

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Penelitian tentang Sistem Informasi sebelumnya pernah dilakukan oleh Suhartini dengan judul “Sistem Informasi Berbasis Web SMA Al-Mukhtariyah Mamben Lauk Berbasis *PHP* dan *MySQL* Dengan *Framework Codeigniter*”. Penelitian tersebut dilakukan untuk mengembangkan Sistem Informasi berbasis web pada SMA Al-Mukhtariyah mengingat sekolah tersebut belum memiliki *website*. Penelitian tersebut menggunakan bahasa pemrograman *PHP* dengan *Framework Codeigniter* dan *MySQL* sebagai *DBMS*[3].

Penelitian tentang Sistem Informasi juga pernah dilakukan oleh Iyus Supriadi tentang Himpunan Mahasiswa Islam Cabang Jayapura berbasis web. Penelitian tersebut dilakukan untuk mengembangkan sistem informasi Himpunan Mahasiswa Islam Cabang Jayapura yang mana penyampaian informasi tersebut masih menggunakan brosur. Sistem tersebut dikembangkan menggunakan bahasa pemrograman *PHP*. Sistem tersebut dapat menampilkan data anggota, profil, berita, galeri, pengguna, dan data *inbox* pada Himpunan Mahasiswa Islam Cabang Jayapura[4].

Penelitian tentang Sistem Informasi juga pernah dilakukan oleh Sugeng Priyanto tentang pendaftaran mahasiswa baru yang ada pada Politeknik Sains & Teknologi Wiratama Maluku Utara berbasis web. Penelitian tersebut dilakukan untuk mengembangkan sistem informasi pendaftaran mahasiswa yang ada di Politeknik Sains & Teknologi Wiratama Maluku Utara yang mana pendaftaran mahasiswa masih dilakukan secara manual yang mana calon mahasiswa harus ke kampus untuk melakukan pendaftaran sehingga kurang efektif dan banyak membuang waktu bagi calon mahasiswa dan panitia penerimaan calon mahasiswa. Penelitian tersebut menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*. Hasil penelitian tersebut dapat mempermudah admin dalam pengolahan dan pengelolaan data calon mahasiswa baru. Selain itu, aplikasi sistem informasi pendaftaran calon mahasiswa baru berbasis *website* dapat membantu panitia dalam penyeleksian calon mahasiswa baru untuk hal-hal yang berkaitan dengan pengaksesan data dan pemroses data.[5].

Melalui beberapa tinjauan pustaka, penulis akan mengembangkan Sistem Informasi Himpunan Mahasiswa Teknik Informatika (HMTI) Berbasis Web. Sistem tersebut selain menampilkan data profil HMTI Politeknik Negeri Cilacap juga dapat melakukan pendaftaran kepengurusan HMTI PNC secara *online*. Penelitian ini menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*.

## **2.2 Landasan Teori**

### **2.2.1 Sistem Informasi**

#### **a) Definisi Sistem**

Sistem yaitu kumpulan prosedur yang saling berhubungan untuk menyelesaikan masalah tertentu, dan juga kumpulan elemen yang berhubungan yang bertanggung jawab atas proses pemasukan (*input*) sehingga menghasilkan keluaran(*output*)[4].

#### **b) Definisi Informasi**

Informasi yaitu hasil dari pengolahan data yang berguna bagi penerima berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan. Informasi sendiri secara etimologi berasal dari Bahasa Perancis Kuno *informacion* yang diambil dari Bahasa Latin *informationem* yang artinya “garis besar, konsep, ide”.

#### **c) Definisi Sistem Informasi**

Menurut Taufiq sistem informasi adalah “kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna”[5].

### **2.2.2 Pendaftaran Online**

Pendaftaran *Online* adalah pencatatan data mulai dari nama, tempat tanggal lahir, alamat, dan informasi lainnya dengan menggunakan media *online* yang tentunya harus terhubung dengan internet. Selain itu pada saat pendaftaran *online*, pendaftar juga dapat melakukannya di jarak jauh tanpa harus mendatangi tempat pendaftaran[6].

