

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang sistem informasi bank sampah sebelumnya telah dilakukan oleh Izzudin^[2] dengan judul “Perancangan dan Pembuatan Sistem Informasi Bank Sampah Berbasis Web”. Sistem ini digunakan untuk memudahkan pengelola dalam management pengelolaan sampah seperti halnya memudahkan untuk menentukan member yang mencapai batas minimal pembayaran dan memudahkan dalam pengambilan keputusan penentuan member terbaik dan melakukan pencetakan laporan. Sistem ini menggunakan metode CPI (*Comparative Performance Index*) untuk membantu dalam mendukung keputusan untuk menentukan member terbaik.

Penelitian lainnya dilakukan oleh Firmansyah^[3] dengan judul “Rancangan Sistem Informasi Bank Sampah (SIBAS) Berbasis Desktop dengan Metode *Waterfall* “. Sistem ini dibangun dengan menggunakan metode dengan model *waterfall*. Sistem bank sampah dikembangkan menggunakan java berbasis desktop, MySQL untuk pembuatan dan pengolahan database. Sistem ini dapat berfungsi mengelola hasil data sampah dengan menampung, memilah, dan mendistribusikan sampah ke fasilitas pengelolaan sampah atau pihak yang membutuhkan. Sistem memberikan keuntungan kepada semua pihak, masyarakat menerima manfaat ekonomis dari hasil penjualan sampah, pengepul mendapatkan manfaat efisien karena sampah terkumpul ditempat dan sudah terpilah.

Penelitian serupa pada sistem bank sampah pernah dibagikan melalui internet, yang diupload melalui website *elib.unikom.ac.id* dengan judul “Sistem Informasi Bank sampah Pada Bank Sampah Binangkit Sukagalih”. Sistem ini dibuat untuk memberikan kemudahan dalam pengolahan data buku tabungan nasabah, pengolahan dan pencarian simpanan dapat di proses dengan cepat. Dalam proses penyimpanan data buku tabungan sistem ini menggunakan database. Sistem ini dapat membantu dan mempermudah dalam mengelola buku besar, pembuatan laporan tabungan, tanda bukti kwitansi. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode SDLC model waterfall, untuk pengujian sistem penulis menggunakan menggunakan metode Black box testing^[4].

Sementara penelitian lain terkait bank sampah telah banyak dilakukan. Salah satunya penelitian yang dilakukan oleh Aziz^[5] dengan judul “Rancangan Fitur Aplikasi Pengelolaan Administrasi dan Bisnis Bank Sampah di Indonesia”. Sistem ini akan menghasilkan rancangan fitur aplikasi berupa rancangan fungsionalitas fitur dan tampilan *user interface* aplikasi sebagai acuan untuk mengembangkan sistem aplikasi pengelolaan administrasi dan bisnis bank sampah. Sistem ini dibuat untuk menyelesaikan permasalahan pada proses data administrasi dan kendala akses pembaharuan informasi oleh nasabah bank. Metode penelitian yang digunakan menggunakan *Object Oriented Method* sebagai pendekatan untuk menghasilkan rancangan fitur aplikasi pengelolaan administrasi dan bisnis bank sampah di Indonesia dengan menggunakan penerapan *Unified Modeling Language* (UML).

Pada penelitian ini penulis bermaksud membangun “Sistem Informasi Bank Sampah di Bank Sampah Envuro’18”. Sistem ini diharapkan dapat membantu pihak bank sampah dalam memberikan informasi kepada nasabah tentang harga dan jenis sampah yang dapat di jual ke pihak bank sampah. Sistem ini juga membantu calon nasabah dalam melakukan proses pendaftaran dengan mudah karena pendafrtan dilakukan secara online.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau melakukan sasaran yang tertentu. Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan^[6].

Sistem informasi adalah suatu kombinasi teratur dari orang-orang, hardware, software, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi yang memiliki fungsi meningkatkan aksesibilitas data yang ada secara efektif dan efisien kepada pengguna tanpa perantara. Sistem informasi merupakan suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan untuk menjalankan operasional perusahaan, dimana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi^[7].

