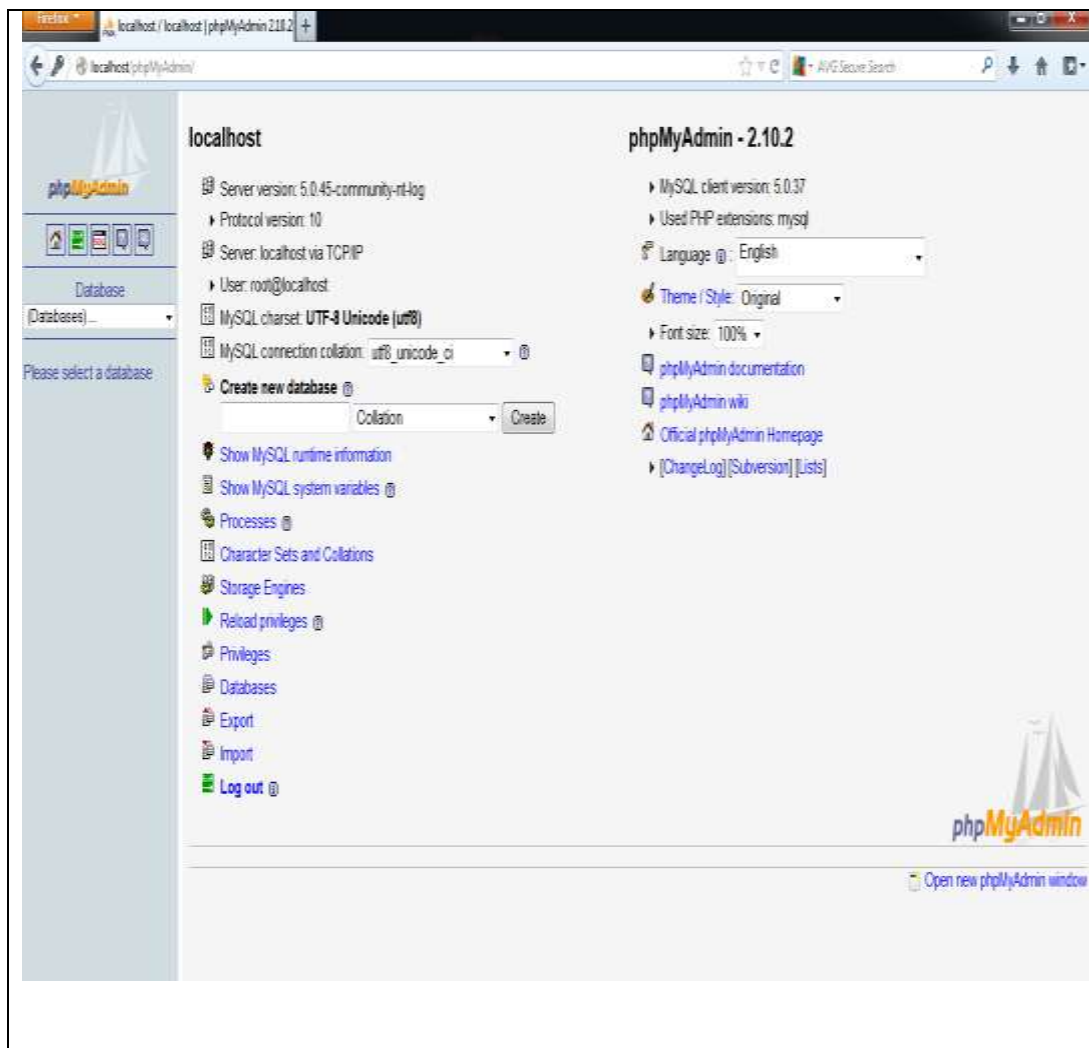
*Client server* adalah jaringan komputer yang salah satu/lebih komputer difungsikan sebagai *server* atau induk bagi komputer lain. *Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *Web*, *e-mail*, *file* atau yang lain. *Client server* banyak dipakai pada *Internet*. Namun *LAN* atau jaringan lain pun bisa mengimplementasikan *client server*. Hal ini sangat tergantung pada kebutuhan masing-masing (Iwan Sofana ; 2008 : 6).