### **2.2.3 Tentang HMTI**

HMTI atau yang disingkat Himpunan Mahasiswa Teknik Informatika yaitu suatu organisasi yang menghimpun mahasiswa pada

Jurusan Teknik Informatika di Politeknik Negeri Cilacap. Urutan kepengurusan HMTI yaitu Ketua, Wakil Ketua, Sekretaris, Bendahara, dan 5 divisi yang terdiri dari Divisi Ristik, PSDM(Pengembangan Sumber Daya Mahasiswa), Humas(Hubungan Masyarakat), Koinfo, dan MU(Manajemen Usaha). Ketua, Wakil Ketua, Sekretaris, Bendahara, dan kepala dari setiap divisi diduduki oleh mahasiswa Teknik Informatika tahun kedua, sedangkan anggota-anggota dari setiap divisi diduduki oleh mahasiswa Teknik Informatika tahun pertama.



#### 2.2.4 Rekayasa Perangkat Lunak


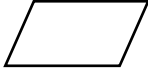

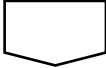

Rekayasa perangkat lunak adalah disiplin rekayasa dengan hasil abstrak, dimana hasil proses rekayasa memberikan bentuk nyata. Pengukuran *volume* sebuah hasil rekayasa yang abstrak menjadi tidak semudah pengukuran *volume* hasil rekayasa yang memberikan hasil konkret, Pengukuran *volume* pekerjaan rekayasa perangkat lunak yang abstrak hanya didasarkan kepada kegunaan dari hasil rekayasa.. Sedangkan menurut *IEEE*, definisi rekayasa perangkat lunak adalah aplikasi dari sebuah pendekatan disiplin dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak yaitu aplikasi dari rekayasa perangkat lunak.[7].

#### 2.2.5 Flowchart

Kegunaan *flowchart* yaitu untuk menyajikan kegiatan manual, kegiatan pemrosesan komputer atau keduanya. Tabel di bawah ini akan menjelaskan simbol-simbol dari *flowchart*[8].

**Tabel 2. 1** Simbol-simbol *Flowchart*

Nama Simbol	Gambar	Keterangan
Terminal		Untuk menunjukkan sumber atau tujuan dari dokumen dan laporan
Operasi Manual		Untuk menunjukkan operasi manual dari suatu flowchart

Dokumen Manual		Untuk menunjukkan dokumen manual dari suatu flowchart
Data		Untuk menunjukkan data dari suatu flowchart
Konektor halaman		Untuk mengkoneksikan antar simbol dalam satu halaman
Konektor halaman lain		Untuk mengkoneksikan antar halaman dalam satu flowchart
Jalur dokumen		Untuk menunjukkan alur dokumen dari suatu flowchart

### 2.2.6 Basis Data

#### A. Pengertian Basis Data

Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data dalam sistem informasi yaitu memasukkan, menyimpan, dan mengambil data[9].

#### B. Database Management System

*Database Management System* (DBMS) adalah aplikasi untuk mengelola dan menampilkan data. Sekitar tahun 1960 *DBMS* mulai berkembang, lalu sekitar tahun 1970 berkembanglah *Relational Database Management System* (*RDBMS*) oleh Edgar J. Cold yaitu suatu model basis data yang mana ada relasi antar tabel[9].

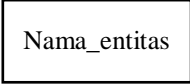
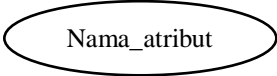
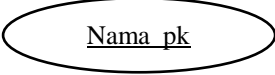
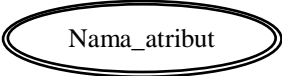


#### C. Structured Query Language

Sekitar tahun 1970 *SQL* mulai berkembang, kemudian pada tahun 1986 sudah digunakan sebagai standar resmi oleh ANSI (*American National Standards Institute*) dan oleh ISO (*International Organization for Standardization*) pada tahun 1987[9].

