



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Komputer secara umum bisa diartikan kepada kumpulan alat elektronik yang tersusun menjadi rangkaian membentuk sebuah mesin berteknologi dengan kontrol sistem operasi disertai program-program yang mampu menerima dan menyimpan data, melakukan pengolahan dan memberikan hasil dalam bentuk informasi sesuai prosedur operasi yang dirumuskan.

Menurut Hamacher et al (dalam Khairil et al, 2012:20) “Komputer adalah mesin penghitung elektronik yang cepat dan menerima informasi input digital, memprosesnya sesuai dengan suatu program yang tersimpan dimemorinya (stored program) dan menghasilkan output informasi”.

Menurut Dhanta (dalam Islamiyah dan Apriyani AR, 2016:29) “Secara umum, komputer adalah alat yang dipakai untuk mengolah data menurut prosedur yang telah dirumuskan. Sedangkan, pengertian komputer secara luas/modern adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dalam hasil pengolahannya, serta menyediakan *output* dalam bentuk informasi”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah input dengan menggunakan suatu program yang tersimpan didalam memori komputer, memberikan output yang berupa informasi dan dapat menyimpan program serta hasil pengolahan dalam suatu media penyimpanan seperti *hardisk*, *flashdisk*, *cloud* atau penyimpanan lainnya.



---

### 2.1.2. Pengertian Data

Data secara umum adalah kumpulan fakta, kejadian yang dijadikan pemecahan masalah berupa informasi.

Menurut Edhy Sutanta (dalam Sembiring dan Nurhayati, 2012;14) “Data adalah sebagai bahan keterangan tentang kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang tidak acak yang menunjukkan jumlah, tindakan, atau hal. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan sebagai file dalam basis data”.

Menurut Yusi et.al (dalam Islamiyah dan Apriyani AR, 2016:29) “Data adalah kumpulan angka, fakta, fenomena atau keadaan lainnya yang disusun menurut logika tertentu merupakan hasil pengamatan, pengukuran atau pencacahan dan sebagainya terhadap variabel dari suatu objek kajian, yang berfungsi dapat digunakan untuk membedakan objek yang satu dengan lainnya pada variabel yang sama”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas , maka dapat disimpulkan bahwa data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

### 2.1.3. Pengertian Basis Data (*Database*)

Secara umum, basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk suatu bangunan data untuk menginformasikan suatu perusahaan atau instansi dalam batasan tertentu.

Menurut Edhy Sutanta (dalam Sembiring dan Nurhayati, 2012;14) “Basis data (*database*) dapat dipahami sebagai “suatu kumpulan data terhubung (*interrelated* data) yang disimpan secara bersama-sama pada suatu media”, tanpa suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol (*controlled redundancy*), data di disimpan dengan cara tertentu, sehingga mudah untuk digunakan/atau ditampilkan kembali; dapat digunakan oleh satu atau lebih program aplikasi secara optimal”.

Menurut Bambang Hariyanto (dalam Minarni dan Susanti, 2014:105) “*Database* adalah kumpulan data (*elementer*) yang secara *logic* berkaitan dalam



---

mempresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam sistem tertentu”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

#### **2.1.4. Pengertian Informasi**

Secara umum, informasi adalah penerangan, keterangan, pemberitahuan, kabar atau berita. Informasi juga merupakan keterangan atau bahan nyata yang dapat dijadikan dasar kajian analisis atau kesimpulan atau informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi.

Menurut Turban et al (dalam Andalia dan Setiawan, 2015:93) “Informasi merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya. Sedangkan menurut Jogiyanto, Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Dapat dikatakan bahwa data merupakan bahan mentah, sedangkan informasi adalah bahan jadi atau bahan yang telah siap digunakan, Jadi, sumber dari informasi adalah data”.

Menurut Ladjamudin (dalam Hermawan et al, 2016:32) “Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa informasi adalah data yang telah diproses menjadi bentuk yang bernilai bagi penerimanya dan bermanfaat dalam setiap pengambilan keputusan.



