# **BAB II LANDASAN TEORI**

1. **Pengertian Sistem**

Sistem adalah suatu urutan – urutan yang tepat dari tahapan–tahapan instruksi yang menerangkan apa yang harus dikerjakan, siapa yang mengerjakan, kapan dikerjakan, dan bagaimana mengerjakannya. [1]

1. **Pengertian Informasi**

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. [2]

1. **Pengertian Sistem Informasi**

Definisi Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah, dan menyebarkan informasi dalam organisasi. [3]

1. **Basis Data (*Data Base*)**

Basis Data adalah himpunan kelompok data yang saling berhubungan dan tanpa pengulangan yang disimpan dalam media elektronik dan diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan dengan cepat mudah. [4]

1. **Analisis dan Perancangan Sistem**

Definisi Analisis dan Perancangan Sistem menurut para ahli :

1. Analisis Sistem adalah suatu studi dari sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau memperbaiki kekurangan dari sistem yang telah ada.
2. Perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru, jika sistem itu berbasis komputer, perancangan dapat dinyatakan spesifikasi peralatan yang digunakan.
3. ***Flowchart***

Suatu gambaran yang mempresentasikan suatu algoritma atau prosedur menyelesaikan suatu masalah. [5]. *Flowchart* terbagi dua, yaitu *Flowchart* sistem dan *Flowchart* program.

* + 1. **UML (*Unified Modelling Languange*)**

*Unified Modelling Languange* merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofi kemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standard dan bersifat independen. [6] Jenis-jenis diagram UML:

1. *Use case* Diagram

Diagram yang menggambarkan aktor, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML use case. *Use case* memiliki dua istilah

1. System use case; interaksi dengan sistem.

2. Business use case; interaksi bisnis dengan konsumen atau kejadian nyata

1. *Class diagram*

*Class diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)

2. Atribut

3. Metoda

1. *Sequence* Diagram

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

1. *State* *Chart* Diagram

Menggambarkan *state*, transisi *state* dan *event*.

1. *Activity* Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*

Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

1. *Component* Diagram

Component diagram menggambarkan struktur fisik dari kode, pemetaan

pandangan logis dari kelas proyek untuk kode aktual di mana logika ini

dilaksanakan.

1. *Object* Diagram

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-

kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non - program yang mungkin menemukan diagram kelas terlalu abstrak.

1. *Deployment* Diagram

*Deployment* diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment*

Diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

1. *Collaboration* Diagram

Diagram ini merupakan bentuk lain dari *sequence* diagram. Diagram ini menggambarkan struktur organisasi dari sistem dengan pesan yang diterima dan dikirim.

1. **Perancangan Basis Data**

Perancangan basis data adalah sebuah pendekatan struktur yang mencakup prosedur, teknik, alat bantu perancangan untuk merencanakan, mengatur dan mengevaluasi pengembangan dari proyek pembuatan basis data tersebut [7]. Menurut Conolly metodologi perancangan basis data terbagi menjadi 2 bagian yaitu:

1. Perancangan Fisik

Merupakan proses pendefinisian entitas dan relasi (*Relationship*) dari dunia nyata yang dirancang, berdasarkan kebutuhan informasi dan pengolahan data dari organisasi yang bersangkutan.

1. *Entity* / entitas adalah sekumpulan objek yang dapat diidentifikasi dan dibedakan di lingkungan pemakai
2. Relasi adalah hubungan yang terjadi antar kelompok entitas.
3. Perancangan Logik

Merupakan proses untuk mengimplementasikan hasil perancangan logika ke dalam komputer secara fisik yang bergantung kepada *software* DBMS yang dipilih. Proses yang dilakukan :

1. Menentukan struktur untuk setiap tabel, meliputi nama *field*, jenis, lebar dan field kuncinya.
2. Menentukan nama basis data dan nama setiap tabel, serta lokasi tempat penyimpanannya (*drive*, *directory* / *folder*).
3. Menghitung perkiraan tempat (space) yg dibutuhkan untuk seluruh tabel dan untuk seluruh *index*.
4. Implementasi dengan menggunakan *software* DBMS.
   1. **DBMS (*Data Base Management* *System*)**

Definisi DBMS adalah sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke *database* [8] . Berikut ini ada beberapa contoh DBMS (*Data* *Base* *Management* *System*) diantaranya ialah :

* + 1. XAMPP

XAMPP adalah program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

1. MySQL

MySQL merupakan aplikasi *database* *server*. Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured* *Query* *Language*. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. MySQL dapat digunakan untuk membuat dan mengelola *database* beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database*.