## II.6. Database

*Database* adalah sekumpulan data yang terdiri dari satu atau lebih tabel yang saling berhubungan. Anda atau *user* mempunyai wewenang untuk mengakses data tersebut, baik untuk menambah, mengubah, atau menghapus data yang ada dalam tabel tersebut (Divisi Penelitian dan Pengembangan *MADCOMS* ; 2005 : 2)

## II.7. MySQL

*Software* ini cukup familiar bagi *programer* berbasis *open source* dan para pengembang aplikasi *client-server*, serta *web programming*. Dalam hal ini penulis menggunakan distribusi pengembangan lain yang memanfaatkan *database* dengan bantuan program pendukung lainnya, yaitu *AppServ*. Program bantu ini untuk manajemen basis data *MySQL* dan skrip *PHP* dan dilengkapi *server Apache* (Miftakhul Huda, Cuk Subiyantoro, Bunafit Nugroho ; 2008 : 63,64).



**Gambar II.2. Tampilan PhpMyAdmin**

(Sumber : *Miftakhul Huda, Cuk Subiyantoro, Bunafit Nugroho* ; 2008 : 65)

## II.8. UML

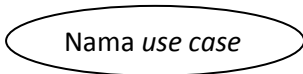
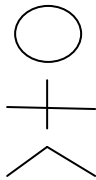

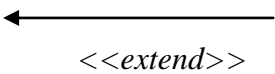
*Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Yuni Sugiarti ; 2013 : 34 )

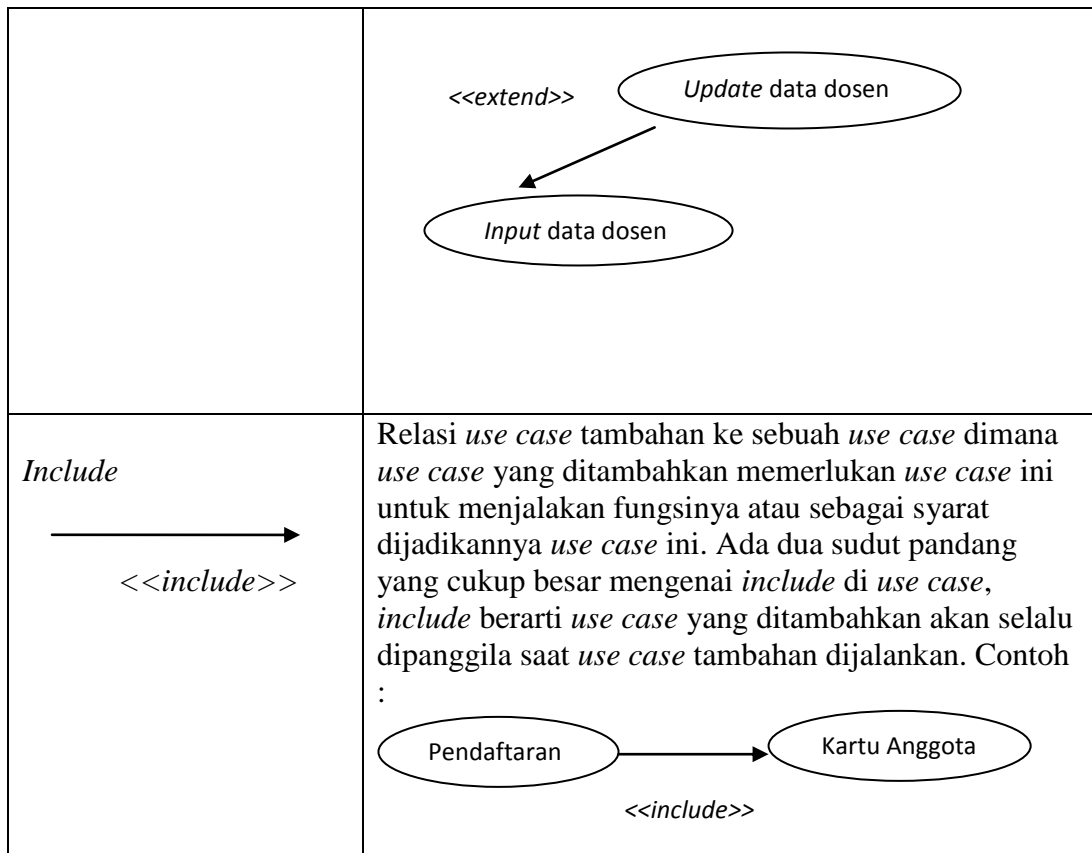
Adapun macam-macam diagram UML diantaranya adalah *Use Case*, *Activity Diagram* dan *Sequence Diagram*.