### 2.1.5. Pengertian Internet

Secara umum, internet adalah jaringan besar yang saling berhubungan dari jaringan-jaringan komputer yang menghubungkan orang-orang dan komputer-komputer diseluruh dunia, melalui telepon, satelit dan sistem-sistem komunikasi yang lain.

Sujatmiko (dalam Islamiyah dan Apriyani AR, 2016:30) "Internet atau *Interconnected Network* adalah jaringan global yang menghubungkan berjuta-juta komputer di seluruh dunia melalui jalur telepon kabel maupun satelit. Dengan adanya internet, jutaan orang di seluruh duniaberbagi segala jenis informasi".

Menurut Oneto dan Sugiarto (dalam Prayitno dan Safitri 2015:2) "Internet adalah jaringan komputer. Ibarat jalan raya, internet dapat dilalui berbagai sarana transportasi, seperti bus, mobil dan motor yang memiliki kegunaan masing-masing".

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa internet adalah jaringan komputer yang saling terhubung antara satu komputer dengan komputer yang lain yang membentuk sebuah jaringan komputer di seluruh dunia, sehingga dapat saling berinteraksi, berkomunikasi, saling bertukar informasi atau tukar menukar data.

## 2.2. Teori Khusus

### 2.2.1. Unified Modeling Language (UML)

Menurut Rosa dan Shalahuddin (dalam Pratama dan Junianto, 2015:216), UML (*Unified Modeling Language*) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metologi tertentu, meskipun pada kenyataannya UML paling banyak diguakan pada metodologi berorientasi objek.



---

### 2.2.2. Diagram UML

Menurut Rosa A.S dan M. Shalahudin (2014:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

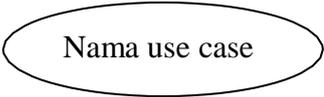
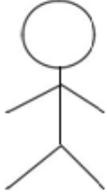
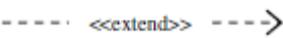
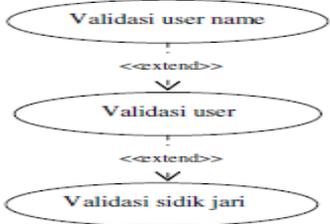
1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

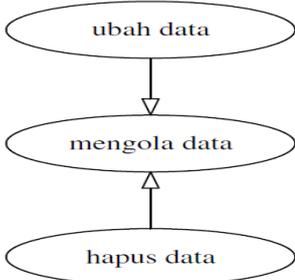
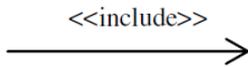
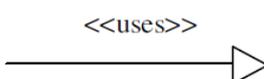
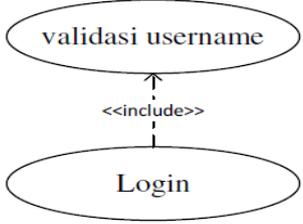
### 2.2.3. Use Case Diagram

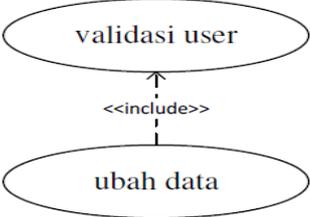
Menurut Rosa dan M. Shalahudin (2014:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

**Tabel 2.1.** Simbol-simbol diagram *Use case*

No.	Simbol	Deskripsi
1.	<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 diawal frase nama <i>use case</i>.</p>
2.	<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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>
3.	<p>Assosiasi/<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>
4.	<p>Exstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <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, misal</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

No.	Simbol	Deskripsi
5.	Generalisasi/ <i>generalization</i>  	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :   arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
6.	Menggunakan / <i>include</i> / <i>uses</i>    	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : - <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut :   - <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut :

No.	Simbol	Deskripsi
		 <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

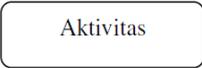
Sumber : Rosa A.S dan M. Shalahudin (2014:156)

#### 2.2.4. Activity Diagram

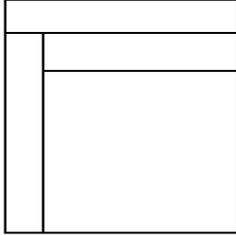
Menurut Rosa dan M. Shalahudin (2014:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2. Simbol-simbol *activity diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.