* 1. **Bahasa Pemrograman**

1. Bahasa pemograman adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Berikut ini adalah beberapa contoh bahasa pemrograman yang biasa digunakan :
2. HTML (Hyper Text Markup Language)

Pengertian HTML : “HTML adalah dokumen text yang bisa dibaca untuk dipublikasikan di *World* *Wide* *Web* (WWW), dan semua nama file dokumen HTML mempunyai ekstensi html atau htm”. [9] Script dasar pada HTML :

<HTML>

<HEAD>

Deskripsi dokumen

</HEAD>

<BODY>

Isi dokumen

</BODY>

</HTML>

1. PHP (Hyper Text Prepocessor)

PHP singkatan dari *Hypertext Processor* yang merupakan *server-side programming*,yaitu bahasa pemrograman yang diproses di sisi server. Fungsi utama PHP dalam membangun website adalah untuk melakukan pengolahan data pada *database*. Data website akan dimasukkan ke *database*, diedit, dihapus, dan ditampilkan pada website yang diatur oleh PHP .[10]

* Script dasar pada PHP :

<html>

<head>

<title>Judul</title>

</head>

<body>

<?php

echo("Isi dokumen php");

?>

</body>

</html>

1. **Yii Framework**

Yii adalah sebuah *source* *framework* menggunakan Bahasa pemrograman PHP yang dibuat oleh Qiang Xue tanggal 1 Januari 2008. Sebelumnya Qiang juga mengembangkan framework PRADO selama bertahun tahun.Yii secara resmi dirilis untuk memenuhi kebutuhan para pengembang perangkat lunak berbasis web. Beberapa pengembang perangkat lunak merespon positif atas perkembangan *framework* PHP.[11]

1. **MVC (*Model*-*View*-*Contoller*)**

Yii mengimplementasikan pola desain *model-view-controller* (MVC), yang diadopsi secara luas dalam pemrograman Web. MVC bertujuan untuk memisahkan logika bisnis dari pertimbangan antar muka pengguna agar para pengembang bisa lebih mudah mengubah setiap bagian tanpa mempengaruhi yang lain. Dalam MVC, model menggambarkan informasi (data) dan aturan bisnis; *view* (tampilan) berisi elemen antar muka pengguna seperti teks, *input* *form*; sementara *controller* mengatur komunikasi antar *model* dan *view*.[12]

Selain implementasi MVC, Yii juga memperkenalkan *front*-*controller* (controller-depan), yang disebut Aplikasi, yang mengenkapsulasi konteks eksekusi untuk memproses sebuah *request*. Aplikasi mengumpulkan beberapa informasi mengenai *request* pengguna dan kemudian mengirimnya ke *controller* yang sesuai untuk penanganan selanjutnya. Diagram berikut memperlihatkan struktur statis sebuah aplikasi Yii:

**Struktur Statis Aplikasi Yii**

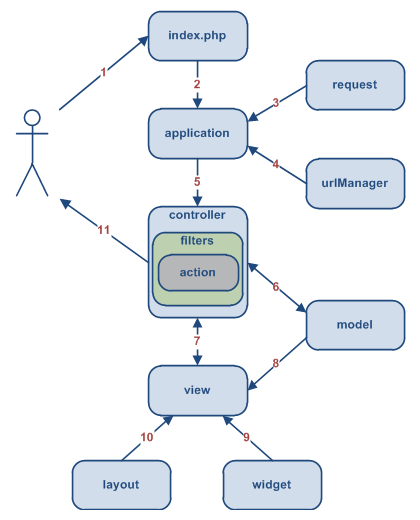


## Gambar 2.1 Struktur Statis Aplikasi Yii

## **Alur kerja Umum**

Diagram berikut memperlihatkan alur kerja umum sebuah aplikasi Yii saat menangani permintaan pengguna:

Alur Kerja Umum Aplikasi Yii



Gambar 2.2 Alur Kerja Umum Aplikasi Yii

1. Pengguna membuat permintaan dengan URL http://www.example.com/index.php?r=post/show&id=1dan server Web menangani permintaan dengan menjalankan skrip bootstrap index.php.Skrip bootstrap membuat sebuah *instance* [Aplikasi](http://www.yiiframework.com/doc/guide/1.1/id/basics.application) dan menjalankannya.
2. Aplikasi mendapatkan rincian informasi permintaan pengguna dari [komponen aplikasi](http://www.yiiframework.com/doc/guide/1.1/id/basics.application#application-component) bernama request.
3. Aplikasi menentukan controller dan aksi yang diminta dengan bantuan komponen aplikasi bernama urlManager. Dalam contoh ini, controller adalah post yang merujuk pada kelas PostController; dan aksi adalah show yang arti sebenarnya ditentukan oleh controller.
4. Aplikasi membuat *instance* *controller* yang diminta untuk selanjutnya menangani permintaan pengguna. *Controller* menentukan aksi *show* merujuk pada sebuah metode bernama *actionShow* dalam kelas *controller*. Kemudian membuat dan menjalankan *filter* (contoh kontrol akses, pengukuran) terkait dengan aksi ini. Aksi dijalankan jika diijinkan oleh *filter*.
5. Aksi membaca *Post* [*model*](http://www.yiiframework.com/doc/guide/1.1/id/basics.model) di mana ID adalah 1 dari *database*.
6. Aksi meyiapkan [*view* (tampilan)](http://www.yiiframework.com/doc/guide/1.1/id/basics.view) bernama *show* dengan model *Post*.
7. View membaca dan menampilkan atribut model *Post*.
8. View menjalankan beberapa [*widget*](http://www.yiiframework.com/doc/guide/1.1/id/basics.view#widget).
9. View menyiapkan hasil yang dipasangkan dalam [*layout*(tata letak)](http://www.yiiframework.com/doc/guide/1.1/id/basics.view#layout).
10. Aksi mengakhiri pembuatan *view* dan menampilkan hasil akhir kepada pengguna.