#### D. Entity Relationship Diagram

*Entity Relationship Diagram* yaitu diagram yang digunakan dalam basis data untuk memberi gambaran suatu relasi dalam basis data yang akan dibuat. Salah satu notasi yang biasa dipakai yaitu *Chen notation*. Tabel di bawah ini akan menjelaskan simbol-simbol ERD dengan *Chen notation*[9].

**Tabel 2. 2** Simbol-simbol ERD

Nama	Gambar	Keterangan
Entitas / <i>Entity</i>		Entitas adalah data yang akan masuk ke basis data berupa tabel
Atribut / <i>Attribute</i>		Atribut adalah kolom yang dibutuhkan suatu entitas
Atribut Kunci Primer / <i>Primary Key Attribute</i>		Kolom yang memiliki <i>primary key</i> (biasanya berupa id kolom) yang nantinya digunakan untuk relasi dengan tabel lain
Atribut Multinilai / <i>Multivalued Attribute</i>		Kolom yang memiliki lebih dari satu nilai yang dibutuhkan yang dibutuhkan suatu entitas
Relasi		Relasi adalah penghubung antara entitas satu dengan yang lain
Asosiasi		Melambungkan hubungan antara

		entitas dengan entitas lainnya atau entitas dengan relasi.
--	--	--

### 2.2.7 Metode *Prototype*

Salah satu metode pengembangan yang diterapkan pada penelitian yaitu Metode *Prototype* yaitu metode yang mana akan menghasilkan sebuah versi awal dari sistem yang disebut *Prototype* yang bisa menjadi penengah antara pengembang dengan pengguna agar dapat berkomunikasi tentang proses pengembangan sistem[2]. Manfaat yang dimiliki oleh Metode *Prototype* yaitu [2]:

- a) Menghasilkan sistem yang sebenarnya dalam sebuah replika sistem yang akan berjalan, menampung masukan dari pengguna untuk kesempurnaan sistem.
- b) Pengguna akan lebih siap menerima setiap perubahan sistem yang berkembang sesuai dengan berjalannya prototype sampai dengan hasil akhir.
- c) Prototype dapat ditambah maupun dikurangi sesuai berjalannya proses pengembangan.
- d) Penghematan sumberdaya dan waktu dalam menghasilkan produk yang lebih baik dan tepat guna bagi pengguna.

Ada tahap-tahap dalam menjalankan Metode Prototyping yaitu [2], [10]:

- a) Identifikasi  
Pada tahap ini pelanggan dan pengembang bertemu untuk menentukan keseluruhan dari sistem yang akan dibuat seperti tujuan dibuatnya sistem, identifikasi kebutuhan yang diperlukan untuk pengembangan sistem.
- b) Desain Cepat  
Pada tahap ini dilakukan pendesainan yang diperlukan untuk pengembangan sistem seperti *UML*, *Mock-Up* secara cepat
- c) Mengembangkan *Prototype*  
Pada tahap ini dibuat prototype untuk sistem yang akan dibangun.
- d) Evaluasi dan perbaikan dengan pelanggan  
Setelah mengembangkan *prototype* lalu dilakukan evaluasi dengan pelanggan, jika masih ada yang perlu diperbaiki maka dilakukan perbaikan.

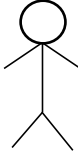
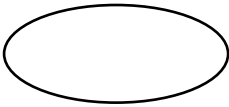

### 2.2.8 Unified Modeling Language

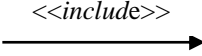
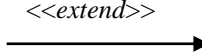
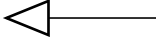
*Unified Modeling Language* (UML) yaitu model yang berbentuk gambar yang berguna untuk memberi gambaran dari sistem yang berorientasi objek [11]. Diagram yang biasa dipakai dalam UML antara lain [7], [12]:

#### a) *Use Case Diagram*

*Use Case Diagram* adalah diagram yang dibuat untuk memberi gambaran tentang interaksi antara aktor dengan sistem yang akan dikembangkan. *Use Case Diagram* memiliki simbol-simbol yang akan dijelaskan pada tabel berikut [9].