No.	Simbol	Deskripsi
4.	Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Rosa A.S dan M. Shalahudin (2014:162)

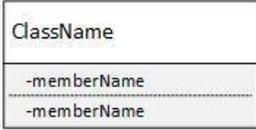
### 2.2.5. Class Diagram

Menurut Rosa dan M. Shalahudin (2014:141), 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 *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

**Tabel 2.3.** Simbol-simbol *class diagram*

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek



No.	Simbol	Deskripsi
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas

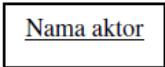
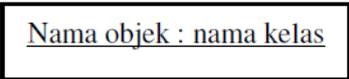
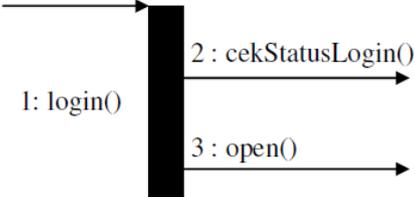
Sumber : Rosa A.S dan M. Shalahudin (2014:146)

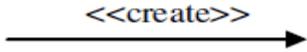
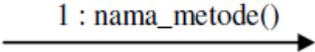
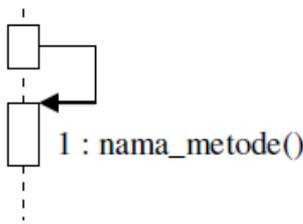
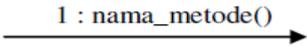
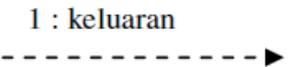
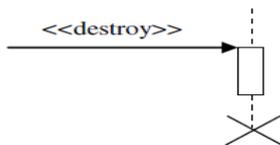
### 2.2.6. Sequence Diagram

Menurut Rosa dan M. Shalahudin (2014:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

**Tabel 2.4.** Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</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 dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

No.	Simbol	Deskripsi
		Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Rosa A.S dan M. Shalahudin (2014:165)



---

## 2.3. Teori Judul

### 2.3.1. Pengertian Aplikasi

Secara Umum, aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju.

Menurut M. Susanto (dalam Marjito dan Tesaria, 2016:42) “Aplikasi biasanya berupa perangkat lunak yang berbentuk software yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan. Selain itu aplikasi juga mempunyai fungsi sebagai pelayan kebutuhan beberapa aktivitas yang dilakukan oleh manusia seperti sistem untuk software jual beli, permainan atau game online, pelayanan masyarakat dan hamper semua proses yang dilakukan oleh manusia dapat dibantu dengan menggunakan aplikasi”.

Menurut Sujatmiko (dalam Islamiyah dan Apriyani AR, 2016:44) “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas–tugas tertentu, misalnya MS-Word, MS-Excel”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa aplikasi adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai

### 2.3.2. Pengertian *Container* (Peti kemas)

Secara umum, petikemas (*container*) adalah merupakan gudang kecil yang berjalan untuk mengangkut barang dari satu tempat ke tempat lain harus bersama-sama alat pengangkutnya yakni, kapal truk atau kereta api sampai ke tempat yang dituju, biasanya ke gudang pemilik barang (*exporter* dan *importer*).

Menurut Suyono (dalam Munir et al 2012:13) “Peti kemas adalah satu kemasan yang dirancang secara khusus dengan ukuran tertentu dapat dipakai berulang kali, digunakan untuk menyimpan dan sekaligus mengangkut muatan yang ada didalamnya”.



Menurut Subandi (dalam Munir et al 2012:13) “Peti kemas adalah *“container is box, bottle, etc, made to contain something”*. Dimana bila diartikan sebagai berikut, peti kemas adalah peti, botol, dan sebagainya yang dibuat untuk dapat menyimpan sesuatu”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa petikemas (container) adalah suatu wadah yang digunakan untuk menyimpan dan sekaligus mengangkut muatan yang ada didalamnya.

### **2.3.3. Pengertian *Location* (Lokasi)**

Secara Umum, lokasi adalah tempat dimana suatu usaha atau aktivitas usaha dilakukan.