### II.8.1 Use Case Diagram

*Use case diagram* atau *diagram use case* merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Yuni Sugiarti ; 2013: 41)

**Tabel II.1. Simbol-Simbol *Use Case Diagram***

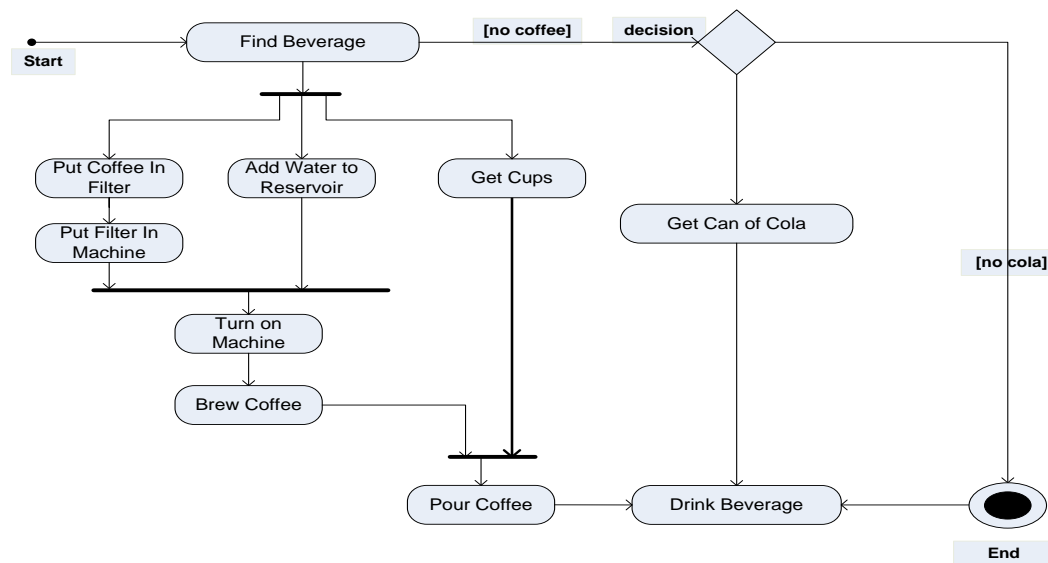
Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor</p>  <p>Nama actor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi/<i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p><i>Extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, arah panah menunjuk pada <i>use case</i> yang dituju. Contoh :</p>



(Sumber : *Yuni Sugiarti* ; 2013 : 42)

## II.8.2 Activity Diagram

Diagram aktifitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan aktor, jadi aktifitas yang dapat dilakukan oleh sistem (*Yuni Sugiarti* ; 2013 : 75).



**Gambar II.3. Contoh Activity Diagram**  
(Sumber : *Yuni Sugiarti* ; 2013 : 76)

### II.8.3 Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *Sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang di instansikan menjadi objek itu (Yuni Sugiarti ; 2013 : 69)