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


Nama	Gambar	Keterangan
<i>Actor</i>		<i>Actor</i> adalah salah satu komponen <i>Use Case Diagram</i> yang merepresentasikan <i>User</i> yang berinteraksi dengan sistem
<i>Use Case</i>		<i>Use Case</i> adalah salah satu komponen <i>Use Case Diagram</i> yang merepresentasikan proses-proses yang berhubungan langsung dengan <i>actor</i> , walaupun ada beberapa <i>Use Case</i> yang tidak berhubungan langsung dengan <i>Actor</i> .
<i>Association</i>		<i>Association</i> adalah salah satu komponen <i>Use Case Diagram</i> untuk

		menghubungkan <i>Actor</i> dengan <i>Use Case</i> .
<i>Include</i>		<i>Include</i> yaitu relasi pada <i>Use Case Diagram</i> yang mana ketika suatu <i>use case</i> hendak dijalankan perlu adanya <i>use case</i> lain yang berkaitan dengan <i>use case</i> yang akan dijalankan
<i>Extend</i>		<i>Extends</i> yaitu relasi pada <i>Use Case</i> yang mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
<i>Generalization</i>		<i>Generalization</i> adalah relasi pada <i>Use Case</i> yang mana <i>use case</i> yang ditambahkan merupakan bagian dari <i>use case</i> lain






b) *Class Diagram*

*Class Diagram* yaitu diagram yang dibuat untuk memberi gambaran struktur kelas, *package*, dan penghubung seperti. Tabel di bawah ini akan menjelaskan komponen-komponen yang ada dalam *Class Diagram*[9].

**Tabel 2. 4** Simbol-simbol *Class Diagram*

Nama	Simbol	Keterangan
<i>Class / Kelas</i>		Nama kelas, atribut, dan operasi yang

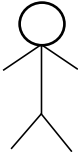


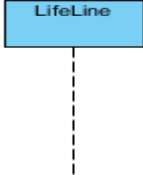
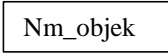

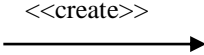
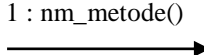
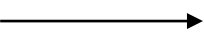
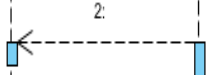
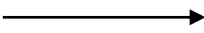
	<table border="1"> <tr><td>+atribut</td></tr> <tr><td>+operasi()</td></tr> </table>	+atribut	+operasi()	ada dalam <i>class diagram</i>
+atribut				
+operasi()				
<i>Interface</i> / Antarmuka		Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek		
<i>Association</i> / Asosiasi		Relasi antarkelas yang bermakna umum		
<i>Direct Association</i> / Asosiasi Berarah		Relasi antarkelas yang bermakna kelas yang satu digunakan oleh kelas yang lain.		
<i>Generalization</i> / Generalisasi		Relasi antarkelas yang bermakna generalisasi / spesialisasi		
<i>Aggregation</i> / Agregasi		Relasi antarkelas yang dengan makna semua-bagian ( <i>whole-part</i> )		

c) *Sequence Diagram*

*Sequence Diagram* yaitu diagram yang dibuat untuk memberi gambaran tentang urutan waktu interaksi antara objek dengan sistem. Tabel di bawah ini akan menjelaskan komponen-komponen yang ada dalam *Sequence Diagram* [9].

