BAB II
DASAR TEORI

BAB II
DASAR TEORI

2.1. Sistem

Sistem adalah kumpulan dari unsur, komponen, maupun variabel yang terorganisir,dan saling berinteraksi, serta bergantung satu sama lain. Dari penjelasan tersebut, maka dapat disimpulkan bahwa sistem merupakan kesatuan yang saling berhubungan untuk melaksanakan kegiatan tertentu bersama-sama dalam rangka mencapai suatu tujuan[3].

2.2. Sistem Informasi

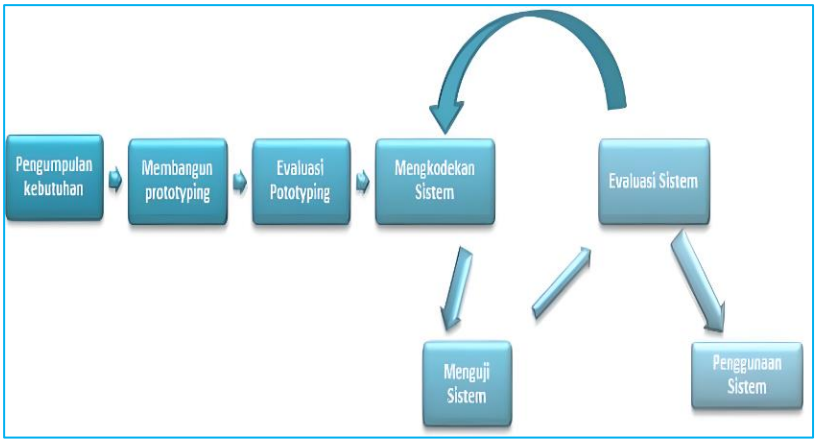
Sistem informasi adalah sebuah kumpulan komponen yang saling berhubungan satu dengan yang lain untuk mencapai suatu tujuan. Dengan adanya sistem informasi maka pekerjaan yang dilakukan melalui *hardcopy* sekarang dapat dilakukan menggunakan komputer. Sistem informasi dinilai sangat dibutuhkan oleh suatu instansi ataupun perusahaan. Hal ini karena sistem informasi yang terintegrasi akan membuat kerja dari perusahaan atau instansi menjadi lebih sistematis serta terarah.[4]

2.3. Inventaris

Inventaris merupakan simpanan barang-barang mentah, material atau barang jadi yang disimpan untuk digunakan dalam masa mendatang atau dalam kurun waktu tertentu. Hal ini menjadi penting karena nantinya akan banyak sekali hal yang bergantung pada catatan atau informasi yang ada di dalamnya, sehingga wajib dipastikan semua data yang ada di dalam daftar ini adalah valid dan benar dari sisi jumlah serta kualitasnya[5].

A. Metode Pengembangan Sistem

Metode Pengembangan Sistem yang akan diterapkan pada sistem ini adalah model prototipe dari siklus hidup pengembangan perangkat lunak (*Software Development Life Cycle/SDLC*). Metode ini memfasilitasi komunikasi antara pengembang dan pengguna untuk memastikan hasil yang sesuai dengan kebutuhan pengguna melalui desain fungsionalitas aplikasi. Berikut adalah tahapan metode prototipe menurut Pressman (2012) [3]:. Metode *prototype* dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Metode *Prototype*

Berdasarkan model *prototype* yang telah digambarkan di atas, dapat diuraikan pembahasan disetiap tahapannya sebagai berikut [6]:

1. Pengumpulan Kebutuhan

   Dalam tahapan ini, dilakukan observasi dan wawancara secara langsung kepada guru dan wakasek yang akan menggunakan sistem peminjaman barang ini. Wawancara dilakukan bersama guru, petugas tata usaha dan wakasek, dengan mengidentifikasi kebutuhan dan garis besar sistem yang akan dibuat.

2. Membuat *Prototipe*

   Kemudian berdsarkan hasil pada tahapan sebelumnya, maka dilanjutkan dengan membuat desain *mockup* sementara dan *prototyping* yang berfokus pada garis besar sistem dan dipresentasikan kepada pengguna.