2.2.2 Sampah

Menurut Undang-Undang Nomor 18 Tahun 2008 tentang Pengelolaan Sampah, sampah adalah sisa kegiatan sehari-hari manusia dan atau proses alam yang berbentuk padat. Kemudian penghasil sampah adalah setiap orang dan atau akibat proses alam yang menghasilkan timbulan sampah. Tempat penampungan sampah adalah tempat sebelum sampah diangkut ke tempat pendauran ulang, pengelolaan, dan atau tempat pengelolaan sampah terpadu. Tempat pengelolaan sampah terpadu adalah tempat dilaksanakannya kegiatan pengumpulan, pemilahan, penggunaan ulang, pendauran ulang, pengelolaan, dan pemrosesan akhir sampah.

Bank Sampah adalah pengumpulan sampah kering yang sudah dipilah dan dibersihkan serta memiliki manajemen layaknya perbankan tapi yang ditabung bukan uang melainkan sampah. Warga yang menabung disebut nasabah, memiliki buku tabungan dan dapat meminjam uang yang nantinya dikembalikan dengan sampah seharga uang yang dipinjam. Sampah yang ditabung ditimbang dan dihargai dengan sejumlah uang nantinya akan dijual ke pengepul sampah yang sudah bekerja sama dengan pihak bank sampah^[8].

2.2.3 Rekayasa Web

Rekayasa Web adalah sebuah aplikasi yang menggunakan pendekatan sistematis, disiplin, dan terukur untuk pengembangan, operasi dan pemeliharaan aplikasi berbasis Web (*Web-based application*). Rekayasa Web adalah subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara, dan melibatkan aplikasi Web^[9].

A. PHP

PHP adalah sebuah bahasa scripting tingkat tinggi yang dipasang pada dokumen HTML. Sebagian sintaks dalam PHP mirip dengan bahasa C, Java dan Perl, namun pada PHP terdapat beberapa fungsi yang lebih spesifik. Tujuan utama dari penggunaan bahasa ini untuk memungkinkan perancang web yang dinamis dan dapat bekerja secara otomatis.

PHP adalah situs web yang bisa menyesuaikan tampilan tergantung ^{dengan} situasi. Website dinamis bisa menyimpan data ke dalam database, membuat halaman yang berubah-ubah sesuai input dari user, memproses form, dan lainnya. Untuk pembuatan web, kode PHP biasanya disisipkan ke dalam dokumen HTML. Karena fitur inilah PHP disebut sebagai Scripting Language atau Bahasa Pemrograman Script^[10].

B. Web Server

Web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui browser kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML. Tugas web server adalah mengatur semua komunikasi yang terjadi antara browser dengan server untuk memproses sebuah website. Terdapat beberapa web server yang paling banyak digunakan untuk melakukan pengembangan aplikasi web yaitu Apache, NGINX, IIS, dan LiteSpeed Web Server^[11].

C. Web Browser

Web browser adalah aplikasi yang bisa menjelajahi, menyajikan, maupun mengambil konten yang ada di berbagai sumber informasi pada jaringan internet atau www. Web browser juga sering disebut perangkat lunak dengan fungsi yang dimilikinya sebagai penerima, pengakses, penyaji berbagai informasi di internet. Sumber informasi yang dimaksud terdiri dari berbagai macam bentuk seperti misalnya berupa halaman situs, gambar, video, infografis, dan yang lainnya. Tujuan utama dari web browser adalah untuk membawa sumber informasi kepada pengguna. Ada berbagai macam perangkat web browser seperti Mozilla Firefox, Google Chrome, Opera, Safari, internet Explorer dan lain sebagainya ^[12].

