



# **BAB II**

# **TINJAUAN PUSTAKA DAN**

# **LANDASAN TEORI**

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Pada tahun 2020 Rudi Kusuma dan Dedi Suhendro sudah melakukan penelitian dengan tujuan sistem ini dapat memberi kemudahan bagi pegawai untuk membuat perpanjangan dan pelunasan gadai emas untuk menghindari terjadinya penumpukan data yang sama. Perhitungan taksiran biaya pemeliharaan yang terbaru berlaku ketika perpanjangan. Setelah sudah mencapai tanggal jatuh tempo maka calon nasabah dapat memperpanjang masa gadai jika belum memiliki dana untuk melakukan pelunasan, atau bisa melunasi emas tersebut tanpa adanya perpanjangan. Sistem ini menggunakan metode perancangan sistem di antaranya membangun rancangan proses, rancangan input, rancangan output, rancangan database, rancangan sistem dan rancangan interface atau bisa disebut dengan metode Waterfall. Dalam merancang sistem ini digunakan Microsoft Visual Basic Net dan database MySQL. Hasil dari penelitian yang dilakukan adalah sebuah sistem yang digunakan untuk memperpanjang dan melunasi barang yang telah digadai<sup>[4]</sup>.

Penelitian sebelumnya tentang sistem informasi pegadaian yang dilakukan oleh Ferdy Ferdiansyah dan M. Sigit Safarudin pada tahun 2018. Dimana sistem ini dibuat untuk menganalisa sistem informasi pembayaran pegadaian secara online pada PT.Pegadaian Kota Batam.

Sehingga membuat akses informasi lebih mudah didapat dan pelayanan yang lebih modern tanpa batas waktu dan jarak tempuh. Penelitian ini menggunakan metode pengembangan perangkat lunak UML. Penelitian ini telah menghasilkan rancangan tentang Sistem Informasi transaksi Pegadaian pada PT Pegadaian Kota Batam kantor cabang Tiban. Hasil rancangan ini bisa digunakan sebagai acuan kepada programmer dalam menghasilkan sebuah sistem informasi transaksi. Dengan diterapkannya Sistem Informasi Pembayaran Pegadaian pada PT Pegadaian Kota Batam kantor cabang Tiban, diharapkan mampu mengatasi permasalahan pada sistem informasi yang sebelumnya<sup>[5]</sup>.

Fajrur Rahman Suprpto dan kawan-kawan melakukan penelitian pada tahun 2020, mereka meneliti tentang sistem informasi penjualan dan pelelangan. Hal ini dikarenakan belum terdapat media untuk mempermudah konsumen melihat produk dan stok barang yang tersedia secara *real-time* dan *up-to-date*. Maka perlu dibuat sistem informasi berbasis web yang bisa memproses penjualan, pelelangan, dan pengelolaan data secara efektif dan efisien. Hal ini juga dimaksudkan untuk memperluas jangkauan pemasaran, dan juga meningkatkan kenyamanan pelanggan untuk melakukan pembelian secara online. Penelitian ini menggunakan metode *Personal eXtreme Programming* (PXP) dipilih karena dapat digunakan oleh pengembang tunggal dan proses pengembangannya dinilai lebih efisien. Hasil dari penelitian ini menghasilkan sistem yang berjalan dengan baik dan lolos tahap pengujian sehingga sistem ini dapat diterapkan di Ricardo Corner MLG<sup>[6]</sup>.

Mariam Novianti sudah melakukan penelitian sebuah sistem informasi untuk PT. Pegadaian Sei Panas pada tahun 2017, pembuatan aplikasi ini memudahkan dalam pembuatan data laporan karena bisa langsung dapat diproses dan digunakan melalui aplikasi yang dibangun ini. Dengan pembuatan aplikasi Pegadaian berbasis *Java Netbeans* dan database MySQL ini maka proses penginputan data Pegadaian dari penginputan data anggota, data simpanan, data pinjaman, serta data angsuran tersimpan dengan aman. Sistem dibuat dengan metode UML (*Unified Modelling Language*). Hasil dari penelitian ini adalah sistem informasi pegadaian yang pada akhirnya benar-benar diterapkan pada PT. Pegadaian Sei Panas<sup>[7]</sup>.