**Tabel 2. 5** Simbol-simbol *Sequence Diagram*

Nama	Gambar	Keterangan
Aktor / <i>Actor</i>		Aktor adalah yang akan berinteraksi dengan sistem, bisa berupa orang atau sistem

Garis hidup / <i>LifeLine</i>		Menyatakan kehidupan suatu objek
Objek		Simbol ini menyatakan objek yang berinteraksi dengan aktor
Waktu aktif		Simbol ini menyatakan waktu aktif pada suatu objek. Biasanya waktu aktif menyatakan tahapan-tahapan yang dijalankan oleh suatu objek.
Pesan <i>create</i>		Simbol ini menyatakan objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan <i>call</i>		Menyatakan suatu objek memanggil metode / operasi yang ada pada objek lain atau dirinya sendiri.
Pesan <i>send</i>		Simbol ini menyatakan suatu objek mengirimkan data.
Pesan <i>return</i>		Simbol ini menyatakan suatu objek mengirimkan pesan kembali.
Pesan <i>destroy</i>		Simbol ini menyatakan suatu objek mengakhiri hidup objek lain.

### 2.2.9 Rekayasa Web

#### A. Pengertian Rekayasa Web

Rekayasa Web adalah suatu aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis Web (*Web-based application*). Rekayasa Web juga merupakan subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi web[13].

#### B. Website

*Website* adalah halaman-halaman yang memuat informasi yang berupa gambar diam atau gerak, teks, suara, animasi, video, dan atau gabungan dari semuanya yang saling berhubungan dengan jaringan halaman (*hyperlink*) untuk membentuk suatu sistem berbasis web. *Website* mempunyai dua sifat yaitu statis dan dinamis yang mana jika informasi yang ada dalam suatu *website* tetap / tidak dapat diubah, maka *website* tersebut bersifat statis, sedangkan jika informasi yang ada dalam *website* bisa tidak tetap / dapat dirubah, maka *website* tersebut bersifat dinamis[7].

#### C. Web Server

Definisi *web server* menurut Betha Sidik dan Husni I. Pohan yaitu komputer yang akan digunakan untuk menyimpan dokumen-dokumen web[11].

#### D. Hypertext Markup Language(HMTL)

*Hypertext Markup Language (HMTL)* adalah bahasa markup yang biasanya dipakai untuk menyusun halaman web terutama tampilan. Biasanya kode-kode *HMTL* berupa tag yang diawali dengan "<" dan diakhiri dengan ">"[5].

#### E. Hypertext Preprocessor (PHP)

*Hypertext Preprocessor* adalah bahasa pemrograman *server side* untuk pengembangan web. PHP dikembangkan oleh Rasmus Lerdorf pada tahun 1995[3]. PHP memiliki ciri-ciri yaitu [7]:

- Kode yang ada pada PHP hanya bisa dijalankan pada web server, contohnya *Apache* dan *XAMPP*
- Kode *PHP* diletakkan dan dijalankan di *web server*

- Kode *PHP* dapat dipakai untuk mengakses basis data seperti *MySQL*, *postgreSQL*, *Oracle*, dll.

### 2.2.10 Pemrograman Berorientasi Objek

Metode berorientasi objek yaitu bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi pemrograman berorientasi objek, dan pengujian berorientasi objek. Metode berorientasi objek mempunyai beberapa keuntungan yaitu [9]:

- Meningkatkan produktifitas  
Metodologi berorientasi objek dapat meningkatkan produktifitas karena kelas dan objek tidak hanya dipakai pada satu masalah tapi bisa digunakan juga pada masalah lainnya
- Kecepatan dalam pengembangan  
Pengembangan sistem dengan metodologi berorientasi objek menjadi semakin cepat karena kesalahan pada saat pengkodean berkurang. Hal tersebut terjadi karena sistem dibangun dengan baik dan benar saat analisis dan perancangan.
- Kemudahan dalam pemeliharaan  
Pola-pola yang cenderung tetap dan stabil dapat dipisahkan pola-pola yang mungkin sering berubah-ubah dengan adanya model objek
- Adanya konsistensi  
Adanya sifat pewarisan dan penggunaan notasi yang sama saat analisis, perancangan, dan pengkodean menjadikan metodologi ini konsisten
- Meningkatkan kualitas perangkat lunak

