

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian yang dilakukan oleh Gisma Gerry Kurniawan (2018) dengan judul “Implementasi *Framework Yii* untuk Sistem Penjualan Online di Toko Passion Yogyakarta”. Aplikasi ini di buat menggunakan bahasa pemrograman PHP dan MySQL. Sistem ini membantu dalam sistem penjualan online.

Penelitian yang dilakukan oleh Wisnu Nugroho (2017) dengan judul “Pemanfaatan *Cloud Back-End Service* Untuk Aplikasi *Spot Photo Hunting* Berbasis *Android*”. Aplikasi ini di buat menggunakan sistem android yang membantu petugas dalam pemanfaatan aplikasi *spot foto hunting*.

Penelitian lainnya dilakukan oleh Deviana Astika Sari (2017) dengan judul “Aplikasi Pemesanan Barang Dengan Menggunakan *Framework Yii* (Studi Kasus Usaha Sub Agen Jaya Mandiri Kota Samarinda)”. Aplikasi ini di buat menggunakan bahasa pemrograman PHP dan MySQL. Sistem ini membantu petugas dalam pengelolaan data pemesanan yang masih dilakukan secara manual.

Penelitian lainnya dilakukan oleh Eko Yuswanto (2018) dengan judul “Implementasi *Framework Codeigniter* Untuk Pengelolaan Nilai di SMK Ash – Shiddiqiyah Balingasal”. Aplikasi ini dibuat menggunakan bahasa PHP dan MySQL Sistem ini membantu dalam pengelolaan data nilai yang selama ini masih dilakukan secara konvensional.

Tabel 2.1 Tinjauan Pustaka

No.	Peneliti	Tahun	Judul	Keterangan
1.	Kurniawan, Gisma Gerry	(STMIK AKAKOM : 2018)	Implementasi Teknologi <i>Framework Yii</i> untuk Sistem Penjualan Online di Toko Passion Yogyakarta	Membahas tentang pembuatan aplikasi penjualan berbasis web secara online dengan menggunakan <i>framework yii</i>
2.	Nugroho, Wisnu	(STMIK AKAKOM : 2017)	Pemanfaatan <i>Cloud Back- End Service</i> Untuk Aplikasi Spot Photo <i>Hunting</i> Berbasis <i>Android</i>	Membahas tentang pembuatan sebuah aplikasi <i>front-end</i> spot foto <i>hunting</i> yang berjalan di <i>platform android</i> dan dapat berkomunikasi dengan <i>back-end</i> yang berjalan secara <i>cloud</i> .
3.	Sari, Deviana Astika	(STMIK AKAKOM : 2017)	Aplikasi Pemesanan Barang dengan Menggunakan <i>Framework Yii</i> (Studi Kasus Usaha Sub Agen Jaya Mandiri)	Membahas tentang pembuatan aplikasi berbasis <i>web</i> yang mengatur data transaksi penjualan dan pemesanan barang secara elektronik atau <i>online</i> berdasarkan pada sistem penjualan yang telah berlaku pada objek yang dipilih dengan menggunakan <i>framework Yii</i> .
4.	Yuswanto, Eko	(STMIK AKAKOM : 2018)	Implementasi <i>Framework Codeigniter</i> Untuk Pengelolaan Nilai di SMK Ash – Shiddiqiyah Balingasal	Membahas tentang pembuatan aplikasi <i>web</i> untuk pengelolaan data nilai pada SMK ASH – SHIDDIQIYAH Balingasal secara online berdasarkan pada sistem pengelolaan yang telah berlaku dengan menggunakan <i>framework codeigniter</i> .
5.	Usulan, Istiqomah	(STMIK AKAKOM : 2019)	Implementasi <i>Framework Yii</i> 2 Pada <i>Back-end</i> Berbasis <i>Web Service</i>	Membahas tentang pembuatan aplikasi <i>web</i> untuk pengelolaan data transaksi pemasukan dan pengeluaran secara online berdasarkan pada sistem yang berlaku pada objek yang dipilih dengan menggunakan <i>framework yii</i> .

Berdasarkan penelitian pada tabel 2.1 dapat di ketahui bahwa terdapat perbedaan antara penelitian sebelumnya, baik berupa perbedaan metode dan bahasa pengrograman dalam merancang aplikasi. Dalam hal ini peneliti melakukan perancangan aplikasi *web* dengan berbasis *web service* menggunakan PHP dan MySQL. Sedangkan untuk pengambilan data menggunakan JSON.