Dari penelitian diatas, penulis bertujuan membuat sebuah sistem informasi. Sistem ini dibuat untuk menerapkan sistem yang mencakup pengajuan gadai dan lelang, hingga pembayaran angsuran. Aplikasi ini diharapkan dapat digunakan oleh PT. Pegadaian Cilacap yang berlokasi di Jalan Swadaya tersebut dalam melakukan gadai dan lelang barang. Sistem yang akan dibuat memiliki fitur pengajuan gadai dan lelang barang, jika barang gadai tidak ditebus dalam waktu yang sudah ditentukan maka secara otomatis barang akan masuk ke tahap pelelangan barang. Sistem ini akan dibuat dengan menggunakan metode Incremental Model, bahasa pemrograman PHP, dan database yang digunakan yaitu Database MySQL.

## **2.2 Landasan Teori**

### **2.2.1 Lelang**

Lelang dikenal sebagai suatu perjanjian yang termasuk jual beli baik dalam *Civil Law* maupun dalam *Common Law*. Di Indonesia lelang mulai dikenal sejak 1908 dengan diatur dalam Peraturan Perundang-Undangan yaitu dalam *Vendu Reglement* (Stbl. Tahun 1908 No. 189 diubah dengan Stbl. 1940 No. 56). Pengertian lelang menurut Pasal 1 angka 1 Peraturan Menteri Keuangan Nomor 106/PMK.06/2013 tentang Perubahan Atas Peraturan Menteri Keuangan Nomor 93/PMK/06/2010 tentang Petunjuk Pelaksanaan Lelang (selanjutnya disebut *Permenkeu Pelaksanaan Lelang*), adalah penjualan barang yang terbuka untuk umum dengan penawaran harga secara tertulis dan/atau lisan yang semakin meningkat atau menurun untuk mencapai harga tertinggi, yang didahului dengan Pengumuman Lelang. Dalam pelaksanaan lelang yang memimpin adalah seorang Pejabat Lelang. Lelang harus dilaksanakan dihadapan Pejabat Lelang. Kesepakatan harga dalam lelang baru terbentuk pada saat pejabat lelang untuk kepentingan penjual menunjuk penawar yang tertinggi dan mencapai harga limit sebagai pembeli lelang<sup>[1]</sup>.

Menurut Tim Penyusun Rancangan Undang-Undang Lelang Direktorat Jenderal Piutang dan Lelang Negara Biro Hukum Sekretariat Jenderal Departemen Keuangan, pengertian lelang adalah cara penjualan barang yang terbuka untuk umum dengan penawaran secara kompetisi yang didahului dengan pengumuman lelang dan atau upaya mengumpulkan peminat<sup>[1]</sup>. Adapun unsur-unsur yang terkandung dalam pengertian lelang adalah :

- a. Cara penjualan barang
- b. Terbuka untuk umum
- c. Penawaran dilakukan secara kompetisi
- d. Pengumuman lelang dan adanya upaya mengumpulkan peminat
- e. Cara penjualan barang yang memenuhi unsur-unsur.

Pada saat Proses lelang telah dilakukan sesuai dengan prosedur dan syarat yang benar, maka kepada pemenang lelang yaitu penawar tertinggi, akan diberikan risalah lelang, yang disebut juga Berita Acara Lelang. Berita Acara adalah risalah mengenai suatu peristiwa resmi dan kedinasan, disusun secara teratur, dimaksudkan untuk mempunyai kekuatan bukti tertulis bilamana diperlukan sewaktu-waktu. Risalah adalah laporan mengenai jalannya sesuatu pertemuan yang disusun secara teratur dan dipertanggungjawabkan oleh Pejabat Lelang, sehingga mengikat sebagai dokumen resmi dari kejadian / peristiwa yang disebut didalamnya<sup>[1]</sup>.

### **2.2.2 Rekayasa Web**

Rekayasa Web adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis Web (*Web-based applications*). Sebagai catatan, kebanyakan literatur tentang rekayasa Web mengacu kepada situs Web, aplikasi berbasis Web, sistem berbasis Web, dan aplikasi Web. Rekayasa Web adalah subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan

melibatkan aplikasi Web.

Aplikasi Web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis Web. Fitur-fitur aplikasi Web biasanya berupa data persistence, mendukung transaksi dan komposisi halaman Web dinamis yang dapat dipertimbangkan sebagai hibridisasi, antara hipermedia dan sistem informasi. Aplikasi Web adalah bagian dari *client-side* yang dapat dijalankan oleh browser Web. *Client-side* mempunyai tanggung jawab untuk pengekseskusion proses bisnis<sup>[8]</sup>. Interaksi Web dibagi ke dalam tiga langkah, yaitu :

1. Permintaan

Pengguna mengirimkan permintaan ke server Web, biasanya via halaman Web yang ditampilkan pada browser Web.