3. Evaluasi Prototipe

   Setelah pengembang melakukan presentasi, kemudian pengguna melakukan evaluasi apakah sesuai dengan kebutuhan mereka, Jika prototipe memenuhi kriteria, maka proses berlanjut pada tahap berikutnya. Namun, jika tidak sesuai, maka langkah harus diulangi dari awal.

4. Pengkodean Sistem

   Setelah tahap prototyping telah disepakati, kemudian diimplementasikan kedalam bahasa pemrogramman yang tepat. Disini pengembang menggunakan bahasa pemrograman PHP dan *framework Codeigniter* serta *Bootstrap5*.

5. Pengujian Sistem

   Setelah sistem telah dikembangkan menjadi perangkat lunak yang siap digunakan,

tahap pengujian dilakukan dengan menggunakan metode *black-box testing* dengan melibatkan pengguna untuk mencoba sistem secara langsung.

6. Evaluasi Sistem

   Kemudian pengguna melakukan evaluasi terhadap sistem yang telah selesai tahap *testing* untuk memastikan kesesuaian dengan harapan pengguna. Jika sistem sesuai dengan kriteria yang diharapkan, maka akan berlanjut ke tahap berikutnya. Namun, jika belum atau tidak sesuai maka tahapan sebelumnya akan diulangi.

7. Penggunaan Sistem

   Pada tahapan terakhir, jika sistem yang telah diuji dan disetujui oleh pengguna maka sistem siap digunakan. Dengan kebutuhan yang ada apakah sistem akan dijalankan secara lokal atau hosting ke *internet*..

B. Metode Pengujian Sistem

Pengujian yang akan digunakan sistem ini yaitu pengujian *black box.* Pengujian black box merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak, dan kemudian keluaran dari perangkat lunak diuji kelayakannya. Keuntungan *dari black box testing* yaitu pengujian ini berasal dari sudut pandang pengguna dan tidak memerlukan pengetahuan yang tinggi dalam pemrogaman sehingga lebih efisien dalam menemukan ketidaksesuaian spesifikasi yang diinginkan pengguna. Sedangkan kekurangan dari pengujian *black box* ini ialah cakupan pengujian dinilai terbatas karena hanya fungsional perangkat lunak yang diujikan [12].

Adapun struktur data merupakan cara menyimpan atau mempresentasikan data didalam komputer agar bisa dipakai secara efisien. Berikut bagian-bagian yang ada pada struktur data:

A. *Flowchart*

Flowchart atau diagram alir adalah jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi secara berurutan dalam suatu sistem. Analis sistem menggunakan *flowchart* sebagai bentuk dokumentasi untuk menjelaskan gambaran logis suatu sistem kepada para *programmer*. *Flowchart* memiliki peran penting dalam memberikan solusi terhadap potensi masalah yang mungkin timbul selama pengembangan sistem. Proses-proses dalam *flowchart*

digambarkan menggunakan simbol-simbol khusus, di mana setiap simbol mencerminkan suatu tindakan atau proses tertentu. Garis penghubung digunakan untuk menghubungkan satu proses dengan proses berikutnya. Berikut adalah contoh simbol-simbol yang umum digunakan dalam flowchart diagram[6]. Simbol *Flowchart* dapat dilihat pada Tabel 2.1:

Tabel 2. 1 Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		*Terminal*	Awal atau akhir dari suatu prosedur.
2		Proses	Proses operasional yang terjadi dalamkomputer.
3		*Output/Input*	Memasukkan atau mengeluarkan data tanpa melihat jenis perangkat yang digunakan.
4		Garis Alur	Menggambarkan alur atau aliran program.
5		*Decision*	Menandakan suatu kondisi yang mengarahkan pada dua kondisiYA/TIDAK.
6		Simbol Dokumen	Untuk mencetak dokumen berupakertas.