## **2.2 Dasar Teori**

### **2.2.1 CV. Central Laundry**

CV. Central Laundry dibentuk dan didirikan pertama kali pada tahun 2001 , di JL. Laksada Adisucipto Yogyakarta (Sekarang Kawasan Ambarukmo Plaza Yogyakarta) dengan tujuan sebagai jasa melayani kebutuhan masyarakat dalam hal sandang. Dalam era modern ini, kebutuhan masyarakat berupa sandang, pangan dan papan mulai berkembang pesat. Sejalan dengan modernisasi yang memerlukan komitmen penuh, khususnya perhatian dalam hal perawatan sandang. Tidak hanya menyediakan infastruktur yang memadai, modern, dan *advance* tetapi juga sistem dan kinerja yang *qualitified*.

CV. Central Laundry, semakin mendapat tempat dan dipercaya di kalangan pengguna jasa pencucian baik perorangan maupun kelembagaan, hotel, restoran, dan rumah sakit. Seiring dengan perjalananya dan demi memberikan pelayanan yang lebih baik dan modern CV. Central Laundry mengembangkan diri dengan melakukan Penambahan baik dalam segi *infrastruktur*, maupun bangunan. Peningkatan kualitas

juga terus dilakukan dengan menempatkan sumberdaya manusia yang terlatih, profesional & kompeten.

CV Central Laundry memiliki pelayanan yang total seperti pelayanan selama 24 jam, Pengerjaan yang lebih cepat (Express) selama 25 menit, melayani secara *Pickup Dilevery Free* atau pelayanan antar dan jemput dengan sejumlah armada yang selalu siap untuk diberangkatkan. ([www.centrallaundry.co.id](http://www.centrallaundry.co.id))

### **2.2.2 Back-End**

*Back-end* adalah tempat dimana proses suatu aplikasi atau sistem berjalan dan berfungsi sebagai penyuplai atau sumber data aplikasi. *Back-end* mengurus segala sesuatu yang biasanya tidak dilihat atau berinteraksi langsung kepada *user*, seperti *database* dan *server*. Jenis pengembangan *web* ini biasanya terdiri dari tiga bagian: *server*, aplikasi, dan *database*. Kode yang ditulis oleh pengembang *back-end* inilah yang mengkomunikasikan informasi *database* ke browser. *Back-end* lebih fokus pada responsivitas dan kecepatan situs. Posisi pengembang *back-end* sering disebut programmer atau pengembang *web*. (Suci Ana Lathifah, 2014)

### **2.2.3 Database**

Basis data atau *database* merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis data dalam menyediakan

informasi bagi para pemakai. Jenjang data yang diperlukan dalam membentuk basis data adalah :

1. *Characters* merupakan bagian data terkecil, yang dapat berupa karakter *numeric*, huruf ataupun karakter-karakter khusus (*special characters*) yang membentuk suatu item data *field*.
2. *Field* mempresentasikan suatu atribut dari *record* yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* yang membentuk suatu *record*.
3. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari *record* membentuk suatu file.
4. File terdiri dari *record-record* yang menggambar suatu kesatuan data yang sejenis.
5. *Database* kumpulan dari file/tabel membentuk suatu *database*. (Fathansyah, 1999)

#### 2.2.4 Yii

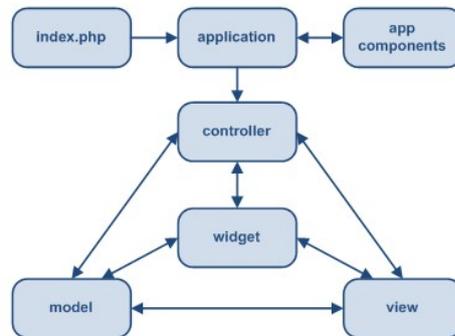
Yii adalah *framework* (bingkai atau kerangka kerja) PHP berbasis komponen yang berkinerja tinggi dan digunakan untuk mengembangkan aplikasi *web modern* secara cepat. Yii merupakan singkatan dari *Yes, It Is!*, yang awalnya dibuat oleh seorang pengembang berkebangsaan Cina bernama Qiang Xue sejak tahun 2008. Yii merupakan *framework web* generik yang dapat digunakan untuk mengembangkan berbagai macam jenis aplikasi *web*, seperti portal, *forum*, *Content Management System*

(CMS), proyek-proyek *e-commerce* dan sebagainya. Sama seperti sebagian besar *framework* PHP lainnya, Yii juga mengimplementasikan pola desain *Model-View-Controller* (MVC).

