

## **BAB II**

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

#### **2.1 Tinjauan Pustaka**

Penelitian yang dilakukan oleh Noor Hasan dkk [2], dengan judul “Aplikasi Sistem Informasi Koperasi Simpan Pinjam dengan Metode *Incremental*”. Metode ini memiliki risiko yang lebih kecil serta memiliki keunggulan pada proses yang parallel sehingga antar tahapan tidak harus selalu saling menunggu. Hasil dari perancangan ini adalah sistem informasi simpan pinjam dengan metode pengembangan sistem *incremental* serta pengujian *black box testing* yang diharapkan sistem akan berjalan baik serta memenuhi kebutuhan *user* sesuai dengan analisa kebutuhan perangkat lunak.

Penelitian yang dilakukan oleh Siti Yuliansyah dkk [1], dengan judul “Sistem Informasi Pencatatan Simpan Pinjam Pada Koperasi PrimKop Kartika Salak” dalam pencatatanya masih menggunakan metode semi-manual. Penumpukan dokumen menimbulkan banyak permasalahan. Demi meningkatkan pelayanan kepada anggota dan mengatasi permasalahan yang ada, peneliti membangun sebuah sistem informasi simpan pinjam menggunakan metode *waterfall* untuk alur pembuatan sistem, menggunakan UML sebagai analisa dan design serta menggunakan *software netbeans* guna penerapan sistem berbasis objek. Pembaruan sistem semi-manual kepada sistem terkomputerisasi diharapkan dapat mengatasi permasalahan yang membantu dalam menghasilkan keputusan akurat sehingga pelayanan terhadap anggota dapat meningkat.

Penelitian yang dilakukan oleh Karya Suhada dkk [3], dengan judul “Perancangan Sistem Informasi Koperasi Simpan Pinjam Berbasis *Dekstop* Pada Perum CKM Karawang.” menggunakan metode *waterfall*. Sedangkan untuk membuat aplikasi, peneliti menggunakan Bahasa pemrograman *Java* dan menggunakan *database MySQL* sehingga data yang telah diproses disimpan ke dalam *database* tersebut. Hasil dari penelitian ini adalah perancangan sistem aplikasi koperasi simpan pinjam pada perum CKM yang diharapkan dapat membuat administrasi yang akurat dan handal. Dengan penelitian ini diharapkan membantu pendataan transaksi simpan pinjam yang akurat dan benar serta data terorganisasi dengan baik, tersimpan dengan aman dan mudah ditelusuri.

Penelitian lain dilakukan oleh Matheus dkk [4], dengan judul “Rancang Bangun Aplikasi Koperasi Simpan Pinjam menggunakan Metode *Research and Development*”. Metode ini untuk menghasilkan sebuah aplikasi dan menguji keefektifannya sehingga benar-benar menghasilkan aplikasi yang baik dan efektif. Hasil dari perancangan ini yaitu Aplikasi Koperasi Simpan Pinjam Menggunakan *Metode Research And Development* dengan *Black box Testing*. Aplikasi ini menggunakan *Visual Basic.Net* serta *MySql* untuk *database* yang dapat mempermudah koperasi dalam mengelola data anggota, pengambilan, peminjaman, serta dapat menampilkan rekapan data, sehingga menjadi informasi yang bermanfaat serta meminimalisir *human-error* karena pencatatan transaksi dilakukan melalui aplikasi.

Penelitian lain dilakukan oleh Widya dkk [5], dengan judul “Penerapan Model *Waterfall* dalam Perancangan Aplikasi Sistem Informasi Simpan Pinjam pada Koperasi PT. Chiyoda”. Penelitian ini menggunakan *software Visual Basic.Net* untuk merancang sistem. Sistem ini membantu mengelola data transaksi seperti mempercepat proses pencatatan dan pelaporan. Namun ada hal yang harus dikembangkan yaitu mengoptimalkan fungsi *backup* untuk keamanan pencatatan transaksi, mengadakan pelatihan kepada *user* agar menggunakan aplikasi dengan baik serta melakukan pemeriksaan program agar mengetahui masalah yang muncul ketika program digunakan.