Menurut Levy dan Weitz (dalam Afifudin 2016:3) “Lokasi merupakan perencanaan dan pelaksanaan program penyaluran produk atau jasa melalui tempat atau lokasi yang tepat”.

Menurut Buchari Alma (dalam Rizal et al, 2017:81) “Lokasi adalah tempat perusahaan beroperasi atau tempat perusahaan melakukan kegiatan untuk menghasilkan barang dan jasa yang mementingkan segi ekonominya”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa lokasi adalah tempat melakukan kegiatan perencanaan dan pelaksanaan program penyaluran produk atau jasa.

### **2.3.4. Pengertian *Barcode***

Secara umum, *barcode* adalah susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, sangat sederhana tetapi sangat berguna. Dengan kegunaan untuk menyimpan data-data spesifik misalnya kode produksi, tanggal kadaluwarsa, nomor identitas dengan mudah dan murah, walaupun teknologi semacam itu terus berkembang dengan ditemukannya media magnetik, RFID,

Menurut Youllia dan Indrawaty (dalam Subaeki dan Jauhari, 2016:111) “*Barcode* atau kode batang adalah suatu kumpulan data optik yang dibaca mesin. Sebenarnya, kode batang ini mengumpulkan data dalam lebar (garis) dan jarak garis paralel dan dapat disebut sebagai kode batang atau simbologi *linear* atau 1D



(1 dimensi). Tetapi juga memiliki bentuk persegi, titik, heksagon dan bentuk geometri lainnya di dalam gambar yang disebut kode matriks atau simbologi 2D (2 dimensi). Selain tak ada garis, sistem 2D sering juga disebut sebagai kode batang”.

Menurut Yudha Yudhanto (dalam Jamal dan Yulianto, 2013:47) ”Sebuah kode batang (atau *barcode*) adalah suatu kumpulan data optik yang dibaca mesin”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa *barcode* adalah sekumpulan data yang digambarkan dengan garis dan jarak spasi (ruang) yang dapat dibaca mesin.

### 2.3.4.1. Jenis-jenis *Barcode*

#### 2.3.4.1.1. *Linear Code (Barcode 1 Dimensi)*

*Linear code* terdiri atas :

1. *Code 39 (code 3 of 9)* yaitu sebuah *barcode alphanumeric (barcode Full ASCII)* yang mempunyai panjang baris yang bervariasi. Implementasi *barcode* jenis ini biasanya adalah untuk tanda pengenal identitas, *inventory* dan *asset tracking*.
2. *Interleaved 2 of 5* , yaitu sebuah *barcode* berbentuk *numeric* dan mempunyai panjang baris yang bervariasi. *Barcode* jenis ini pada umumnya dipakai untuk industri dan laboratorium.
3. *Code 128*, yaitu suatu *barcode alphanumeric (Full ASCII)* yang mempunyai kerapatan (*density*) sangat tinggi serta dengan panjang baris yang bervariasi. Pemakaian *barcode* jenis ini biasanya untuk *system shipping and, warehouse management*.
4. *UPC (Universal Product Code)* adalah sebuah *barcode numeric* dan memiliki panjang baris yang tetap (*fixed*). *UPC* banyak digunakan untuk pelabelan pada produk-produk berukuran kecil/ritel.



#### 2.3.4.1.2. Matrix Code (Barcode 2 Dimensi)

*Matrix code* (barcode 2 Dimensi) lebih canggih dibandingkan dengan *Linear Code* (barcode 1 Dimensi) hal ini karena *matrix code* dapat memuat ratusan digit karakter dan tampilannya juga berbeda dengan *Linear Code*. Pada *matrix code* (barcode 2 Dimensi), informasi atau data yang besar bisa disimpan dalam ruang (*space*) yang kecil.

Sebuah *barcode* 2D dapat menyimpan sejumlah besar informasi dan mungkin tetap dapat dibaca bahkan ketika dicetak pada ukuran kecil atau terukir ke produk. *Barcode* 2D digunakan dalam berbagai industri, dari manufaktur, pergudangan logistik dan kesehatan. Contoh dari *Barcode* 2D yaitu *QR Code* yang juga disebut "2D Matrix" adalah simbol-simbol yang dapat membawa semua *GS1 keys* dan atribut.