7		*Manual Operation*	Menunjukan pengolahn yang tidak dilakukan oleh komputer

B. *Unified Modelling Language (UML)*

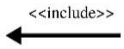
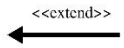
UML (Unified Modeling Language) merupakan metode pemodelan secara visual yang digunakan untuk merancang sistem berorientasi objek. *UML* juga dapat diinterpretasikan sebagai bahasa standar untuk visualisasi, perancangan, dan dokumentasi sistem perangkat lunak[7]. Beberapa model diagram yang umum digunakan dalam *UML* mencakup:

a) *Use Case* Diagram

Use case adalah serangkaian deskripsi atau uraian terkait yang membentuk sistem secara terstruktur dan dilaksanakan atau diawasi oleh aktor tertentu [17]. Simbol-simbol pada diagram *use case* digambarkan dalam Tabel 2.2.

Tabel 2. 2 Simbol *Use Case* Diagram

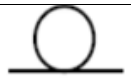
No	Simbol	Nama	Keterangan
1		Aktor	Menggambarkan peran orang atau tokoh yang ada dalam sistem, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*.
2		*Use case*	Interaksi antara sistem dan aktoryang dinyatakan dengan kata kerja.
3		Asosiasi	Penghubung antara aktor dengan *use case*.
4		Generalisasi	Spesialisasiaktor untuk dapat berpartisipasi dengan *use case*.

5		*Include*	Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya.
6		*Extend*	Menunjukkan bahwa suatu *use case* merupakan tambahan fungsionalitas dari *use case* lainnya jika suatu kondisi terpenuhi

b) *Sequence* Diagram

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan[8]. Berikut ini simbol-simbol pada sequence diagram dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol *Sequence Diagram*

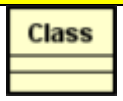
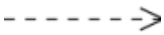
No	Simbol	Nama	Keterangan
1		*Actor*	Menggambarkan sebuah objek atau tokoh yang berinteraksi dengan sistem.
2		*Entitiy Class*	Menunjukkan hubungan yang akan dilakukan dalam sistem.
3		*BounderyClass*	Menggambarkan scope atau biasanya berupa form.
4		*Control Class*	Menggambarkan hubungan antara boundary dengan tabel.
5		*Life Line*	Menggambarkan tempat dimulai dan berakhirnya *message*.
6		*Message*	Menggambarkan terjadinya pengiriman

			pesan.

c) *Class Diagram*

Class diagram adalah representasi visual dari struktur dan penjelasan kelas, paket, dan objek, beserta hubungan-hubungan seperti containment, pewarisan, asosiasi, dan sebagainya[9]. Simbol-simbol yang digunakan dalam diagram kelas dijelaskan pada Tabel 2.4

Tabel 2. 4 *Simbol Class* Diagram

No	Simbol	Nama	Keterangan
1		*Class*	Penggambaran nama kelas, atribut, atau properti atau data danmethod
2		*Association*	Hubungan antara objeksatu dengan objek lainnya.
3		*Navigable Association*	Relasi antar kelas yangmemilik arti suatu kelas digunakan oleh kelas lain.
4		*Generalization*	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
5		*Dependency*	Relasi antar kelas dengan kebergantungan antarkelas
6		*Aggregation*	Relasi antar kelas dengansemua bagian (*whole- part*).

2.4. Pemrograman Berbasis Objek (PBO)

Pemrograman Berbasis Objek (PBO) adalah strategi pembangunan perangkat lunak yang mengorganisir perangkat lunak sebagai kumpulan objek yang mengandung data dan operasi yang dapat diterapkan pada objek tersebut. PBO dapat dijelaskan sebagai paradigma atau teknik pemrograman yang berfokus pada objek[10]. Metodologi pengembangan sistem yang berorientasi objek melibatkan beberapa konsep dasar yang perlu dipahami, antara lain:

a. Kelas (*Class*)

   Kelas adalah kumpulan objek dengan karakteristik yang sama. Sebuah kelas memiliki atribut, operasi/*method*, relationship dan arti. Suatu kelas dapat diturunkan ke kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

b. Objek (*Object*)

   Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi dan dimusnahkan.

c. Metode (*Method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi berfungsi untuk memanipulasi objek itu sendiri.

d. Atribut (*Attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara 14 individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga enkasulapsi.