Berdasarkan kajian penelitian yang telah dilakukan, maka akan dibuat Sistem Informasi Simpan Pinjam Berbasis *Website*. Meskipun sistem dengan judul sejenis ini telah ada sebelumnya, namun tetap memiliki perbedaan pada metode pengembangan, bahasa pemrograman, dan fitur yang ada di dalamnya. Metode pengembangan yang digunakan adalah metode *waterfall*, yang diharapkan dapat mempermudah proses transaksi simpan pinjam serta meningkatkan kinerja koperasi, sehingga pelayanan kepada anggota koperasi dapat lebih baik. Sistem ini berbasis *website*, yang memungkinkan akses dari mana saja oleh anggota koperasi. Sistem ini juga dilengkapi dengan fitur notifikasi *whatsapp* tentang status permohonan, baik ketika mengisi *form* permohonan, telah mengajukan permohonan, maupun saat menerima keputusan permohonan. Verifikasi akan dilakukan melalui sistem dengan tanda tangan *digital* pada *form* yang dicetak. Selain itu, sistem ini mencakup grafik anggota dan transaksi simpan pinjam, laporan keaktifan anggota, serta laporan koperasi yang dapat dipantau langsung oleh pengawas dan ketua koperasi.

## **2.2 Landasan Teori**

### **2.2.1 Sistem Informasi**

Sistem adalah sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan tugas secara bersama-sama[6]. Informasi adalah data yang diolah menjadi bentuk yang berguna dan berarti bagi penerimanya. Sumber informasi adalah sebuah data yang menggambarkan suatu kejadian-kejadian yang nyata dan terjadi pada saat tertentu[7].

Fungsi informasi sendiri yaitu untuk menambah pengetahuan atau mengurangi ketidak pastian pemakai informasi karena informasi berguna memberikan gambaran mengenai suatu permasalahan sehingga pengambil keputusan dapat menentukan keputusan lebih cepat. Sistem informasi adalah sistem yang berisi pengolahan data yang memproses pengumpulan data, mengelola data yang tersimpan dan menyebarkan informasi[8]. Kegiatan yang ada di dalam Sistem Informasi mencakup:

1. *Input*, menggambarkan suatu kegiatan untuk menghasilkan suatu informasi yang bernilai tambah.
2. Proses, menggambarkan bagaimana suatu data di proses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, kegiatan untuk menghasilkan laporan proses diatas.
4. Penyimpanan, suatu kegiatan untuk memelihara data.
5. Kontrol, suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan[9].

### **2.2.2 Rekayasa Perangkat Lunak**

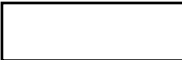
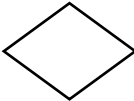

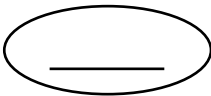


Perangkat lunak merupakan kumpulan perintah yang digunakan untuk memproses informasi dalam bentuk program atau prosedur. Program adalah serangkaian instruksi yang dipahami oleh komputer, sementara prosedur adalah perintah yang diperlukan oleh pengguna dalam mengolah informasi. Rekayasa Perangkat Lunak (RPL) adalah ilmu yang mencakup seluruh tahapan produksi perangkat lunak, mulai dari analisis kebutuhan pengguna, penentuan spesifikasi, desain, pengkodean, pengujian, hingga pemeliharaan sistem setelah digunakan[8].

#### **1. Entity Relationship Diagram (ERD)**

ERD (*Entity-Relationship Diagram*) termasuk ke dalam bidang rekayasa perangkat lunak dan desain basis data. ERD adalah alat visual yang digunakan dalam fase analisis dan perancangan perangkat lunak atau

sistem yang menggunakan basis data untuk menggambarkan hubungan antara entitas-entitas (objek-objek) dalam basis data tersebut. *Entity relationship diagram* menggambarkan data dan hubungan antar data secara global dengan menggunakan *entity relationship diagram*[7]. Berikut adalah simbol-simbol pada ERD yang dapat dilihat pada **Tabel 2.4**