Hal ini dapat diterapkan pada barang perdagangan untuk membawa informasi seperti tanggal kadaluwarsa, nomor seri atau *batch/lot* nomor. Simbol 2D juga dapat digunakan untuk informasi tambahan pada kemasan (*extended information*) yang berisi tautan halaman web (URL). *Barcode* 2D mendukung pemindaian aplikasi ponsel.

##### 2.3.4.1.2.1. QR Code (Quick Response)

*QR Code* (*Quick Response*) adalah *barcode* 2D, yang terdiri dari modul hitam (persegi titik/piksel), diatur dalam pola persegi di latar belakang putih. Kode-kode ini bisa diterjemahkan secara singkat dan cepat hanya menggunakan kamera ponsel yang telah ditambahkan aplikasi pembaca *QR Code*. *QR Code* diciptakan oleh perusahaan Jepang, Denso-Wave dari Denso Corp. pada tahun 1994, yang digunakan dalam industri otomotif.

Karakteristik dari *QR Code* :

1. Alpha Numerik (dapat memuat angka dan huruf).
2. Informasi yang terkandung sangat padat.
3. Pembacaan alat *scanner* menggunakan *scanner* 2D.
4. Kapasitas : 3116 digit angka atau 2335 gabungan angka dan huruf.
5. Mengukung penggunaan kunci identifikasi *GS1*.



**Gambar 2.1.** QR Code (*Quick Response*)

### **2.3.5. Pengertian Scanner (Pemindai)**

Menurut Youllia dan Indrawaty (dalam Subaeki dan Jauhari, 2016:111) “Kode batang dapat dibaca oleh pemindai optik yang disebut pembaca kode batang atau dipindai dari sebuah gambar oleh perangkat lunak khusus. Di Jepang, kebanyakan telepon genggam memiliki perangkat lunak pemindai untuk kode 2D, dan perangkat sejenis tersedia melalui *platform smartphone*”.

### **2.3.6. Pengertian Android**

Android secara umum adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.

Menurut Burnette (dalam M. Ichwan dan Hakiky, 2011:15) “Android merupakan *toolkit* perangkat lunak *open source* baru untuk perangkat mobile masa depan, android sendiri diciptakan oleh perusahaan Google yang bekerja sama dengan *Open Handset Alliance*”.

Menurut Safaat (dalam M. Ichwan dan Hakiky, 2011:15) “Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc.



Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa android adalah sistem operasi yang berjalan pada smatphone yang menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri

### **2.3.7. Pengertian Perseroan Terbatas (PT)**

Kamus Besar Bahasa Indonesia (2012:1288) menyatakan, “Perseroan Terbatas (PT) merupakan perserikatan dagang atau perusahaan yang mempunyai modal bersama yang dibagi atas saham–saham (tanggungjawab pemegang saham terbatas sampai nilai nominal saham yang dimiliki)”.

### **2.3.8. Pengertian PT. Pelabuhan Indonesia II (Persero) Cabang Palembang**

“PT Pelabuhan Indonesia II (Persero) Cabang Palembang atau sering dikenal dengan Pelindo II adalah Badan Usaha Milik Negara yang bergerak dibidang logistik, secara spesifik pada pengelolaan dan pengembangan pelabuhan.” (PT. Pelabuhan Indonesia II (Persero) Cabang Palembang. 2012. <http://www.indonesiaport.co.id/read/tentang-perusahaan.html>, 18 Mei 2018)

### **2.3.9. Pengertian Metode Agile (*Agile Development Methods*)**

“*Agile Development Methods* adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Agile development methods merupakan salah satu dari Metodologi pengembangan perangkat lunak yang digunakan dalam pengembangan perangkat lunak. Agile memiliki pengertian bersifat cepat, ringan, bebas bergerak, dan waspada. Sehingga saat membuat perangkat lunak dengan menggunakan agile development



methods diperlukan inovasi dan responsibiliti yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan kelincuhan dari tim seimbang.” (Mawan. 2017. <http://mawanfki.blogspot.com/2015/03/other-agile-process-model.html>, 18 Mei 2018)