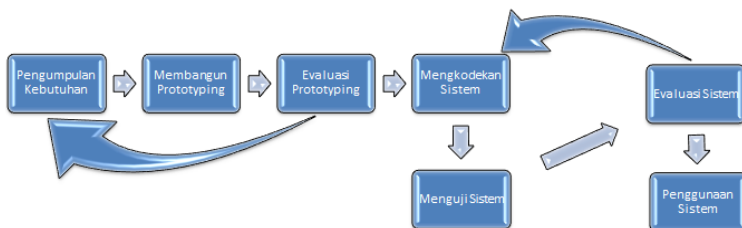
2.2.4 Rekayasa perangkat lunak

Rekayasa Perangkat Lunak adalah sebuah disiplin ilmu yang mencakup segala hal yang berhubungan dengan proses pengembangan perangkat lunak sejak dari tahap perancangan hingga tahapan implementasi serta pasca implementasi sehingga siklus hidup perangkat lunak dapat berlangsung secara efisien dan terukur. Perangkat lunak sering dibuat dan pada akhirnya tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti keengganan pemakai perangkat lunak (user) untuk mengubah cara kerja dari manual ke otomatisasi, atau ketidakmampuan user menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai.

A. Metode Pengembangan Perangkat Lunak

SDLC atau *System development Life Cycle* adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Adapun model SDLC yang digunakan untuk penelitian ini yaitu model *prototype*.

Model SDLC *prototype* salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*)^[1]. Berikut langkah-langkah pada model *prototype*.



Gambar 2. 1 Prototype Model

1. Analisis Kebutuhan Perangkat Lunak

Pada tahap analisis kebutuhan perangkat lunak pengembang melakukan identifikasi *software* dan semua kebutuhan sistem yang akan dibuat.

2. Membangun *prototyping*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian pelanggan.

3. Evaluasi *prototyping*
Evaluasi ini dilakukan untuk mengetahui apakah *prototyping* sudah sesuai dengan harapan pelannga.
4. Pengkodean sistem
Pada tahap ini *prototyping* yang sudah disetujui akan diubah ke dalam bahasa pemrograman
5. Menguji sistem
Tahap ini dilakukan untuk menguji sistem perangkat lunak yang sudah dibuat.
6. Evaluasi sistem
Perangkat lunak yang sudah siap jadi akan dievaluasi oleh pelanggan untuk mengetahui apakah sistem sesuai dengan yang diharapkan.
7. Menggunakan sistem
Perangkat lunak yang sudah diuji dan disetujui oleh pelanggan siap digunakan.

B. Tools/Alat Bantu

1. UML (Unified Modeling Language)

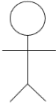





UML (*Unified Modeling Language*) adalah bahasa yang menjadi standar pada visualisasi, perancangan dan pendokumentasian sistem software^[13].

a. Use Case Diagram

Use case diagram atau diagram use case merupakan salah satu jenis diagram pada UML yang menggambarkan interaksi

antara sistem dan actor, use case diagram dapat mendeskripsikan tipe interaksi antara pengguna dengan sistem. Berikut tabel 2.1 adalah simbol-simbol yang ada pada diagram usecase^[13].


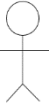



Tabel 2. 1 Simbol Usecase


No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Use case</i>	Deskripsi dari urutan aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

b. Sequence Diagram

Sequence diagram merupakan salah satu jenis diagram UML yang menjelaskan interaksi objek berdasarkan urutan waktu, sequence diagram dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk menghasilkan sesuatu seperti *use case diagram*. Berikut Tabel 2.2 adalah symbol-simbol yang ada pada diagram *sequence diagram*^[13].

Tabel 2. 2 Simbol Secequence


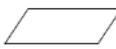
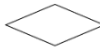

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna,
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan Tabel.




6.		<i>Entity</i> <i>Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
----	---	-------------------------------	---

c. Flowchart

Flowchart merupakan bagan-bagan yang mempunyai arus yang menggambarkan langkah penyelesaian suatu masalah. Flowchart salah cara penyajian dari suatu algoritma. Berikut Tabel 2.3 adalah symbol-simbol yang ada pada *Flowchart*^[14].