Dalam teknik pemrograman berorientasi objek, MVC adalah nama dari suatu metodologi atau pola desain (*design pattern*) yang digunakan untuk merelasikan data dan *user-interface* aplikasi secara efisien. Hampir semua *framework* untuk pengembangan aplikasi *web* mengimplementasikan arsitektur MVC. Dalam pola MVC, komponen aplikasi dibagi menjadi tiga bagian, yaitu :

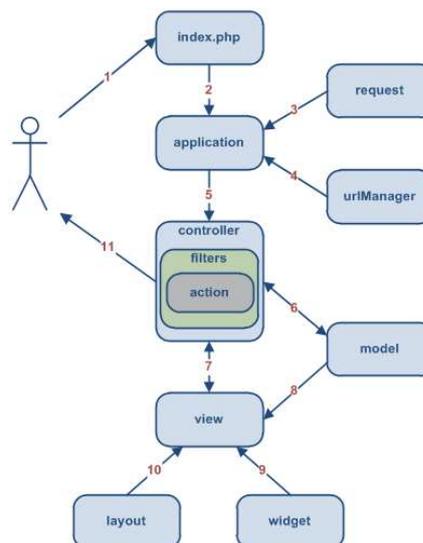
1. *Model*, yaitu merepresentasikan struktur data.
2. *View*, yang merupakan representasi keluaran (*output*) dari suatu model.
3. *Controller*, yaitu komponen yang bertugas mengambil masukan (*input*) dari *user* dan mengubahnya menjadi perintah untuk *model* dan atau *view*.

Dalam mengimplementasikan pola desain MVC, Yii memperkenalkan bagian *controller* depan (*front-controller*) yang disebut *application*. Elemen ini bertugas untuk memproses permintaan (*request*) yang dikirim oleh *user* dan mengirimnya ke *controller* yang sesuai, untuk proses penanganan lebih lanjut. Dalam beberapa *framework web* lain, elemen ini sering juga disebut sebagai *router*. Berikut ini gambar 2.1 yang menunjukkan arsitektur aplikasi di dalam Yii.



Gambar 2.1 Struktur Statis Aplikasi Yii

Pada gambar 2.1, *app component* berperan sebagai elemen pendukung yang akan membantu elemen *application* dalam menjalankan tugasnya. Dalam elemen *controller*, terdapat elemen lain yang disebut *action* dan *filter*. Berikut ini gambar 2.2 yang menunjukkan gambar diagram alur kerja umum dari sebuah aplikasi *Yii* ketika menangani permintaan pengguna.



Gambar 2.2 Alur Kerja Umum Aplikasi Yii

Keterangan dari gambar 2.2 diatas adalah sebagai berikut :

1. *User* mengirim permintaan dengan format *Uniform Resource Locator* (URL) seperti berikut : `http://namahost/index.php?r=namacontroller/namaaksi`  
*Server web* akan menangani permintaan ini dengan menjalankan skrip `index.php` (dalam Yii, file ini disebut *entry script* atau *bootstrap script*).
2. Skrip `index.php` akan membuat *instance* aplikasi (*application*) dan menjalankannya.
3. Aplikasi mendapatkan rincian informasi permintaan yang dikirim oleh user dari komponen aplikasi bernama *request*.
4. Aplikasi menentukan *controller* dan aksi yang diminta dengan bantuan komponen aplikasi bernama *urlManager*.
5. Aplikasi membuat *instance* dari *controller* yang diminta untuk menangani permintaan *user*. *Controller* akan menentukan aksi (*action*), kemudian membuat dan menjalankan *filter*. Suatu aksi akan dijalankan jika diizinkan oleh *filter*.
6. Aksi membaca model untuk mendapatkan data yang akan ditampilkan (misalnya: data yang berasal dari *database*).
7. Aksi menyiapkan *view* (tampilan).
8. *View* akan membaca dan menampilkan atribut yang didefinisikan di dalam model.
9. *View* memanggil daftar *widget* yang digunakan.