### **2.3.10. Pengertian Aplikasi *Container Location* Menggunakan Barcode Scanner Berbasis Android Pada PT. Pelabuhan Indonesia II (Persero) Cabang Palembang dengan Metode *Agile***

Aplikasi *Container Location* Menggunakan Barcode Scanner Berbasis Android Pada PT. Pelabuhan Indonesia II (Persero) Cabang Palembang dengan Metode *Agile* adalah suatu program aplikasi berbasis android yang digunakan untuk mempermudah proses input data *container* dan mendapatkan informasi lokasi *container* secara akurat di PT. Pelabuhan Indonesia II (Persero) Cabang Palembang dengan menggunakan metodologi pengembangan perangkat lunak *Agile*.

## **2.4. Teori Program**

### **2.4.1. Pengertian Java**

Secara umum, Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman dan sekaligus suatu platform.

Menurut Hendra et al (dalam Rusmana et al, 2014:41) “Java adalah bahasa pemrograman yang dapat dijalankan diberbagai perangkat komputer, termasuk pada ponsel atau dengan kata lain bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi”.

Menurut Abdul Kadir (dalam Afrizal et al, 2017:3) “Java merupakan pemrograman yang bersifat lintas *platform*. Artinya, bahasa ini dapat dipakasi untuk menyusun program pada berbagai sistem operasi (misalnya Linux, Windows, dan UNIX)”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa java adalah bahasa pemrograman yang muti platform



---

dan multi device dapat dijalankan diberbagai perangkat komputer, termasuk pada ponsel.

### **2.4.2. Pengertian Eclipse**

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent).

Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut plug-in.

#### **2.4.2.1. Sejarah Eclipse**

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.



#### 2.4.2.2. Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah dipasang (diinstal). Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP).

Berikut ini adalah komponen yang membentuk RCP:

1. Core platform
2. OSGi
3. SWT (Standard Widget Toolkit)
4. JFace
5. Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE adalah :

1. Terbuka (open),
2. Mudah diperluas (extensible) untuk apa saja, dan
3. Tidak untuk sesuatu yang spesifik.

#### 2.4.2.3. Versi-versi Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse.

Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya.



Tabel 2.5. Versi-versi Eclipse

<i>Code name</i>	<i>Date</i>	<i>Platform Version</i>
N/A	21 June 2004	3.0 <sup>[14]</sup>
N/A	28 June 2005	3.1
Callisto	30 June 2006	3.2
Europa	29 June 2007	3.3
Ganymede	25 June 2008	3.4
Galileo	24 June 2009	3.5
Helios	23 June 2010	3.6
Indigo	22 June 2011	3.7 <sup>[Notes 1]</sup>
Juno	27 June 2012	4.2 <sup>[15]</sup>
Kepler	26 June 2013	4.3
Luna	25 June 2014	4.4

### 2.4.3. Pengertian PHP

Secara umum, PHP adalah Hypertext Preprocessor, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.

Menurut Kadir (2014:2), menurut dokumen resmi PHP, PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya adalah yang dikirimkan ke *client* tempat pemakai menggunakan *browser*.

Menurut Sidik (2016:5), *PHP* adalah kependekan dari *PHP:Hypertext Preprocessor* (rekursif, mengikut gaya penamaan di \*nix), merupakan bahasa utama *script server-side* yang disisipkan pada HTML yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi desktop.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang dijalankan di *server*, dan juga bisa digunakan untuk membuat aplikasi desktop.



---

## 2.4.4. Sekilas Tentang MySQL

### 2.4.4.1. Pengertian MySQL

Secara umum, MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structured Query Language*).

Menurut Anhar (dalam Prayitno dan Safitri, 2015:2) “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 lainnya”.

Menurut Abdul Kadir (dalam Yusmiarti, 2016:4) “MySQL tergolong sebagai DBMS (*Database Management System*). - Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat”.

Dari beberapa definisi yang didapat dari berbagai sumber diatas, maka dapat disimpulkan bahwa MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL *Database Management System* atau DBMS yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structured Query Language*).

