

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Informasi**

Menurut (Gordon B, 1991), Sistem informasi adalah suatu sistem yang menerima input atau masukan data dan instruksi, mengolah sesuai dengan instruksi dan mengeluarkan hasilnya.

Menurut (Lani, 1995), Sebuah sistem informasi adalah sistem buatan manusia yang berisi himpunan terintegrasi dari komp komponen manual dan komponen – komponen terkomputerisasi yang bertujuan untuk mengumpulka memproses data, dan menghasilkan informasi untuk pemakai.

#### **2.2 PHP**

Menurut (Arief, 2011), PHP adalah Bahasa *server-side –scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan server-side-scripting maka sintaks dan perintah-perintah PHP akan dieksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan format HTML

Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web.

#### **2.3 Framework Codeigniter**

*Framework* dalam Bahasa Indonesia dapat diartikan sebagai kerangka kerja. Kerangka kerja merupakan kumpulan dari fungsi atau prosedur dan *class* untuk tujuan tertentu yang sudah siap digunakan sehingga dapat mempermudah dan mempercepat pekerjaan seorang programmer.

Codeigniter merupakan sebuah framework yang berbasiskan PHP yang bersifat *open source* dan menggunakan model MVC (*Model, View, Controller*) yang

difungsikan untuk memudahkan pengembang web untuk membuat web yang dinamis agar lebih cepat dan mudah.

## 2.4 MVC

MVC adalah kependekan dari *Model*, *View*, *Controller* yang merupakan konsep pengembangan website dengan memisahkan beberapa komponen utama yaitu *Model* yang digunakan untuk manipulasi data; dan *View* untuk mengatur antarmuka pengguna; dan *Controller* yang menjadi kontrol aplikasi.

Komponen tersebut secara lanjut dapat dibagi sebagai berikut:

1. View, merupakan bagian yang menangani logika antarmuka pengguna. Pada bagian ini biasanya berupa berkas templat HTML, yang diatur oleh Controller. View berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Bagian ini tidak memiliki akses langsung terhadap bagian Model.
2. Model, biasanya berhubungan langsung dengan pangkalan data untuk memanipulasi data (*insert*, *update*, *delete*, *search*), menangani validasi dari bagian Controller, tetapi tidak dapat berhubungan langsung dengan bagian View.
3. Controller, merupakan bagian yang mengatur hubungan antara bagian Model dan bagian View. Controller dapat dikatakan sebagai jembatan antara View dan Model. View akan menerima perintah dari pengguna dan perintah tersebut akan diolah oleh Controller, dan kemudian Controller akan mengambil data menggunakan Model, dan data dari Model tersebut akan ditampilkan di View oleh Controller. Secara mudah, Controller berfungsi untuk menerima permintaan dan data dari pengguna kemudian menentukan apa yang akan diproses oleh website.

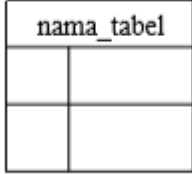
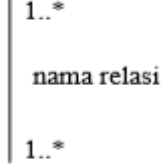
## 2.5 CDM

*Conceptual Data Model* atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang di simpan dalam basis data. CDM sudah di buat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data.

CDM merupakan hasil penjabaran lebih lanjut dari ERD. Ada aturan-aturan yang harus diikuti dalam melakukan konversi ERD menjadi CDM

Berikut adalah simbol-simbol yang ada pada CDM:

**Tabel 2.1. Simbol CDM**

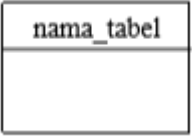
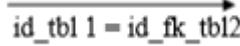
Simbol	Deskripsi
Entitas/Tabel 	Entitas atau tabel yang menyimpan data dalam basis data
Relasi 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .

## 2.6 PDM

Model Relasional atau *Physical Data Model* (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan konsep yang menerangkan detail dari bagaimana data di simpan di dalam basis data.

Berikut adalah simbol-simbol yang ada pada PDM:

**Tabel 2.2. Simbol PDM**

Simbol	Deskripsi
Tabel 	Tabel yang menyimpan data dalam basis data
Relasi 	Relasi antar tabel yang terdiri dari persamaan antara <i>primary key</i> (kunci primer) tabel yang di acu dengan kunci yang menjadi referensi acuan di tabel lain

## 2.7 Iconix Process

ICONIX process merupakan metode perancangan atau pengembangan yang mengacu pada use case. Use case ditentukan pada awal pengembangan yang mana menjadi dasar dalam penentuan model dan perilaku dari sistem yang sedang dibangun (Rosenberg & Stephens, 2007).

ICONIX *process* terdiri dari beberapa tahap. Setiap tahapnya menghasilkan keluaran yang digunakan dalam pengerjaan tahap berikutnya. Pendokumentasian pada setiap keluaran ditujukan untuk membantu proses pengembangan. Model statis yang dihasilkan terus diperbaiki secara bertahap dengan bantuan model dinamis yaitu *use case*, *robustness analysis* dan *sequence diagram*. ICONIX *process* memiliki 6 tahapan, yaitu: *Requirement*, *Analysis / Preliminary Design*, *Milestone 2: Preliminary Design Review*, *Detailed Design*, *Milestone 3: Critical Design Review*, dan *Implementation*. Berikut tabel gambaran tiap tahap dan keluarannya.