10. *View* menyiapkan hasil dan menggabungkannya dengan *layout* (tata letak) atau *template* yang ada.
11. Aksi mengakhiri pembuatan *view* dan menampilkan hasil akhir ke *user*. (Budi Raharjo, 2015)

### 2.2.5 MySQL

MySQL adalah *database server relational* yang gratis dibawah *lisensi General Public License* (GNU). Dengan sifatnya yang *open source*, memungkinkan juga user untuk melakukan modifikasi pada *source code* untuk memenuhi kebutuhan spesifik mereka sendiri. MySQL merupakan *database server multi-user* dan *multitrade* yang tangguh (*robust*). Dengan memiliki banyak fitur MySQL bisa bersaing dengan *database komersil* sekalipun. Tidak mengejutkan, MySQL bisa menjadi *database pilihan* untuk banyak pengguna PHP. MySQL dikembangkan oleh 14 MySQL AB, sebuah perusahaan *komersil* yang membangun layanan bisnisnya melalui *database MySQL*. (Ferrar Utdirartatmo, 2001)

### 2.2.6 RESTful Web Service

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai *protocol* untuk komunikasi data. Pada arsitektur REST, REST *server* menyediakan *resources* (sumber daya/data) dan REST client mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs

(*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

**Keuntungan REST :**

1. Bahasa dan *platform agnostic*.
2. Lebih sederhana atau simple untuk dikembangkan ketimbang SOAP.
3. Mudah dipelajari, tidak tergantung pada *tools*.
4. Ringkas, tidak membutuhkan layer pertukaran pesan (*Messaging*) tambahan.
5. Secara desain dan filosofi lebih dekat dengan web

**Kelemahan REST :**

1. Mengasumsi model *point-to-point* komunikasi tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara
2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri")
3. Berkaitan dengan model transport HTTP.

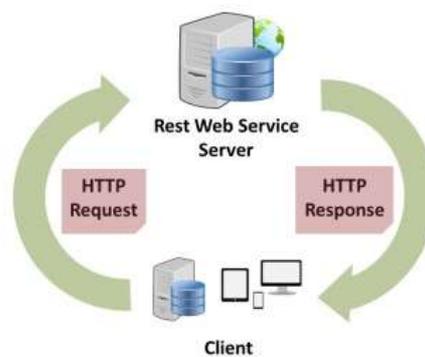
Berikut metode HTTP yang umum digunakan dalam arsitektur berbasis REST.

- GET, menyediakan hanya akses baca pada *resource*
- PUT, digunakan untuk menciptakan *resource* baru
- DELETE, digunakan untuk menghapus *resource*

- POST, digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru
- OPTIONS, digunakan untuk mendapatkan operasi yang disupport pada *resource*

*Web service* adalah aplikasi yang dibuat agar dapat dipanggil atau diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan. Format pertukaran data dapat dilakukan dengan dengan XML dan JSON. Sebuah *Web Service* dapat dipanggil oleh aplikasi dengan menggunakan bantuan HTTP (*HyperText Transfer Protocol*). *Web service* yang berbasis arsitektur REST kemudian dikenal sebagai RESTful *web services*. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

### Cara Kerja RESTful Web Service



Gambar 2.3 Struktur *web service*

Sebuah *client* mengirimkan sebuah data atau *request* melalui HTTP *Request* dan kemudian *server* merespon melalui HTTP *Response*. Komponen dari http request :

- *Verb*, HTTP *method* yang digunakan misalnya GET, POST, DELETE, PUT dll.
- *Uniform Resource Identifier* (URI) untuk mengidentifikasi lokasi *resource* pada *server*.
- HTTP *Version*, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
- *Request Header*, berisi metadata untuk HTTP *Request*. Contoh, *type client/browser*, format yang didukung oleh *client*, format dari *body* pesan, *setting cache* dll.
- *Request Body*, konten dari data.

Sedangkan komponen dari *http response* :

- *Status/Response Code*, mengindikasikan status *server* terhadap *resource* yang *di request*. misal : 404, artinya *resource* tidak ditemukan dan 200 *response* OK.
- HTTP *Version*, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
- *Response Header*, berisi metadata untuk HTTP *Response*. Contoh, *type server*, panjang *content*, tipe *content*, waktu *response*, dll.
- *Response Body*, konten dari data yang diberikan. (hafid mukhlisin,2015)