### 2.4.4.2. Fitur-fitur MySQL

Sebagai *software* DBMS, MySQL memiliki sejumlah fitur seperti dibawah ini:

1. Multiplatform  
MySQL tersedia pada beberapa platform (Windows, Linux, Unix, dan lain– lain.
2. Andal, cepat, dan mudah digunakan  
MySQL tergolong sebagai *database server* (server yang melayani permintaan terhadap database) yang andal, dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.
3. Jaminan keamanan akses



MySQL mendukung pengamanan database dengan berbagai kriteria pengaksesan.

#### 4. Dukungan SQL

MySQL mendukung perintah SQL (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan standar dalam pengaksesan database relasional.

### 2.4.4.3. Tipe Data

MySQL mendukung banyak tipe data yang terbagi menjadi beberapa kategori, antara lain tipe numeric, tipe date dan time, dan tipe string. Berikut ini akan dibahas tentang kategori tipe data masing–masing tersebut.

#### 2.4.4.3.1. Tipe Numerik

MySQL mendukung hampir semua tipe data numeric/angka, asalkan sesuai dengan standar SQL. Termasuk di dalamnya adalah tipe data untuk bilangan bulat (misalnya, Integer, SmallInt, Decimal, dan Numeric) serta tipe data yang digunakan untuk bilangan desimal atau pecahan (misalnya, Float, Real, dan Double Precision). Kata kunci INT merupakan persamaan untuk Integer dan kata kunci DEC merupakan persamaan untuk Decimal.

Untuk menyesuaikan dengan standar SQL, MySQL juga mendukung tipe *integer* lain, seperti TinyInt, MediumInt, dan BigInt. Berikut ini tabel yang menunjukkan jenis tipe data numeric/angka.

**Tabel 2.6.** Tipe Data Numerik

Type	Byte	Signed	Unsigned
TinyInt	1	-128 s/d 127	0 s/d 255
SmallInt	2	-32768 s/d 32767	0 s/d 65535
MediumInt	3	-8388608 s/d 8388607	0 s/d 16777215
Int, Integer	4	-2147483648 s/d 2147483647	0 s/d 4294967295
BigInt	8	-922337203685475808 s/d 9223372036854775807	0 s/d 18446744073709551615
Float	4	-3.402823466E+38 s/d -1.175494351E-38	



Type	Byte	Signed	Unsigned
		1.175494351E-38 s/d 3.402823466E+38	
Double	8	-1.7976931348623157E+308 s/d 2.2250738585072014 E-308	

#### 2.4.4.3.2. Tipe *Date* dan *Time*

MySQL memiliki tipe data yang berhubungan dengan penanganan tanggal dan pewaktuan, antara lain *Datetime*, *Date*, *Timestamp*, *Time*, dan *Year*. Berikut ini jenis-jenis tipe data yang berhubungan dengan tipe date dan time.

**Tabel 2.7.** Tipe *Date* dan *Time*

Type	Keterangan
Datetime	Ukuran 8 byte. Kombinasi tanggal dan jam dengan jangkauan mulai '1000 - 01 - 01 00:00:00' s/d '9999 - 12 - 31 23:59:59'.
Date	Ukuran 3 byte. Tanggal dengan jangkauan mulai '1000 - 01 - 01' s/d '9999 - 12 - 31'.
Timestamp	Ukuran 4 byte. Kombinasi tanggal dan jam dengan jangkauan mulai dari '1970 - 01 - 01 00:00:00 s/d tahun 2037.
Time	Ukuran 3 byte. Waktu dengan jangkauan mulai dari '- 838:59:59' s/d '838:59:59'.
Year	Ukuran 1 byte. Data tahun antara 1901 s/d 2155.

**Tabel 2.8.** Nilai Kosong pada Tipe *Date* dan *Time*

Type	Jika bernilai kosong
Datetime	'0000 - 00 - 00 00:00:00'
Date	'0000 - 00 - 00'
Timestamp	'0000 - 00 - 00 00:00:00'
Time	'00:00:00'
Year	0000



### 2.4.4.3.3. Tipe Character

Tipe character adalah tipe yang berhubungan dengan string. Adapun jenis tipe datanya antara lain sebagai berikut.