**Tabel 2. 1** Simbol-Simbol ERD

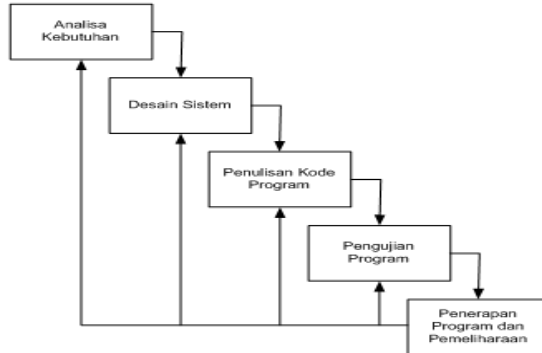
No	Simbol	Fungsi
1.	Entitas 	Entitas adalah data inti yang akan disimpan agar dapat diakses oleh <i>computer</i>
2.	Relasi 	Relasi adalah yang menghubungkan antar entitas, biasanya diawali dengan kata kerja
3.	Atribut 	Atribut adalah <i>field</i> atau kolom data yang butuh disimpan dalam suatu entitas
4.	Atribut Kunci Primer 	Atribut kunci primer adalah <i>Field</i> yang disimpan dalam suatu entitas kunci akses <i>record</i> yang diinginkan
5.	Atribut Multinilai 	Atribut multinilai adalah field yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
6.	Alur 	Asosiasi adalah penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

(Sumber: Rosa A. S. & Shalahuddin, 2015)

## 2. Metode Pengembangan Sistem

Metode penelitian yang digunakan adalah metode *waterfall*. *Waterfall* atau air terjun adalah model yang dikembangkan untuk pengembangan perangkat lunak. Metode ini berkembang secara

sistematis dari satu tahap ke tahap lain. Model *waterfall* terdiri dari 5 tahapan, yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, penerapan program dan pemeliharaan. Model *waterfall* dapat dilihat pada gambar berikut:



**Gambar 2. 1** Pemodelan *Waterfall* menurut Sommerville[10]

Tahapan-tahapan metode *waterfall* menurut Sommerville adalah[10] :

1. Analisa Kebutuhan (*Analysis*)

Pada tahap ini, data dikumpulkan melalui wawancara dengan pengguna. Seorang analis akan mencari informasi sebanyak mungkin dari pengguna untuk menciptakan sebuah sistem komputer yang dapat menjalankan tugas-tugas sesuai keinginan pengguna. Hasil dari tahap ini adalah dokumen yang berisi keinginan pengguna dalam pembuatan sistem. Dokumen tersebut akan menjadi acuan untuk menerjemahkan ke dalam bahasa pemrograman.

2. Desain Sistem (*Design*):

Proses desain sistem akan menerjemahkan kebutuhan sistem ke dalam perancangan perangkat lunak yang dapat diprediksi sebelum *coding* dilakukan. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, tampilan antarmuka, dan detail prosedur. Tahap ini menghasilkan dokumen yang disebut *software requirement*. Dokumen ini akan digunakan oleh programmer untuk membuat sistem.

3. Penulisan Kode Program (*Coding*):

Tahap ini adalah penerjemahan desain ke dalam bahasa pemrograman yang dimengerti komputer. Setelah pengkodean, dilakukan pengujian terhadap sistem untuk menemukan dan memperbaiki kesalahan.

4. Pengujian Program (*Testing*):

*Testing* merupakan tahap akhir pembuatan sistem. Setelah analisis, desain, dan pengkodean, sistem akan digunakan oleh pengguna.

5. Penerapan Program dan Pemeliharaan (*Maintenance*):

*Software* yang telah diserahkan kepada pengguna kemungkinan akan mengalami perubahan karena harus disesuaikan dengan lingkungan atau karena pengguna memerlukan peningkatan fungsionalitas.

**3. Metode Pengujian Sistem**

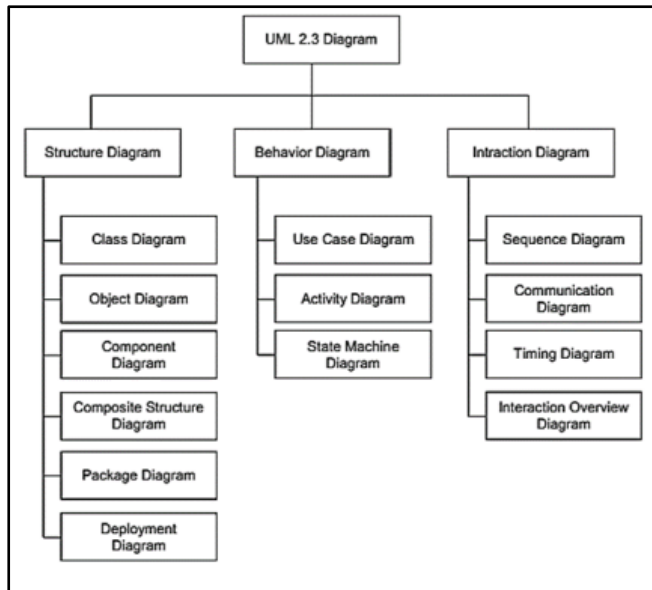
*Black box testing* yaitu menguji *software* dari segi spesifikasi fungsional tanpa menguji desain dan kode program[11]. Pengujian ini untuk mengetahui fungsi, *input*, dan *output* dari *software* sesuai dengan spesifikasi yang dibutuhkan. Pengujian ini dilakukan dengan mencoba semua fungsi menggunakan *software*. Kasus uji harus dibuat dengan kasus benar dan salah, misalkan untuk kasus proses *login* maka yang dilakukan adalah:

- a) Jika *user* memasukan *username* dan *password* yang benar
- b) Jika *user* memasukan *username* dan *password* yang salah, contohnya *username* benar *password* salah, atau sebaliknya, atau keduanya salah.

**4. Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sistem dengan menggunakan diagram teks pendukung. UML terbaru adalah UML 2.3 yang terdiri dari 13 macam diagram yang dikelompokkan menjadi 3 kategori. Tujuan dari Unified Modeling Language (UML) sebagai berikut:

- a) Memodelkan sistem yang menggunakan konsep berorientasi objek.
- b) Menciptakan bahasa pemodelan yang dapat digunakan oleh manusia maupun mesin.
- c) Memberikan bahasa yang bebas dari berbagai bahasa pemrograman.



**Gambar 2. 2** Pembagian kategori dan macam-macam diagram


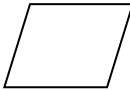
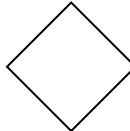
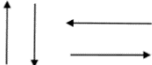




(Sumber: Rosa A. S. & Shalahuddin, 2015)

- a) *Structure diagrams* untuk menggambarkan struktur statis dari sistem yang dimodelkan.
- b) *Behavior diagrams* untuk menggambarkan rangkaian perubahan yang terjadi pada sebuah sistem.
- c) *Interaction diagrams* untuk menggambarkan interaksi sistem dengan sistem lain atau interaksi antar subsistem pada suatu sistem.

**a) Flowchart Sistem**

Flowchart merupakan kumpulan simbol yang menggambarkan urutan proses dalam menyelesaikan suatu permasalahan[11]. Flowchart menunjukkan urutan proses pemecahan masalah. Setiap simbol dihubungkan dengan garis dari awal sampai akhir. Flowchart dimulai dengan simbol terminator mulai yang menyatakan awal dan diakhiri dengan simbol terminator selesai untuk menyatakan akhir. Simbol-simbol flowchart beserta keterangannya dapat dilihat pada **Tabel 2.2**

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

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program.
2.		<i>Input – Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
4.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain. simbol arus ini sering disebut dengan <i>connecting line</i> .
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Manual Input</i>	Memasukkan data secara manual <i>on-line keyboard</i> .
8.		<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer.

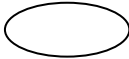
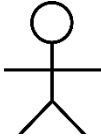

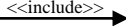
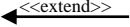

(Sumber: Rosa A. S. &amp; Shalahuddin, 2015)



**b) Usecase Diagram**

*Use case diagram* merupakan sebuah diagram yang menggambarkan interaksi antara pengguna dengan sebuah sistem yang dapat menjadi gambaran untuk menjelaskan batasan-batasan dari sistem[12]. Terdapat dua elemen penting dalam diagram use case yaitu actor dan use case. Actor adalah pengguna yang berinteraksi langsung dengan sistem, sedangkan use case adalah gambaran dari interaksi yang terjadi antara aktor dengan sistem. Simbol-simbol *Use Case Diagram* bisa dilihat pada **Tabel 2.2**.

**Tabel 2. 3** Simbol-Simbol *Use Case Diagram*

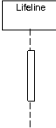
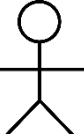
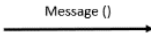
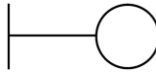

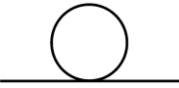
No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor
2.		<i>Actor</i>	Proses yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri
3.		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang saling berpartisipasi pada <i>use case</i>
4.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> ini untuk menjalankan fungsi
5.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana use case yang ditambahkan bisa berdiri sendiri
7.		<i>Generalisasi</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

(Sumber: Rosa A. S. & Shalahuddin, 2015)

c) **Secuence Diagram**

*Sequence diagram* adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek[11]. Banyaknya *sequence diagram* yang harus digambar mementingkan semua *use case* yang sudah didefinisikan interaksi jalanya pesan dicakup dalam *sequence diagram* agar semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada *sequence diagram*, sesuai pada **Tabel 2.4**.