2. Pemrosesan

Server Web menerima permintaan yang dikirimkan oleh pengguna, kemudian memproses permintaan tersebut.

3. Jawaban

Browser menampilkan hasil dari permintaan pada jendela browser.

Pendekatan aplikasi Web mengambil keuntungan dari dasar Web server yang menyediakan komponen-komponen fungsional yang dapat digunakan kembali (*reusable*) antaraplikasi Web. Tidak semua pendekatan pada pengembangan aplikasi Web merupakan contoh dari framework aplikasi Web. Framework aplikasi Web yang benar akan menyediakan suatu pendekatan yang digabungkan untuk membangun

aplikasi Web yang dinamis. Pendekatan harus memberikan pengembang dan perancang halaman suatu arsitektur yang konsisten untuk membangun dan mengakses permintaan elemen-elemen konteks yang dapat ditempelkan pada halaman yang akan disajikan. Hal ini mencakup dukungan *untuk state, session management, dan authentication*, seperti halnya transformasi dan akses data<sup>[8]</sup>.


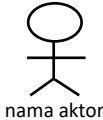

Perancangan sistem adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai perancangan sistem dapat dirancang dalam bentuk bagan alir sistem, yang merupakan alat bantu grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem<sup>[9]</sup>. Secara umum, pemodelan sistem yang ada di dalam perancangan sistem informasi ada 2 yaitu pemodelan terstruktur dan pemodelan berorientasi objek. Pemodelan terstruktur sebagai alur yang mengalir atau aliran proses (*Flowchart*), diagram konteks dan diagram alir data. Sedangkan pemodelan berorientasi objek bisa dilihat menggunakan UML (*Unified Modeling Language*). UML Digunakan untuk menspesifikasikan, menggambarkan, membangun dan mendokumentasikan dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung<sup>[9]</sup>. Pada pemodelan sistem informasi pengajuan gadai dan lelang, penulis menggunakan beberapa diagram UML yaitu :


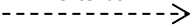
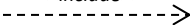


### 1. Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun<sup>[9]</sup>. *Use case* dideskripsikan sebagai interaksi antara aktor satu dengan yang lainnya menggunakan sistem informasi yang akan dibangun. *Use case* berguna untuk mengetahui fungsi apa saja yang ada dalam sistem informasi dan aktor siapa saja yang berhak menggunakannya. Simbol-simbol *Use Case Diagram* sebagai berikut :

**Tabel 2. 1** Simbol *Use Case Diagram*

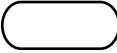

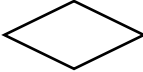

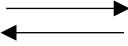
<b>Simbol</b>	<b>Keterangan</b>
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor</p>
<p>Aktor / <i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</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>

Generalisasi/ <i>generalization</i>  	Hubungan generalisasi dan spesialisasi aktor untuk dapat berantisipasi dengan <i>use case</i> .
Ekstensi / <i>extend</i>  <<extends>> 	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
<i>Include</i>  <<include>> 	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.

## 2. *Flowchart*

*Flowchart* adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *Flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *Flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek[9]. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan programmer) akan lebih mudah[9]. Berikut adalah simbol-simbol dari *Flowchart* :

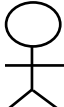
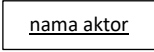



**Tabel 2. 2** Simbol *Flowchart*

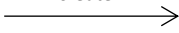

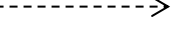
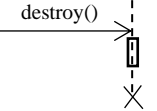
<b>Simbol</b>	<b>Keterangan</b>
<i>Terminator</i> 	Menyatakan permulaan atau akhir suatu program.
<i>Process</i> 	Proses penghitung/ proses pengolahan data
<i>Decision</i> 	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban.
<i>Document</i> 	Mencetak keluaran dalam bentuk dokumen.
<i>Arus atau flow</i> 	Menjelaskan jalannya arus suatu proses.

### 3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek<sup>[9]</sup>. Berikut adalah simbol-simbol dari *Sequence Diagram* :

**Tabel 2. 3** Simbol *Activity Diagram*

Simbol	Keterangan
<p>Aktor</p>  <p>nama aktor</p> <p>atau</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.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi.</p>
<p>Pesan tipe <i>create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain,</p>

<p style="text-align: center;"><code>&lt;&lt;create&gt;&gt;</code></p> 	<p>arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>send</i></p> <p style="text-align: center;">1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data /informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> <p style="text-align: center;">1 : keluaran</p> 	<p>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</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

### 2.2.3 Basis Data

Basis data merupakan mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi, dan dihapus<sup>[10]</sup>.

