

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Beberapa jurnal yang berhubungan dengan penelitian untuk dijadikan sebagai bahan referensi antara lain:

Penelitian Rahmat Fauzi (2018) yang berjudul “*Perancangan Aplikasi Marketplace Jasa Percetakan Berbasis Website*” menjelaskan bahwa permintaan mencetak dokumen yang tinggi diruang lingkup kampus ataupun sekolah membuat jasa percetakan sedikit kewalahan dalam menghadapi permintaan percetakan yang tinggi tersebut. Akan tetapi belum banyak yang menyediakan layanan jasa printing atau jasa pemesanan tersebut secara *online* dan *real time*. Maka dari itu, dibuatlah sebuah aplikasi *marketplace* jasa percetakan yang dimana dapat memberikan nilai tambah bagi pemilik jasa percetakan maupun masyarakat seperti pencarian jasa percetakan, melihat perbandingan harga antar jasa percetakan dan menghemat waktu tanpa harus mengantri lama. Penulis merancang Aplikasi Marketplace Jasa Percetakan ini menggunakan metode *SDLC* (*System Development Life Cycle*) dengan pendekatan model *Waterfall*. Metode *SDLC* (*System Development Life Cycle*) *Waterfall* ini merupakan metode pengembangan perangkat lunak secara berurutan dimana seluruh proses pengerjaan memiliki tahapan yang berkembang secara sistematis terus mengalir dari atas ke bawah seperti air terjun. Hasil dari penelitian adalah aplikasi yang dapat menyediakan jasa percetakan secara efisien dan efektif. Masyarakat juga bisa membandingkan harga percetakan yang satu dengan percetakan yang lain. Selain itu, pemilik jasa percetakan juga bisa mempromosikan jasa percekannya sehingga dikenal oleh masyarakat luas [5].

Penelitian Putry Noviandhiny (2018) dalam jurnalnya yang berjudul “*Rancang Bangun Aplikasi Penjualan dan Pembelian Berbasis Web Pada Apotek Neofarma Sanggau*”. Penulis menjelaskan Apotek adalah suatu usaha yang bergerak dalam bidang obat-obatan. Salah satunya Apotek Neofarma. Sistem yang berjalan diapotek ini juga masih menggunakan cara sederhana,

semua transaksi masih didata dengan mencatatnya ke dalam sebuah buku. Kendala yang seringkali muncul yaitu saat melakukan pengecekan data obat memakan waktu terlalu lama dan pencatatan obat kadaluarsa yang dilakukan masih belum terkontrol satu sama lain. Tentu saja hal ini dapat mengakibatkan kesulitan dalam mencari informasi persediaan obat dan obat yang hampir mendekati masa kadaluarsa. Maka dari itu, dibuatlah sebuah aplikasi penjualan dan pembelian pada Apotek Neofarma Sanggau, guna mengatasi masalah yang ada pada apotek tersebut. Penulis merancang aplikasi menggunakan metode *System Development Life Cycle (SDLC)* dengan menggunakan model *Waterfall* sebagai pendekatan sistematis dan berurutan untuk pengembangan perangkat lunak. Hasil dari penelitian ini adalah berupa aplikasi yang dapat memberikan informasi mengenai stok obat serta dapat melakukan transaksi jual beli obat hingga mencetak laporan [6].

Penelitian Silvia Sofian (2017) dalam jurnalnya yang berjudul "*Penerapan Sistem Informasi E-Commerce Berbasis Web (Studi Kasus Untuk Vendor Pernikahan)*" menjelaskan bahwa kurangnya waktu dan informasi yang diperlukan guna mempersiapkan pernikahan menjadi permasalahan tersendiri bagi setiap orang. Untuk itu dibutuhkan sebuah sistem informasi berbasis website yang dapat memberikan penawaran penyedia jasa pernikahan sehingga calon pengantin dapat memilih jasa pernikahan yang sesuai dengan konsep dan keinginan mereka. Serta diperlukan sebuah sistem informasi yang mampu mencari dan membandingkan penyedia jasa pernikahan yang sesuai dengan kebutuhan dari calon pengantin tersebut. Maka dari itu penulis merancang Sistem Informasi *E-Commerce* berbasis Website untuk penyedia jasa pernikahan dan calon pengantin. Metode yang digunakan penulis untuk merancang sistem informasi *e-commerce* berbasis website ini adalah Metode *Waterfall*. Hasil dari penelitian ini berupa sistem informasi yang dapat memberikan nilai lebih bagi pemilik jasa pernikahan maupun bagi calon pengantin [7].

