



# **BAB II**

# **DASAR TEORI**

## **BAB II DASAR TEORI**

### **2.1 Sistem Informasi**

Sistem informasi dalam arti yang luas dapat dipahami sebagai gabungan dari berbagai komponen-komponen seperti perangkat lunak, perangkat keras, dan juga prosedur-prosedur yang bekerja secara terstruktur untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang relevan untuk mencapai tujuan tertentu. Beberapa karakteristik yang harus dimiliki sebuah sistem yaitu [1]:

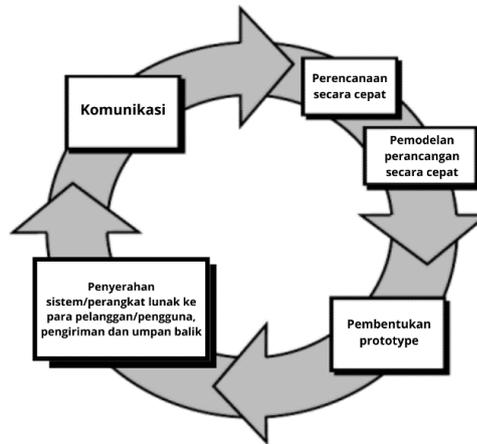
1. Komponen (*Component*) Sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan. Komponen sistem dapat berupa sub sistem atau bagian-bagian dari sistem.
2. Batasan Sistem (*Boundary*) Daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luar dinamakan dengan batasan sistem. Batasan sistem ini memungkinkan sistem dipandang sebagai satu kesatuan dan juga menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
3. Lingkungan Luar Sistem (*Environment*) Apapun yang berada di luar batas dari sistem dan mempengaruhi sistem tersebut dinamakan dengan lingkungan luar sistem. Lingkungan luar yang bersifat menguntungkan wajib dipelihara dan yang merugikan harus dikendalikan agar tidak mengganggu kelangsungan sistem.
4. Penghubung Sistem (*interface*), media penghubung diperlukan untuk mengalirkan sumber-sumber daya dari sub sistem ke sub sistem lainnya dinamakan dengan penghubung sistem.
5. Masukan Sistem (*input*), energi yang dimasukkan ke dalam sistem dinamakan dengan masukan sistem (*input*) dapat berupa perawatan dan masukan sinyal. Perawatan ini berfungsi agar sistem dapat beroperasi dan masukan sinyal adalah energi yang diproses untuk menghasilkan keluaran (*output*).
6. Keluaran Sistem (*output*) Hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dinamakan dengan keluaran sistem (*output*). Informasi merupakan contoh keluaran sistem.
7. Pengolah sistem, untuk mengolah masukan menjadi keluaran diperlukan suatu pengolah yang dinamakan dengan pengolah sistem.
8. Sasaran sistem, sistem pasti memiliki tujuan atau sasaran yang sangat menentukan input yang dibutuhkan oleh sistem dan keluaran yang dihasilkan.

### **2.2 Rekayasa Perangkat Lunak**

Rekayasa perangkat lunak (RPL), sebagai proses pembangunan menggunakan prinsip rekayasa, bertujuan menghasilkan perangkat lunak dengan nilai ekonomi, kehandalan, dan efisiensi dalam penggunaan mesin. Di dalamnya, Rekayasa Web menyajikan pendekatan sistematis, disiplin, dan terukur dalam pengembangan, operasi, dan pemeliharaan aplikasi berbasis web. Rekayasa web juga merujuk pada subdisiplin rekayasa perangkat lunak yang memberikan metodologi untuk perancangan, pengembangan, pemeliharaan, dan keterlibatan aplikasi web [13]. Berikutnya, metode pengembangan sistem dan pengujian sistem yang akan diterapkan

## A. Metode Pengembangan Sistem

Pengembangan metode pengembangan sistem yang akan digunakan pada sistem ini yaitu metode pengembangan perangkat lunak Software Development Life (SDLC) model *prototype*. Metode ini membantu pengembang dalam berkomunikasi dengan pengguna supaya mendapatkan hasil yang sesuai dengan keinginan pengguna melalui rancangan tampilan fungsionalitas aplikasi. Berikut ini adalah tahapan metode *prototype* menurut Pressman (2012) [3]. Dalam Gambar 2.1.



**Gambar 2. 1** Metode *Prototype*

Berdasarkan model *prototype* yang telah digambarkan di atas, dapat diuraikan pembahasan disetiap tahapannya sebagai berikut [3]:

### 1. Komunikasi

Pada tahap ini, penulis berkolaborasi dengan pengguna untuk mengidentifikasi kebutuhan dan tujuan utama sistem yang akan dikembangkan. Informasi yang diperoleh digunakan untuk merancang dan mengembangkan prototipe awal yang sederhana. Tujuan dari tahap ini adalah untuk memahami kebutuhan dan ketentuan secara jelas, serta sebagai dasar referensi dalam pengembangan *prototype* secara menyeluruh.

### 2. Perencanaan secara Cepat

Pada tahap ini, dilakukan perencanaan awal dalam pembuatan *prototype* pertama yang akan digunakan sebagai gambaran awal bagi pengguna. Meskipun *prototype* awal ini belum lengkap, namun upaya dilakukan untuk memenuhi sebanyak mungkin kebutuhan pengguna. Tahapan iterasi selanjutnya pada tahap ini melibatkan perencanaan singkat guna menghasilkan *prototype* yang sesuai dengan kebutuhan pengguna.