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang disediakan untuk membantu memelihara dan memanfaatkan kumpulan data yang besar. Kebutuhan akan sistem termasuk penggunaannya yang berkembang pesat. Alternatif penggunaan DBMS adalah menyimpan data dalam file dan menulis kode aplikasi tertentu untuk mengaturnya<sup>[10]</sup>.

Model Basisdata Relasional adalah tipe basisdata yang paling populer digunakan saat ini. Banyak perkembangan yang telah dibuat dari model basisdata sebelumnya menyederhanakan manajemen data dan pengambilan data. Data menjadi lebih mudah diatur, terutama melalui penggunaan batasan integritas. Pengambilan data merupakan proses penyaringan yang memungkinkan pengguna memvisualisasikan basisdata melalui struktur tabel relasional dan meminta data tertentu tanpa perlu memahami struktur basisdata. Unit penyimpanan utama dalam basisdata adalah tabel atau kelompok data

yang saling berhubungan. Sebuah tabel terdiri atas baris dan kolom. Baris berhubungan dengan record dalam tabel dan kolom mengandung nilai semua baris yang berhubungan dengan *field* tertentu. Tabel dapat dihubungkan satu sama lain melalui nilai kolom yang disebut kunci (*key*)<sup>[10]</sup>.

Ada tiga tipe hubungan antartabel, yaitu *one-to-one*, *one-to-many*, dan *many-to-many*. Suatu basisdata harus memiliki lebih dari satu hubungan berbeda. Hubungan antartabel ditentukan oleh integritas referensial (*referential integrity*) yang memerlukan penggunaan batasan kunci utama (*primary key*) dan kunci tamu (*foreign key*). Integritas referensial adalah penggunaan batasan di atas untuk membuat valid data yang dimasukkan ke dalam tabel dan mengatur hubungan antara tabel orangtua dan anak. Tipe batasan lain dapat pula dibuat untuk mengontrol data yang bisa dimasukkan dalam kolom tertentu dan membuat hubungan antartabel<sup>[10]</sup>.

*Entity Relationship Diagram* (ERD) adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antarentitas<sup>[10]</sup>.

- a. Entitas (*Entity*) Entitas adalah sesuatu yang nyata atau abstrak di mana kita akan menyimpan data.
- b. Relasi (*Relationship*) Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas
- c. Atribut (*Attribute*) Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data, dan *field*.

Metodologi ERD yaitu :


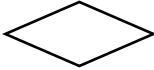


1. Menentukan Entitas  
Menentukan peran, kejadian, lokasi, hal nyata, dan konsep dimana pengguna akan menyimpan data.
2. Menentukan Relasi  
Menentukan hubungan antar pasangan entitas menggunakan matriks relasi.
3. Gambar ERD sementara  
Entitas digambarkan dengan kotak dan relasi dengan garis yang menghubungkan entitas.
4. Isi Kardinalitas  
Menentukan jumlah kejadian satu entitas untuk sebuah kejadian pada entitas yang berhubungan.
5. Tentukan Kunci Utama  
Menentukan atribut yang mengidentifikasi satu dan hanya satu kejadian masing-masing entitas.
6. Gambar ERD berdasar kunci  
Menghilangkan relasi *many-to-many* dan memasukkan primary dan kunci tamu pada masing-masing entitas.
7. Menentukan Atribut  
Menuliskan *field-field* yang diperlukan oleh sistem.
8. Pemetaan Atribut  
Untuk masing-masing atribut, memasangkan atribut dengan satu entitas yang sesuai.



9. Gambar ERD dengan atribut  
Mengatur ERD dari langkah 6 dengan menambah entitas atau relasi yang ditentukan pada langkah 8.
10. Periksa Hasil  
Periksa apakah ERD sudah menggambar sesuai dengan sistem yang akan dibangun.

Simbol-Simbol ERD dapat dilihat pada tabel 2.4 :

**Tabel 2. 4** Simbol *ERD*

Simbol	Keterangan
Entitas / <i>entity</i> 	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
Relasi 	Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda.
Atribut 	Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> diberi garis bawah).
Garis 	Sebagai penghubung antara relasi dan entitas atau entitas dengan atribut.