**Tabel 2. 4** Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Orang atau proses yang berinteraksi dengan sistem informasi yang akan dibuat
3.		<i>Message</i>	Menyatakan sebuah objek membuat objek yang lain, arah panah mengarah ke objek yang akan dibuat
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan

(Sumber: Rosa A. S. & Shalahuddin, 2015)

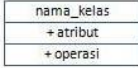
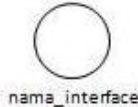


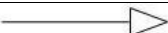

### a) *Class Diagram*

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Suatu kelas mempunyai atribut dan metode.

1. Atribut adalah *variable* yang dipunyai oleh suatu kelas.
2. Metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas pada struktur sistem harus melakukan fungsi yang sesuai berdasarkan kebutuhan sehingga pembuat *software* dapat membuat kelas di dalam program *software* sesuai dengan perancangan diagram. Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2. 5** Simbol-Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem
2.		<i>Interface</i> sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Asosiasi adalah relasi antarkelas dengan makna umum
4.		Asosiasi berarah adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
5.		Generalisasi adalah relasi antarkelas dengan makna (umum khusus)
6.		Kebergantungan adalah relasi antarkelas dengan makna kebergantungan antarkelas

(Sumber: Rosa A. S. & Shalahuddin, 2015)

### 2.2.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah strategi pembangunan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi

yang diberlakukan terhadapnya[13]. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktifitas  
 Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai diulang kembali untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
2. Kecepatan pengembangan  
 Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan  
 Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan.
4. Adanya konsistensi  
 Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak  
 Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Beberapa konsep dasar metodologi berorientasi objek adalah:

- 1) Kelas (*class*)  
 Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.
- 2) Objek (*object*)  
 Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, suatu organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status

objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3) Metode (*method*)

Operasi atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

4) Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama dan sebagainya.

5) Abstraksi (*abstraktion*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6) Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7) Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8) Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain.

9) *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10) Generalisasi dan Spesifikasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

11) Komunikasi Antar objek

Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirimkan dan satu objek ke objek lainnya.

12) Polimorfisme (*polymorphism*)

Kemampuan suatu objek yang digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13) *Package*

*Package* adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga bernama sama disimpan dalam *package* yang berbeda

### 2.2.4 Basis Data

*Database* adalah sistem terkomputerisasi yang bertujuan untuk memelihara data yang sudah diolah serta untuk menyimpan data agar dapat diakses dengan mudah[11]. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database*, yaitu (*Structured Query Language*). *SQL* adalah bahasa untuk mengakses data di dalam sebuah *database*. *SQL* sering juga disebut dengan istilah *query*. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan manajemen data yang terdapat dalam *database*. Secara garis besar, *SQL Server* mempunyai 2 (Dua) jenis perintah *SQL* yaitu:

1. ***Data Definition Language (DDL)*** dapat digunakan oleh pengguna untuk menguraikan *database*, contohnya seperti menentukan detail tipe dan batasan data yang akan disimpan dalam *database*. Terdapat tiga perintah penting dalam DDL, yaitu:
  - a) *CREATE*, untuk membuat, termasuk di dalamnya membuat *database* baru, tabel baru view baru, dan kolom baru. Contoh: *CREATE DATABASE nama\_database*
  - b) *ALTER*, untuk mengubah struktur tabel yang telah dibuat termasuk mengubah nama tabel, menambah, mengubah, menghapus, dan memberikan atribut pada kolom. Contoh: *ALTER TABLE nama\_tabel ADD nama\_kolom datatype.*
  - c) *DROP*, untuk menghapus objek dari basis data, seperti tabel, indeks, atau *view* yang memiliki dampak permanen, sehingga objek yang dihapus tidak dapat dipulihkan tanpa pemulihan dari (*backup*) basis data sebelumnya. Contoh: *DROP TABLE nama\_tabel; DROP INDEX nama\_indeks ON nama\_tabel; DROP VIEW nama\_view;*
2. ***Data Manipulation Language (DML)*** dapat digunakan pengguna untuk memanipulasi data, seperti menambahkan, menghapus, dan mengambil data dari *database*. Terdapat 4 perintah penting dalam DML, yaitu:
  - a) *INSERT*, untuk memasukkan data baru ke dalam tabel yang dijalankan ketika *database* dan tabel sudah dibuat. Contoh: *insert into nama\_tabel values (data1, data2, dst...);*