### 3. Pemodelan Perancangan secara Cepat

Dalam tahap ini, dilakukan pemodelan alur dan desain sistem dengan ringkas. Tujuan dari tahap pemodelan ini adalah untuk menyajikan gambaran alur dan antarmuka perangkat lunak yang dapat dilihat oleh pengguna.

#### 4. Pembentukan *Prototype*

Tahap ini digunakan untuk merancang dengan cara mengkodekan sistem dan menguji sistem yang sedang dikembangkan, meskipun masih belum memiliki fungsi yang sempurna dan belum memiliki fitur yang lengkap. *Prototype* pada iterasi selanjutnya akan diperbaiki sesuai dengan kebutuhan dari pengguna.

#### 5. Penyerahan sistem/ perangkat lunak ke para pelanggan/ pengguna,

*Prototype* kemudian diserahkan kepada pengguna untuk dievaluasi, dengan tujuan mendapatkan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Iterasi terjadi ketika penulis melakukan perbaikan terhadap *prototype* berdasarkan umpan balik yang diterima.

### 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 pemrograman 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 [10].

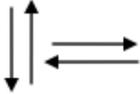
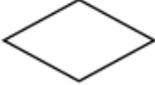
Adapun struktur data merupakan cara menyimpan atau mempresentasikan data didalam komputer agar bisa dipakai secara efisien. Berikut bagian-bagian yang ada pada struktur data:

#### A. *Flowchart*

*Flowchart* atau diagram alir adalah jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi secara berurutan dalam suatu sistem. Analisis 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 [13]. Berikut adalah contoh simbol-simbol yang umum digunakan dalam *flowchart* diagram [14]. Simbol *Flowchart* dapat dilihat pada Tabel 2.1:

**Tabel 2. 1** Simbol *Flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminal</i>	Awal atau akhir dari suatu prosedur.
2		Proses	Proses operasional yang terjadi dalam komputer.

3		<i>Output/Input</i>	Memasukkan atau mengeluarkan data tanpa melihat jenis perangkat yang digunakan.
4		Garis Alur	Menggambarkan alur atau aliran program.
5		<i>Decision</i>	Menandakan suatu kondisi yang mengarahkan pada dua kondisi YA/TIDAK.
6		Simbol Dokumen	Untuk mencetak dokumen berpakertas.
7		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer

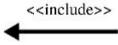
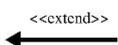
## B. Unified Modelling Language (UML)

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 [16]. 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 <i>use case</i> .
2		<i>Use case</i>	Interaksi antara sistem dan aktor yang dinyatakan dengan kata kerja.
3		Asosiasi	Penghubung antara aktor dengan <i>use case</i> .
4		Generalisasi	Spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
5		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

### 2.3 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 [15]. 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 individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privat untuk menjaga enkapsulasi.

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 memungkinkan 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 mengakses 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.

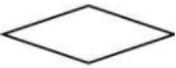
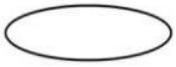
## 2.4 Database

Database atau basis data merupakan kumpulan data yang terhubung secara logis yang dipakai bersama dan deskripsi dari data ini dirancang untuk memenuhi kebutuhan informasi sebuah organisasi [23]. Pada penelitian Sistem Informasi Pengajuan Jabatan Fungsional Dosen ini, database yang akan digunakan ialah MariaDB. Adapun diagram ERD yang dapat digunakan sebagai alat bantu dalam pembuatan database.

### 1. ERD (Entity Relationship Diagram)

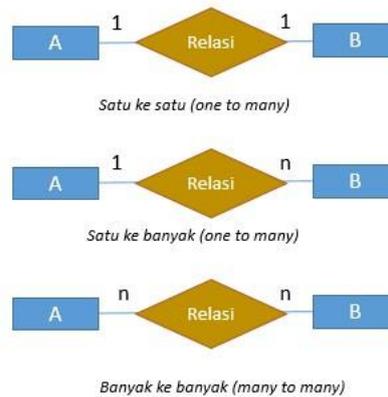
Diagram Entitas Hubungan (ERD), atau Entity Relationship Diagram, adalah model yang digunakan untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki keterkaitan antar relasi. Berikut adalah simbol-simbol yang umum digunakan dalam ERD [24]. Simbol *Entity Relationship Diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2. 3** Simbol *Entinty Relationship Diagram*

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 himpunan entitas 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 (DBMS)* adalah perangkat lunak yang dirancang untuk mengendalikan proses pembuatan, pemeliharaan, pengolahan, dan pemanfaatan data dalam skala besar. Penggunaan SMD saat ini menjadi sangat penting dalam berbagai bidang, baik itu dalam skala besar maupun kecil [25]. Berikut adalah beberapa contoh SMD yang umum digunakan:

1. Oracle
2. MySQL
3. PostgreSQL

## 2.5 Jabatan Fungsional

Jabatan fungsional dosen adalah pengakuan, penghargaan, dan kepercayaan atas kompetensi, kinerja, integritas dan tanggung jawab dalam pelaksanaan tugas. Oleh sebab itu, sebaiknya para dosen mengajukan kenaikan pangkat akademik, sehingga jenjang karir para dosen dapat berasosiasi positif dengan pangkat (golongan) dan masa kerjanya [4].

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