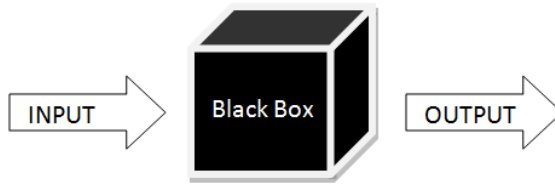
Tabel 2. 3 Simbol Flowchart

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program.
2.		<i>Input –</i> <i>Output</i>	Memasukan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
4.		<i>Flow</i>	Menghubungkan antara symbol satu dengan symbol yang lain atau menyatakan jalannya arus dalam suatu proses. symbol arus ini sering

			disebut dengan <i>connecting line</i> .
5.		<i>Document</i>	Merupakan symbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line keyboard</i> .

C. Metode Pengujian Sistem

Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Salah satu jenis pengujian sistem yaitu Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji *software* tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini, *tester* menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya. Metode pengujian sistem ini memiliki beberapa kelebihan yaitu efisiensi untuk segmen kode besar dan akses kode tidak diperlukan. Akan tetapi metode pengujian sistem ini memiliki kelemahan yaitu cakupan terbatas karena hanya sebagian kecil dari scenario pengujian yang dilakukan^[15].



Gambar 2. 2 Black Box Testing

2.2.5 Pemrograman berorientasi objek

Dasar pertimbangan untuk mengembangkan kemampuan *object oriented* dari PHP (*Personal Home Page Tools*) adalah perkembangan aplikasi *web* sebagai sebuah *platform* yang terus meluas dengan cepat karena ruang lingkup aplikasi *web* semakin luas sehingga aplikasi *web* yang dibangun juga semakin besar, rumit, dan kompleks.

Pemrograman berorientasi objek (*object oriented programming*, OPP) merupakan suatu pendekatan pemrograman yang menggunakan *object* dan *class*. Konsep OPP semakin berkembang. OPP bukan hanya sekedar penulisan sintaks dalam menganalisa sistem dan permasalahan pemrograman. Dalam OPP, setiap bagian dari program adalah *object*.

Keuntungan yang diperoleh dari pendekatan *object oriented* adalah kemudahan untuk mengelola kompleksitas aplikasi yang dibuat dan kemudahan untuk melakukan perubahan dan pengembangan dari aplikasi tersebut. Berikut ini beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek^[16].

1. Kelas (*class*)

Kelas merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method.

2. Objek (*object*)

Objek merupakan bentuk representasi dari sebuah kelas, membungkus data dan fungsi bersama menjadi suatu unit atau entitas dalam sebuah program.

3. Metode (*method*)

Metode adalah fungsi atau prosedur yang dibuat oleh pemrogram didalam suatu class, metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman procedural. Pada sebuah method di dalam sebuah kelas juga memiliki izin akses seperti atribut pada kelas.

4. Atribut (*attribute*)

Atribut adalah variabel yang dimiliki sebuah kelas, atribut dapat memiliki hak akses privat, public maupun protected. Atribut juga disebut state atau propertis atau field merupakan bagian yang dimiliki oleh sebuah class yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai atribut.

5. Abstraksi (*abstraction*)

Abstraksi adalah cara untuk melihat suatu *object* dalam bentuk yang lebih sederhana. Abstraksi merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses, yaitu kemampuan untuk fokus pada inti.

6. Enkapsulasi (*encapsulation*)

Enkapsulasi merupakan pembungkusan atribut atau property dan fungsi atau *method* dari sebuah *class* sehingga masing-masing dari atribut dan *method* tersebut sehingga hak akses terhadap atribut dan *method* dapat dikendalikan,

7. Pewarisan (*inheritance*)

Pewarisan merupakan salah satu konsep dasar dalam pemrograman berorientasi objek. Konsep ini mengizinkan atribut dan *method* dari *superclass* diwariskan kepada *subclass*. Konsep ini memudahkan developer dalam membuat kode program dan menghilangkan redundansi untuk atribut dan *method* dalam beberapa *class* yang berbeda.

8. Antarmuka (*interface*)

Antarmuka adalah wadah dari kumpulan method yang bersifat abstrak atau tidak memiliki implementasi. Antar muka merupakan bentuk perluasan dari kelas abstrak. Antarmuka berisi sekumpulan variabel, namun variabel yang dideklarasikan di dalam interface harus bersifat final (nilainya tidak dapat diubah/konstan).