Penelitian-penelitian diatas tersebut, hampir sama dengan yang diteliti oleh peneliti dalam mengimplementasikan Aplikasi *Marketplace* Jasa *Catering* menggunakan *Framework Codeigniter* di Bangsri. Tetapi dari

penelitian tersebut diatas secara umum bisa diperoleh informasi-informasi yang dibutuhkan oleh peneliti. Dengan adanya ketiga referensi jurnal yang ada, peneliti akan melakukan penelitian untuk membuat sebuah aplikasi *marketplace* jasa *catering*.

2.2 Tinjauan Pustaka

2.2.1 Perancangan

Perancangan sistem merupakan tahapan awal dari analisa sistem yang dirancang sedemikian rupa agar dapat memberikan skema yang jelas dan dapat digunakan untuk memecahkan masalah supaya dapat dikomputerisasi. Selain itu perancangan ini juga dapat memeberikan bantuan guna menunjang pengolahan data yang cepat dan akurat, sehingga dapat mengurangi kesalahan yang tak diinginkan [8].

2.2.2 Web

Website merupakan sebuah media informasi yang ada di *internet*. *Website* juga merupakan sekumpulan dari halaman situs yang terangkum dalam sebuah domain atau subdomain yang berada didalam *World Wide Web* (WWW) di Internet. Halaman *web* adalah sebuah dokumen yang ditulis dalam format *HTML* (*Hyper Text Markup Language*) dan hampir selalu bisa diakses melalui *HTTP* (*Hyper Text Transfer Protocol*). *HTTP* sendiri yaitu *protocol* yang fungsinya menyampaikan informasi dari *server website* untuk ditampilkan kepada para *user* melalui *web browser*. Semua publikasi dari *website* tersebut akan membentuk sebuah jaringan informasi yang sangat besar. Halaman-halaman dari *website* bisa diakses melalui sebuah *URL* yang biasa disebut *Homepage*. *URL* ini mempunyai fungsi mengatur halaman-halaman situs untuk menjadi sebuah hirarki meskipun *hyperlink-hyperlink* yang ada di halaman tersebut mengatur para *user* dan memberitahu mereka susunan ini secara keseluruhan. Beberapa *website* membutuhkan suatu *subskripsi* (data masukan) agar para *user* bisa mengakses sebagian atau keseluruhan isi *website* tersebut [9].

Kelebihan ini yang menjadikan *web* sebagai *service* yang paling cepat pertumbuhannya. Web juga mengijinkan pemberian *highlight*

(penyorotan atau penggaris bawahan) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti *frase*, *movie clip*, file suara dan lain-lain. Web dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke sembarang tempat di dokumen lain. Dengan sebuah browser yang memiliki *Graphical User Interface* (GUI), link-link dapat dihubungkan ke tujuannya dengan menunjuk link tersebut dengan mouse dan menekannya [10].

2.2.3 Marketplace

Marketplace merupakan suatu bentuk pasar elektronik (*virtual market*) dimana penjual dan pembeli bertemu dan dihubungkan melalui suatu transaksi elektronik (*online*) yang mungkin dapat diakses dengan cepat, aman, kapanpun dan dimanapun [11]. Banyak keuntungan yang dapat diambil dari adanya *marketplace* ini diantaranya:

- 1) Mempermudah pencarian dan membandingkan produk beserta segala informasi terbaru.
- 2) Mengurangi biaya pengadaan barang.
- 3) Dapat menjadi sarana untuk mempromosikan produk.
- 4) Mengurangi biaya transaksi dan *sales*.