e. Enkapsulasi (*Encapsulation*)

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

f. Pewarisan (*Inheritance*)

   Mekanisme yang memungkin suatu objek mewarisi sebagian atau seluruh defenisi dan objek lain sebagai bagian dari dirinya.

g. Antarmuka (*Interface*)

   Antarmuka atau interface biasa digunakan agar kelas lain tidak dapat mengakses langsung ke suatu kelas, melainkan hanya mengangkses antarmukanya.

h. *Reusabilily*

   Pemanfaatan kembali objek yang sudah didefenisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut. Misalkan pada sebuah aplikasi peminjaman buku diperlukan pada kelas anggota, maka ketika membuat aplikasi *VCD,* kelas anggota ini bisa digunakan kembali dengan sedikit perubahan untuk aplikasi penyewaan *VCD* tanpa harus membuat dari awal lagi.

i. Komunikasi Antar Objek

Komunikasi antara objek dilakukan melalui pesan (*message*) yang dikirim dari satu objek ke objek yang lainnya.

j. Polimorpisme (*Polymorphism*)

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

k. *Package*

Package adalah sebuah kontainer yang dapat digunakan untuk mengelompokan kelas-kelas sehingga memungkinkan beberapa kelas bernama sama disimpan dalam package yang berbeda.

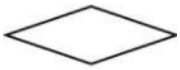
2.5. Database

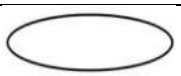
Basis data atau database adalah kumpulan data yang terhubung secara logis, digunakan bersama, dan dirancang untuk memenuhi kebutuhan informasi suatu organisasi [23]. Dalam penelitian Sistem Informasi Peminjaman Barang di SMA Negeri 1 Karangnunggal ini, Maria DB dipilih sebagai sistem basis data yang akan digunakan. Terdapat pula diagram ERD yang dapat digunakan sebagai alat bantu dalam proses pembuatan database.

1. ERD (Entity Relationship Diagram)

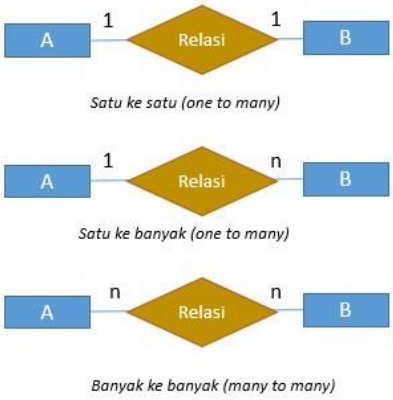
Diagram Entitas Hubungan (*ERD),* atau *Entity Relationship Diagram*, adalah model yang digunakan untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki keterkaitan antar relasi. Berikut adalah simbol-simbol yang umum digunakan dalam ERD[11]. Simbol *Entity Relationship Diagram* dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol Entity Relationship Diagram[11]

No	Simbol	Nama	Keterangan
1		Entitas	Kumpulan objek yang dapat diidentifikasi secara unik.
2		Relasi	Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan yang ada meliputi: satu ke satu, satu ke

			banyak,dan banyak ke banyak.
3		Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
4		Garis	Hubungan antara entitas dan atributnya dan himpunanentitas dengan himpunan relasi.

Derajat relasi atau kardinalitas adalah hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Berikut macam – macam relasi dapat dilihat pada Gambar 2.2:



Gambar 2. 2 Derajat Relasi atau Kardinalitas

a. *One to one*

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B begitu pula sebaliknya.

b. *One to many*

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

c. Many to many

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

2. DBMS (Database Management System)

   Database Management System (DBMS) adalah perangkat lunak yang dirancang untuk mengendalikan proses pembuatan, pemeliharaan, pengolahan, dan pemanfaatan data dalam skala besar. Penggunaan SMBD saat ini menjadi sangat penting dalam berbagai bidang, baik itu dalam skala besar maupun kecil[12]. Berikut adalah beberapa contoh SMBD yang umum digunakan:

   1. Oracle
   2. MySQL
   3. PostgreSQL

(~~*Halaman ini sengaja dikosongkan*~~)