- b) *UPDATE*, untuk memperbarui data pada tabel. Contoh: *update nama\_tabel set kolom1=data1, kolom2=data2,... where kolom=data;*
- c) *DELETE*, untuk menghapus data dari tabel. Contoh: *delete from nama\_tabel where kolom=data;*
- d) *SELECT*, untuk mengambil dan menampilkan data dari tabel atau beberapa tabel dengan penggunaan relasi. Contoh: *select nama\_kolom1, nama\_kolom2 from nama\_tabel;*

### 2.2.5 MySQL


MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau *Database Management System* (DBMS), basis data ini *multithread* atau *multi-user*. SQL pertama kali didefinisikan oleh *American National Standard Institute* (ANSI) pada tahun 1986. Kepopuleran MySQL karena menggunakan SQL sebagai Bahasa dasar untuk mengakses *database*-nya, sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi kebutuhan *database* perusahaan skala menengah-kecil. MySQL juga bersifat *open source General Public License* (GPL) mulai versi 3.23, pada bulan Juni 2000[14].

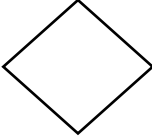
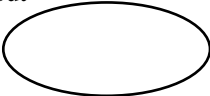
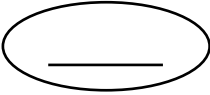


MySQL bersifat *relational*, artinya data yang dikelola dalam basis data akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. MySQL dapat digunakan untuk mengelola basis data mulai dari yang kecil sampai yang sangat besar. MySQL juga dapat menjalankan perintah *Structured Query Language* (SQL) untuk mengelola basis data-basis data yang ada di dalamnya.

### 2.2.6 Entity Relationship Diagram (ERD)

Dalam rekayasa perangkat lunak diperlukan adanya *Entity Relationship Diagram* (ERD). ERD merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Di dalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjuk objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya. Simbol-simbol ERD dapat di lihat pada **Tabel 2.6**.

**Tabel 2. 6** Simbol – simbol ERD

No	Simbol	Fungsi
1.		Entitas adalah data yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh komputer.

2.	Relasi 	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
3.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
4.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan.
5.	Atribut Multinilai 	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
6.	Alur 	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

### 2.2.7 Koperasi

Koperasi berasal dari bahasa Inggris yaitu *cooperation* yang artinya bekerja sama. Secara umum koperasi diartikan sebagai sekumpulan orang yang secara sukarela mempersatukan diri untuk peningkatan kesejahteraan ekonomi bersama melalui sebuah badan usaha yang dikelola secara demokratis[15]. Ada beberapa jenis koperasi salah satunya adalah koperasi simpan pinjam. Koperasi simpan pinjam merupakan badan usaha yang memberikan bantuan peminjaman dan penyimpanan dana. Dana tersebut berasal dari uang administrasi, simpanan pokok, simpanan wajib, dan simpanan sukarela dari anggota koperasi. Koperasi ini bertujuan untuk membantu anggota dalam memberi pinjaman dengan bunga yang tidak terlalu tinggi serta



membantu untuk menabung sehingga dana yang terkumpul nanti dapat digunakan oleh anggota koperasi. Ada tiga macam simpanan dalam koperasi yaitu:

1. Simpanan Pokok  
Simpanan pokok merupakan sejumlah uang yang wajib dibayarkan kepada koperasi saat masuk menjadi anggota, yang tidak dapat diambil kembali selama yang bersangkutan masih menjadi anggota koperasi.
2. Simpanan Wajib  
Simpanan wajib adalah jumlah uang yang harus dibayarkan oleh anggota secara teratur dengan ketentuan yang ditetapkan oleh koperasi[16].
3. Simpanan Sukarela  
Simpanan sukarela adalah sejumlah uang yang tidak wajib dibayarkan oleh anggota, namun jika anggota tersebut ingin membayar simpanan ini maka boleh membayarkan nominalnya sesuai dengan keinginan anggota.