**Tabel 2.3. Artefak Metode ICONIX Proses**

Phase	Artifacts	UML
<i>Requirement</i>	<i>Functional Requirement</i>	No
	<i>Domain Modelling</i>	Yes
	<i>Behavioral Requirement</i>	Yes
	<i>Requirement Review</i>	No
<i>Analysis/ Preliminary Design</i>	<i>Robustness Analysis</i>	Yes
<i>Milestone 2: Preliminary Design Review</i>	<i>Domain Model</i>	Yes
<i>Detailed Design</i>	<i>Sequence Diagram</i>	Yes
	<i>Class Diagram</i>	Yes
<i>Milestone 3: Critical Design Review</i>	<i>Consistecy Analysis</i>	No
	<i>Correctness</i>	No

## 2.7.1 Fase Requirement

### a) *Functional Requirement*

Pada tahap ini dilakukan identifikasi pemangku kepentingan, identifikasi permasalahan, identifikasi aktor serta analisis dan spesifikasi kebutuhan system.





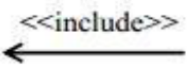
### b) *Domain Modeling*


Dilakukan identifikasi kata benda dan istilah (*noun dan noun-phrase*) yang terdapat pada tahap *functional requirement*. Domain model umumnya digambarkan memakai notasi UML class diagram yaitu notasi attribute, association, role dan multiplicity. Oleh karena itu dalam penyajiannya, domain model mampu menjelaskan informasi properti, aturan dan keterhubungan relasi antarkelas.

### c) *Behavioral Requirement*

Dilakukan pembangunan GUI *storyboard* atau rancangan tampilan awal serta *use case diagram* yang menggambarkan fitur dan fungsi system serta siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut Simbol-simbol yang digunakan dalam *Use Case Diagram*.

**Tabel 2.4. Simbol Use Case Diagram**

Simbol	Keterangan
	Aktor : mewakili peran orang, system yang lain, atau alat ketika berkomunikasi dengan use case
	<i>Use Case</i> : Abstraksi dan interaksi antara system dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara actor dengan use case
	Generalisasi : Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya

	<p>Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi.</p>
---	---

#### **d) Milestone 1: Requirement Review**

Pada tahap ini dilakukan peninjauan terhadap analisis sebelumnya. Hasil yang didapatkan adalah sedikit penambahan dan pembaharuan pada *functional requirement* dan pada *GUI storyboard*.

### **2.7.2 Analysis/Preliminary Design**

#### **a) Robustness Analysis**

Tahap ini dilakukan penelitian menggunakan Robustness diagram yang merupakan jembatan/penghubung antara analisis dan perancangan sistem. Diagram ini merupakan gambaran objek dari use case yang telah dibuat. *Robustness Diagram* ini diklasifikasikan kedalam : *boundary object*, *entity object* (objek entitas), dan *Controller* (aktivitas).

#### **2.7.1. Milestone 2: Preliminary Design Review**







Tahap ini dilakukan pembaruan terhadap Domain Model berdasarkan pada tahap sebelumnya apabila terdapat data yang dapat dimasukkan atau dihapus dari domain model.

### **2.7.2. Detailed Design**

#### **a) Sequence Diagram**

*Sequence diagram* dibuat sesuai *use case* dan *robustness diagram*. Penamaan untuk *actor*, *boundary*, *control*, dan *entity* yang dipakai merupakan istilah yang disesuaikan dengan *domain model* yang telah dibuat. Alur *sequence diagram* mengikuti *robustness diagram* yang sebelumnya mengikuti *use case scenario*. Berikut simbol-simbol yang digunakan dalam Sequence Diagram.

**Tabel 2.5. Simbol Sequence Diagram**

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
4		<i>Control Class</i>	Digunakan untuk menghubungkan boundary dengan tabel.
5		<i>Entity Clas</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

**b) Class Diagram**

*Class diagram* dibuat dari perkembangan domain model dan diagram-diagram sebelumnya.

**2.7.3. Milestone 3: Critical Design Review**

Pada tahap ini dilakukan evaluasi dengan tujuan memastikan perancangan sistem yang telah dibuat sesuai dengan analisis dan kebutuhan yang telah dijabarkan sebelumnya.

**2.7.4. Implementation**

**a) Coding/unit testing**

**b) Integration and scenario testing**

Melakukan integrase dalam penggunaan sistem berdasarkan basic course dan alternate course.

**c) Melakukan review kode dan pembaharuan model untuk dilanjutkan ke tahap pengembangan selanjutnya.**

Dalam membangun Jurnal Kerja dan Mailing system ini, dari penjelasan masing-masing diagram tersebut yang akan dibuat adalah Domain Model, Class Diagram, Use Case Diagram, Robustness Diagram dan Sequence Diagram.

**2.8 PostgreSQL(PGAdmin)**

Database PostgreSQL merupakan salah satu alternatif solusi bagi pengguna database yang mendukung banyak platform dan bebas lisensi. PostgreSQL termasuk sebagai database server yang handal dengan berbagai macam fitur-fitur pendukungnya, sehingga menjadikan database ini begitu ideal sebagai media

penyimpanan dari aplikasi sistem informasi. PostgreSQL dikembangkan oleh University of California di Berkeley Computer Science Department. Dengan sifatnya yang open source menjadikan pula database ini dapat dikembangkan sesuai dengan kebutuhan.