2.2.4 Framework CodeIgniter

CodeIgniter (CI) merupakan framework pemrograman PHP mengimplementasi teknik pemrograman MVC dengan pendekatan yang fleksibel. CodeIgniter juga dibangun untuk para *dev* dengan pemrograman PHP yang membutuhkan alat untuk membuat sebuah website dengan fitur yang lengkap. Framework CodeIgniter sendiri dikembangkan oleh CEO Ellislab Inc, Rick Ellis [12]. Adapun kelebihan tersendiri framework codeigniter ini yaitu :

1. Gratis

Framework Codeigniter mempunyai lisensi dibawah *Apache/BSD open-source* sehingga bersifat gratis.

2. Berukuran sangat kecil

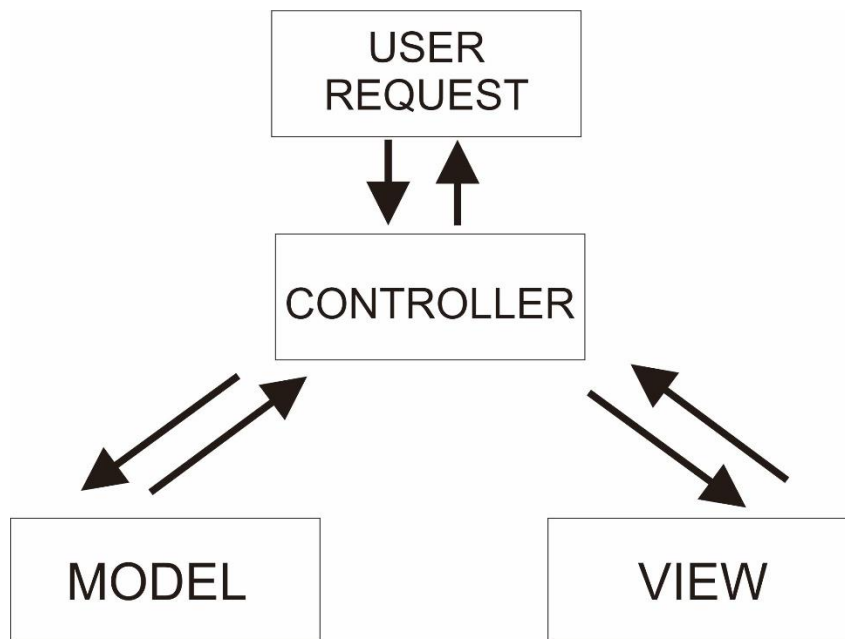
Ukurannya yang sangat kecil merupakan salah satu keunggulan tersendiri jika dibandingkan framework lain yang membutuhkan space yang besar dan membutuhkan resource yang besar juga.

3. Menggunakan konsep MVC

Codeigniter menggunakan konsep MVC yaitu *Model, View dan Controller* yang gunanya memisahkan antara *layer application-logic* dengan *presentation*. Dengan menggunakan konsep ini, kode PHP, query MySQL, Javascript dan CSS dapat dipisahkan sehingga hanya memakan space yang kecil dan lebih mudah diperbaiki apabila terjadi kesalahan.

- a) Model, merupakan struktur data dari aplikasi yang berupa basis data maupun data lain. Biasanya dimodel ini berisi *class* atau *function* yang gunanya untuk mengambil data, melakukan update data dan menghapus data suatu website.
- b) View, berfungsi untuk menerima dan mempresentasi kan data dari hasil model dan controller ke user interface.
- c) Controller, berfungsi untuk mengatur hubungan atau aliran antara model dan view. Pada controller ini juga terdapat *class* dan *function* yang memproses permintaan dari view ke dalam data didalam model.

Adapun diagram dari program framework CodeIgniter yang menggunakan konsep MVC ini ditunjukkan sebagai berikut:



Gambar 2.1 Alur MVC [12].

2.2.5 PHP

Hypertext Preprocessor atau sering disebut PHP ini adalah salah satu bahasa pemrograman bersifat *open source* yang sangat cocok atau dirancang secara khusus untuk pengembangan *web* dan dapat ditanamkan pada sebuah skrip HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman lain seperti C, *Java* dan *Perl*. PHP juga merupakan bahasa pemrograman *scripting server-side* yang mana pemrosesan datanya dilakukan pada sisi *server*. Secara sederhana, *server* yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain dari PHP atau *Hypertext Preprocessor* ini, yaitu bahasa pemrograman berbasis kode atau *script* yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* dalam bentuk kode HTML. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*,

mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode *HTML* dan menampilkannya.

```

1  <?php
2      echo "Hello World!";
3  ?>

```

Gambar 2.2 Contoh syntax PHP [13].

Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe *file* yang diminta *user*. Jika tipe *file* yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut [13].

2.2.6 MySQL

MySQL adalah sebuah *Database Manajemen System* (DBMS) yang memiliki fungsi sebagai *Relational Database Manajemen System* (RDBMS). MySQL juga merupakan aplikasi yang sifatnya *open source* juga server basis data MySQL memiliki kinerja yang cepat, fleksibel dan mudah untuk dioperasikan serta bekerja dengan arsitektur *client server* atau *embedded system*. Dikarenakan sifatnya yang *open source* maka cocok digunakan untuk mendemonstrasikan proses replikasi basis data. Adanya fasilitas *Application Programming Interface* (API) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman lain yang dapat mengakses basis data MySQL. Tipe data MySQL adalah tipe data yang terdapat dalam sebuah table berupa field-field yang berisi nilai dari data tersebut. Nilai data dalam field juga memiliki tipe tersendiri [14]. Adapun perintah SQL yang digunakan untuk mengelola database adalah sebagai berikut:

1. DDL

DDL atau *Data Definition Language* ini digunakan untuk mengoperasikan struktur pada database, perintah yang sering

digunakan DDL adalah CREATE (membuat database atau table), RENAME (mengubah nama database atau table), ALTER (mengubah struktur table) dan DROP (menghapus database atau table). Contoh:

```
Mysql> CREATE DATABASE PERPUSTAKAAN;

Mysql> ALTER TABLE buku add pengarang
varchar(100);

Mysql> DROP TABLE buku;
```

2. DML

DML atau *Data Manipulation Language* ini digunakan untuk mengelola database, perintah yang sering digunakan adalah UPDATE (*update* atau pembaruan *data*), DELETE (menghapus data), INSERT (memasukkan data baru) dan SELECT (menampilkan data). Contoh:

```
Mysql> SELECT * FROM customers;

Mysql> UPDATE `customers` SET `name` = 'Anto'
WHERE `customers`.`id` = 2;

Mysql> INSERT INTO `customers` (`id`, `name`,
`email`, `address`, `phone`) VALUES ('334', 'Khoirul
Umam', 'umam.khoirul@gmail.com', 'Mlonggo',
'085234879XXX');
```

2.2.7 Sublime Text

Sublime text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, cross platform, mudah dan simpel yang cukup terkenal di kalangan para *dev*, penulis dan *designer*. Para *dev* atau *programmer* biasanya menggunakan sublime text untuk menyunting *source code* yang sedang ia kerjakan. Hingga sekarang sublime text sudah mencapai versi 3 (build 3207). Sublime text juga mempunyai beberapa keunggulan yang dapat membantu pengguna dalam membuat sebuah *web*

development. Berikut ini adalah kelebihan sublime text 3 diantaranya, *Multiple Selection, Command Profile, Distraction Free Mode, Find in Project, Plugin API Switch, Drag and Drop* dan *Split Editing* [15].

2.2.8 XAMPP

XAMPP adalah suatu perangkat lunak bebas yang mendukung banyak sistem operasi merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*) terdiri atas program *Apache HTTP Server, MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL, PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bersifat bebas yang merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya bisa langsung *men-download* dari web resminya di <https://www.apachefriends.org/index.html> [10].

2.2.9 UML

Unified Modelling Language atau sering disebut UML ini adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok penulisannya untuk perangkat lunak dalam bahasa yang berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB (*Visual Basic*) atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah

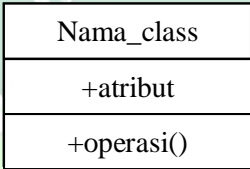


ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering) [16].