**Tabel 2.9.** Tipe Data Character

Type	Jika bernilai kosong
binary, char	Ukuran M byte, $1 \leq M \leq 255$ . Data string dengan panjang yang tetap.
varbinary, varchar	Ukuran bersifat fleksibel dan maksimal M byte.
Tinyblob, tinytext	Tipe yang mampu mengendalikan data besar, tergantung seberapa besar kapasitas penyimpanannya.
Blob, text	
Mediumblob, mediumtext	
Longblob, longtext	

**Tabel 2.10.** Perbedaan Tipe Character Char dan Varchar

Nilai	Char(4)	Ukuran	Varchar(4)	Ukuran
' '	' '	4 byte	' '	1 byte
'ab'	'ab '	4 byte	'ab '	3 byte
'abcd'	'abcd '	4 byte	'abcd '	5 byte
'abcdefgh'	'abcd '	4 byte	'abcd '	5 byte

### 2.4.4.4. Perintah-perintah SQL Dasar

#### 1. *Select*

Select digunakan untuk membaca data dari suatu tabel atau *view*.

Sintaks:

Select namafield1, namafield2 ... from namatabel1 where kriteria

Atau

Select \* from namatabel1 where kriteria

#### 2. *Insert*

Digunakan untuk menambahkan data ke dalam tabel.

Sintaks:

Insert into namatabel (field1, ...) values ('nilai', ...)

#### 3. *Update*

Digunakan untuk memperbaiki data suatu field dalam suatu tabel.

Sintaks:

Update namatabel set field1=nilai, field2=nilai2... where kriteria

#### 4. *Delete*

Digunakan untuk menghapus data dari suatu tabel.

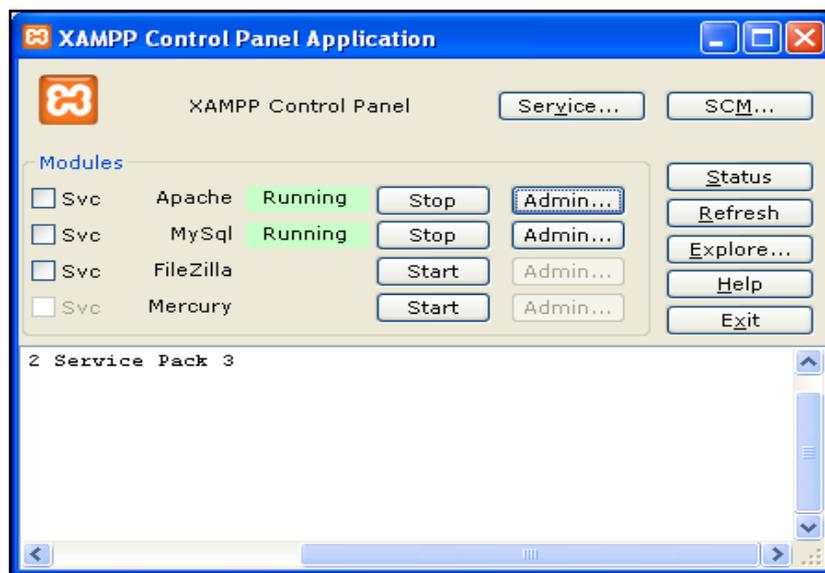
Sintaks:

Delete from namatabel where kriteria

### 2.4.5. Sekilas Tentang XAMPP

Sidik (2012:72) menyatakan, “XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) merupakan paket *serverweb* PHP dan databae MySQL yang paling populer di kalangan pengembang *web* dengan menggunakan PHP dan MySQL sebagai databasenya.

Sebelum menjalankan XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) yang harus dilakukan adalah memastikan bahwa Apache dan MySQL terkoneksi, dengan cara mengubah tombol *stop* pada action menjadi *start*. Berikut adalah tampilan XAMPP (X(Windows/Linux) Apache MySQL PHP dan Perl) control panel untuk menjalankan Apache dan MySQL.



**Gambar 2.2.** Tampilan XAMPP Control Panel Application



Berikut ini adalah tampilan awal dari XAMPP:



**Gambar 2.3.** Tampilan Awal XAMPP