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainya yang melibatkan objek tersebut.

10. Generalisasi dan Splesialisasi

Menunjukkan hubungan antar kelas dan objek yang umum dengan kelas dan objek khusus.

11. Komunikasi Antar Objek

Komunikasi antar-objek dilakukan lewat pesan yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme

Polimorfisme merupakan konsep yang digunakan suatu interface sama untuk memerintah suatu object agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda.

13. Package

Package adalah sebuah container atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.

2.2.6 Basis Data

Basis data adalah kumpulan informasi yang disimpan di dalam komouter secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Secara konsep basis data atau database adalah kumpulan dari data-data

yang membentuk suatu berkas yang file berhubungan dengan tatacara yang tertentu untuk membentuk data baru atau informasi^[17].

1. DBMS


DBMS adalah perangkat lunak yang menangani semua pengaksesan *database*. Dengan DBMS, diharapkan *database* dapat dikelola dengan baik dan mudah dalam penggunaannya. DBMS menyediakan beberapa fasilitas sebagai berikut^[18].

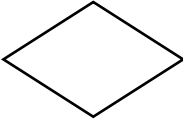
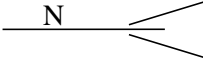


- a. DDL (*Data Definition Language*) memungkinkan pengguna untuk mendefinisikan tipe data (*data type*), struktur (*structure*), dan batasan-batasan (*constraints*) pada data yang disimpan ke dalam *database*.
- b. DML (*Data Manipulation Language*) memungkinkan pengguna untuk memasukkan (*insert*), mengubah (*update*), menghapus (*delete*), dan menampilkan (*retrieve*) data dari *database*.


2. Entity Relationship Diagram (ERD)

ERD adalah jenis diagram structural untuk digunakan dalam desain *database*. ERD berisi symbol dan konektor berbeda yang memvisualisasikan dua informasi penting. Entitas utama dalam ruang lingkup sistem, dan hubungan antar entitas yang lain. Berikut Tabel 2.4 adalah symbol-simbol yang digunakan pada ERD^[19].

Tabel 2. 4 Simbol – Simbol ERD

No.	Simbol	Nama	Keterangan
1.		<i>Entity /</i> Entitas	Merupakan data inti yang akan disimpan atau bakal tabel pada basis data. Penamaan entitas biasanya lebih mengarah pada kata benda dan belum merupakan nama tabel.
2.		Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.		<i>Multivalued /</i> Atribut multinilai	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu

4.		Relasi	Relasi yang menghubungkan antar entitas; biasanya diawali kata kerja
5.		<i>Association</i> / Asosiasi	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakain (kardinalitas)
6.		Relasi Satu ke Satu (<i>One to one</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas berhubungan dengan tepat satu himpunan entitas lainnya
7.		Relasi satu ke Banyak (<i>One to Many</i>)	Relasi yang menunjukkan bahwa hubungan antara entitas

			pertama dengan entitas kedua adalah satu banding banyak tetapi tidak sebaliknya
8.		Relasi Banyak ke Banyak (<i>Many to Many</i>)	Relasi yang menunjukkan bahwa setiap himpunan entitas boleh berhubungan dengan banyak entitas lainnya dan sebaliknya

2.2.7 Framework CI

CodeIgniter (CI) pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. CI menjadi sebuah framework PHP dengan model MVC (*Model, View, Controller*). Selain ringan dan cepat, CI juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kode^[20].

MVC adalah sebuah pola desain arsitektur pengembangan aplikasi yang memisahkan dan mengelompokkan beberapa kode sesuai dengan fungsinya. MVC membagi aplikasi kedalam bagian fungsional : model, view, dan controller^[21].

- 1) Model, Kode-kode untuk model bisnis dan data. Berhubungan langsung dengan database untuk memanipulasi data, menangani validasi dari bagian controller, namun tidak dapat langsung berhubungan dengan view.
- 2) View, Bagian yang menangani *presentation logic*, berisi kode untuk tampilan.
- 3) Controller, Bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.