DDL (*Data Definition Language*) merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, dalam hal ini database dan tabel<sup>[11]</sup>. Perintah SQL yang termasuk dalam DDL antara lain :

1. **Create**

*Create* berfungsi untuk membuat sebuah database baru dengan query 'CREATE DATABASE nama\_database'.

2. **Alter**

*Alter* berfungsi untuk mengubah struktur tabel yang sebelumnya telah dibuat. Salah satunya untuk menghapus kolom, mengubah nama tabel dan sebagainya.

3. **Rename**

*Rename* berfungsi untuk mengubah atau mengganti nama tabel yang sebelumnya telah dibuat dengan query 'RENAME TABLE nama\_tabel\_lama TO nama\_tabel\_baru'.

4. **Drop**

*Drop* berfungsi untuk menghapus tabel, kolom, database, dan index.

DML (*Data Manipulation Language*) DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data atau *record* dalam table<sup>[11]</sup>. Perintah SQL yang termasuk dalam DML adalah :

1. **Select**

*Select* berfungsi untuk menampilkan data dari tabel.

2. **Insert**

*Insert* berfungsi untuk menginput atau memasukkan data yang baru ke sebuah tabel.

3. **Update**

*Update* berfungsi untuk menambah atau membaharui data

yang sudah ada.

#### 4. *Delete*

*Delete* berfungsi untuk menghapus data yang sebelumnya telah dibuat dalam sebuah tabel.

### 2.2.4 Pemrograman Berbasis Objek

Pemrograman berorientasi objek, atau sering juga disebut dengan OOP (*Object Oriented Programming*), merupakan paradigma baru dalam pengembangan sistem/perangkat lunak. Tidak seperti pendahulunya, yaitu pemrograman dengan teknik terstruktur (*structured programming*) yang sering mengalami kegagalan dalam hal sistem perangkat lunak yang tidak sesuai dengan kebutuhan dan harapan pengguna (*user's need and expectations*) serta kerap kali tidak dapat memenuhi jadwal yang telah ditentukan sebelumnya, pemrograman berorientasi objek dengan paradigma barunya, menjanjikan tingkat penggunaan-ulang (*reusability*), kinerja (*performance*) serta keandalan (*reliability*) yang lebih tinggi<sup>[12]</sup>.

Dalam sistem berorientasi objek kita menggabungkan potongan-potongan informasi dan perilaku-perilaku spesifik yang bekerja pada informasi tersebut, kemudian mengemasnya menjadi apa yang disebut sebagai objek. Ini dirujuk dengan kata pembungkusan (*encapsulation*). Cara untuk memahami *encapsulation* adalah membagi aplikasi-aplikasi menjadi bagian-bagian kecil yang secara fungsional berhubungan. Keuntungan dari *encapsulation* adalah membatasi efek-efek perubahan pada sistem<sup>[12]</sup>.

Pewarisan (*Inheritance*) Pewarisan adalah konsep dasar pemrograman berorientasi objek yang lainnya. Pada sistem berorientasi objek, pewarisan adalah mekanisme yang memungkinkan kita menciptakan objek-objek baru berdasarkan objek lain yang sudah ada: Objek Anak mewarisi segala sesuatunya dari objek Induk. Salah satu keuntungan dan pewarisan (*inheritance*) adalah kemudahannya untuk dipelihara<sup>[12]</sup>.

*Polimorfisme* adalah prinsip penting yang ketiga dari pemrograman berorientasi objek. *Polimorfisme* berarti suatu fungsionalitas yang diimplementasikan dengan berbagai cara yang berbeda. Pada terminologi berorientasi objek, ini berarti kita dapat memiliki berbagai implementasi untuk sebagian fungsionalitas tertentu. Saat pengguna mau menggambar sesuatu, entah itu garis, atau lingkaran, atau elips, sistem akan memunculkan perintah gambar. Sistem mengenali berbagai bentuk gambar masing-masing dengan perilakunya. Seandainya pengguna akan menggambar sebuah garis, maka perintah untuk menggambar garis akan dibangkitkan. Dengan *polimorfisme*, kode untuk menggambar dilakukan hanya dengan memanggil fungsi *DrawMe()* untuk setiap objek yang akan digambar. Dimana masing-masing bentuk memiliki fungsi *DrawMe()* untuk menggambar bentuknya masing-masing. Salah satu keuntungan dari *polimorfisme* adalah kemudahan pemeliharannya<sup>[12]</sup>.