2.2.8.1 Class Diagram

Diagram Class atau kelas *diagram* menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem. *Class* menggambarkan suatu keadaan (atribut/properti) sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [16]. *Class* memiliki tiga area pokok:

1. Nama (Stereotype)
2. Atribut
3. Metoda

Tabel 2.1 Class Diagram

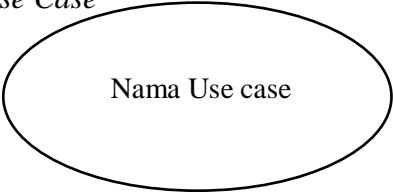
No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas struktur sistem
2.	<p>Antarmuka/interface</p>  <p>Nama_interface</p>	Memiliki konsep yang sama dengan interface di pemrograman berorientasi objek.
3.	<p>Asosiasi/Association</p> 	Relasi atau hubungan antarkelas yang biasanya disertai multiplicity

4.	Asosiasi berarah 	Relasi atau hubungan antarkelas yang bermakna kelas yang digunakan serta kelas lain yang biasanya disertai multiplicity
5.	Generalisasi 	Relasi atau hubungan antarkelas bermakna generalisasi-spesialisasi
6.	Ketergantungan/dependency 	Relasi atau hubungan antarkelas bermakna ketergantungan antarkelas
7.	Agregasi/agregation 	Relasi atau hubungan antarkelas bermakna semua bagian (whole-part)

2.2.8.2 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* juga digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut [17]. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

Tabel 2.2 Use Case





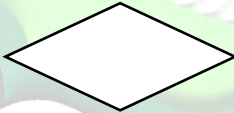
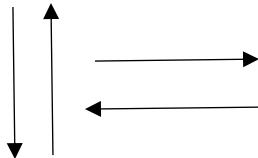
Gambar	Keterangan
<p><i>Use Case</i></p> 	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bentukan pesan antar unit dengan aktor, biasanya dinyatakan dengan</p>

	menggunakan kata kerja diawal nama <i>use case</i> .
<p><i>Aktor</i></p> 	<i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk dapat mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks sistem.
<p><i>Asosiasi</i></p> 	Asosiasi antara dua aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa saja atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
<p><i>Generalisasi</i></p> 	Asosiasi antara dua aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif.
<p><i>Include</i></p> 	<i>Use Case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, misal pemanggilan sebuah program.
<p><i>Extend</i></p> 	Perluasan dari use case lain jika kondisi atau syarat terpenuhi.

2.2.8.1 Activity Diagram

Activity Diagram atau Diagram aktivitas adalah suatu aliran kerja (workflow) atau aktivitas dari sebuah sistem atau proses [17]. Adapun simbol-simbol yang digunakan dalam activity diagram yaitu sebagai berikut:



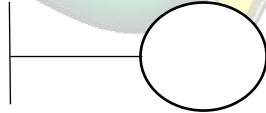
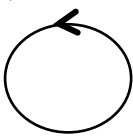
Tabel 2.3 Activity Diagram

No	Simbol	Nama Simbol	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri
5.		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu.
6.		<i>Line Connector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

2.2.8.2 Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek tertentu. Bentuknya berupa *message* yang digambarkan terhadap waktu dan pesan yang menggambarkan langkah-langkah yang dilakukan sebagai tanggapan dari sebuah event untuk menghasilkan output tertentu. Dengan menggunakan *sequence diagram*, pengembang bias menjelaskan interaksi apa yang akan dipanggil ketika sebuah use case dieksekusi dan urutan seperti apa interaksi –interaksi tersebut akan terjadi [18]. Berikut ini adalah table dari sequence diagram:

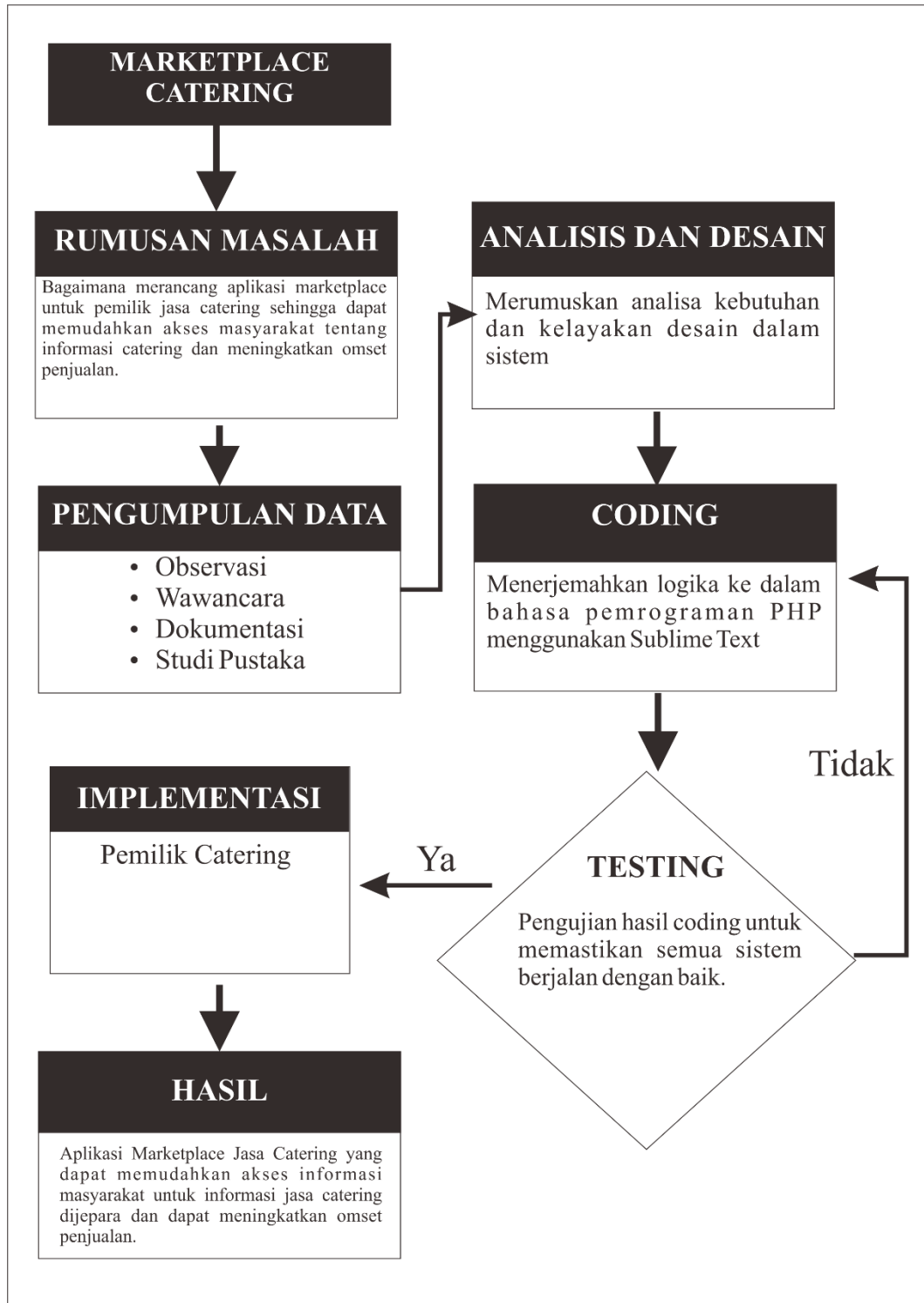
Tabel 2.4 Sequence Diagram

No	Simbol	Keterangan
1.	Aktor 	Menggambarkan siapa yang berinteraksi dengan sistem.
2.	Entity (Nama Tabel) 	Menggambarkan hubungan kegiatan yang akan dilakukan.
3.	Boundary (Interface) 	Menggambarkan sebuah penggambaran dari form.
4.	Control (Pembacaan) 	Menggambarkan penghubung antara boundary dengan tabel.

5.	Lifeline (Garis hidup) 	Menggambarkan tempat mulai dan tempat berakhirnya sebuah pesan.
6.	Message (Pesan) Pesan →	Menggambarkan sebuah pengiriman pesan.



2.2.10 Kerangka Pemikiran



Gambar 2.3 Kerangka Pemikiran