Dalam metodologi berorientasi objek ada konsep dasar yang harus dipahami yaitu [9]:

- *Class* / kelas  
Kumpulan objek-objek dengan karakteristik sama yaitu *class*. Pada *class* memiliki atribut, metode / operasi, dan *relationship* / hubungan.
- *Object* / objek  
Objek adalah abstraksi yang mewakili dunia nyata seperti manusia, benda, tempat, organisasi, tempat, struktur, kejadian, status, dan

lain-lain. Secara teknis objek hanya akan muncul saat program dijalankan dan jika belum dijalankan maka disebut kelas.

- *Method* / metode  
Metode adalah prosedur yang ada dalam suatu objek dalam sistem dengan metodologi berorientasi objek
- *Attribute* / atribut  
Atribut adalah variabel global yang dimiliki oleh suatu kelas yang berupa nilai atau elemen-elemen yang dimiliki objek, dan dimiliki secara individual oleh objek, misal
- *Abstraction* / abstraksi  
Abstraksi yaitu prinsip untuk membentuk model yang sederhana dari dunia nyata yang kompleks dengan mengabaikan aspek aspek lain yang tidak sesuai dengan permasalahan
- *Encapsulation* / enkapsulasi  
Enkapsulasi adalah salah satu dari konsep metodologi berorientasi objek yang mana atribut dan metode yang dimiliki oleh objek dibungkus untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.
- *Inheritance* / pewarisan  
Pewarisan yaitu mekanisme yang mana memungkinkan suatu objek mewariskan sebagian atau seluruh definisi kepada objek warisannya.
- *Interface* / antarmuka  
Antarmuka adalah suatu konsep dalam pendekatan berorientasi objek yang sangat mirip dengan kelas, hanya saja antarmuka tidak memiliki atribut dan metode yang dideklarasikan tidak berisi, melainkan dapat diimplementasikan ke kelas lain. Metode pada antarmuka yang diimplementasikan pada suatu kelas harus sama persis dengan metode yang ada di antarmuka.
- *Reusability*  
*Reusability* adalah suatu konsep yang mana objek yang telah didefinisikan untuk suatu permasalahan dapat dipakai lagi untuk permasalahan lainnya yang mana objek tersebut terlibat di dalamnya.
- Generalisasi dan Spesialisasi  
Generalisasi menunjukkan hubungan kelas dan objek secara umum sedangkan spesialisasi menunjukkan hubungan kelas dan objek secara khusus.

- Komunikasi antar objek  
Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.
- *Polymorphism* / Polimorfisme  
Polimorfisme yaitu suatu konsep pada metodologi berorientasi objek yang mana suatu objek dapat digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.
- *Package*  
Package adalah wadah yang digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang sama namanya disimpan ke dalam package yang berbeda.

### **2.2.11 Metode Pengujian *Black Box***

*Black Box* adalah salah satu metode pengujian sistem yang menitikberatkan pada segi fungsionalitas yaitu dengan menentukan kesuksesan sistem dengan melakukan perbandingan antara input yang diberikan dari ke dalam sistem dengan output yang keluar dari sistem apakah sesuai dengan perencanaan atau belum [14]. Yang diujikan pada oleh metode *Black Box* pada penelitian ini yaitu kesesuaian antara input dengan output pada *login*, berita HMTI, pengurus, anggota, akun pengelola, jabatan, tugas jabatan, visi, misi, jadwal pendaftaran, tahapan seleksi, biodata, foto profil, alasan mendaftar, pengalaman organisasi, mendaftar, membatalkan mendaftar, data pendaftar, penetapan kepengurusan, hasil seleksi, dan pembatalan kepengurusan (selengkapnya bisa dilihat pada Bab III).