

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Menurut Heriyanto, Ilham dan Irfan (2020) dalam penelitiannya yang berjudul **“Sistem Aplikasi Perpustakaan Berbasis Web Menggunakan Framework Codeigniter pada SMAS Islamic Centre”** menjelaskan bahwa Sistem aplikasi peminjaman dan pengembalian buku ini merupakan sistem aplikasi yang diharapkan bisa membantu staff perpustakaan dalam mengelola data perpustakaan dengan efektif dan efisien. Selain itu juga demi tercapainya tujuan sistem itu sendiri yaitu memudahkan pustakawan atau pustakawati dalam mencari buku dan kemudahan dalam segala hal. Dalam pembuatan aplikasi ini juga ada data pendukung yang di perlukan di antaranya data anggota, buku, transaksi, dan alur dari sistem yang sebelumnya. Dari data-data tersebut diharapkan bisa menjadi dasar pembuatan aplikasi ini agar dapat menyajikan informasi yang di butuhkan dengan cepat dan efisien, serta untuk meningkatkan kinerja petugas perpustakaan. Persyaratan anggota perpustakaan adalah siswa/siswi SMAS ISLAMIC CENTRE. Kemudian membuat formulir pendaftaran untuk menjadi anggota siswa/siswi SMAS ISLAMIC CENTRE. Hasil yang di harapkan dari pembuatan sistem ini adalah agar data-data yang selama ini disimpan dan di kelola secara manual dapat mulai dikomputerisasikan sehingga dapat mengefektifkan kinerja serta mempermudah petugas perpustakaan di dalam menjalankan kegiatan yang berhubungan dengan perpustakaan itu sendiri.

Menurut Pratama, dan Crishtioko (2021) dalam penelitiannya yang berjudul **“Aplikasi Sistem Perpustakaan Berbasis Web Menggunakan Codeigniter”** menjelaskan bahwa Perpustakaan adalah kumpulan atau bangunan fisik sebagai tempat buku dikumpulkan dan disusun menurut sistem tertentu atau keperluan pemakai, berfungsi sebagai laporan petugas kepada guru kelas dan kepla sekolah. Dalam perpustakaan tersebut, petugas untuk pengelolaan masih menggunakan cara manual *book* untuk peminjaman dan pengembalian buku serta arsip data buku yang berada di perpustakaan. Data laporan pengunjung yang ada hanyalah data tertulis yang setiap bulan di laporkan kepada kepala sekolah, petugas melaporkan rincian

data pengunjung perbulan dan data buku yang tersimpan di perpustakaan. Oleh karena itu, perlu dikembangkan sistem perpustakaan berbasis *web* dengan menggunakan *framework codeigniter*. Dalam melakukan penelitian, penulis menggunakan metode *prototype* yaitu komunikasi, perancangan secara cepat, pemodelan perancangan secara cepat, pembentukan *prototype*, dan penyerahan sistem. Alat bantu penelitian yang digunakan adalah *UML (Unified Modeling language)*, *PHP* dan *MySQL*. Dengan adanya sistem ini maka diharapkan dapat membantu pihak sekolah dalam mengelola perpustakaan dari transaksi peminjaman buku dan pengembalian buku cetak laporan dan memudahkan kinerja petugas perpustakaan sehingga pekerjaan menjadi lebih efisien dan cepat.

Menurut Yahya (2020) dalam penelitiannya yang berjudul **“Rancang Bangun Aplikasi Perpustakaan Menggunakan *Framework CodeIgniter* (Studi Kasus SDN Cibubur 05)”** menjelaskan bahwa Perpustakaan merupakan bagian dari sumber belajar yang harus dimiliki oleh setiap sekolah atau perguruan tinggi, karena dapat memudahkan peserta didik dalam mencari informasi atau ilmu pengetahuan. Dengan adanya perkembangan teknologi membuat manusia berfikir untuk dapat bekerja lebih efektif dan efisien. Salah satunya yaitu membuat sistem konvensional menjadi sistem yang terkomputerisasi. Dengan memanfaatkan fasilitas *website* yang terhubung ke internet, perpustakaan dapat lebih efektif dan efisien dalam pencarian dan peminjaman buku. Pada penelitian ini dirancang suatu aplikasi perpustakaan berbasis *web* menggunakan *Framework Codeigniter*. Dengan sistem ini, diharapkan mampu mengatasi berbagai kebutuhan dari *user* untuk mencari buku, *filter* buku yang sering dipinjam, pencatatan data anggota, pencatatan data buku, pengembalian buku, perhitungan denda serta memudahkan administrasi sekolah dalam sirkulasi peminjaman buku dan pembuatan laporan. Adapun metodologi pengembangan sistem yang digunakan adalah *System Development Life Cycle model Waterfall* yang meliputi tahapan perencanaan, analisis perancangan, dan implementasi sistem. Sistem dibangun menggunakan bahasa pemrograman *Hypertext Pre Processor* dan *database* menggunakan *mysql*. Dari hasil penelitian yang telah dilaksanakan penulis mengimplementasikan hasil penelitian tersebut kedalam Rancang Bangun Aplikasi Perpustakaan Berbasis *Web* di Sekolah Dasar Negeri Cibubur 05 Jakarta Timur.

Perbedaan dengan penelitian yang dilakukan sekarang dengan penelitian terdahulu dapat dilihat pada tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No	Judul Penelitian	Persamaan	Perbedaan
1.	"Sistem Aplikasi Perpustakaan Berbasis Web Menggunakan <i>Framework Codeigniter</i> pada SMAS <i>Islamic Centre</i> " (Heriyanto, et al., 2020)	Menggunakan <i>database Mysql</i> , Bahasa Pemrograman <i>PHP</i> dan Menggunakan <i>Framework CodeIgniter</i>	Aplikasi sekarang menggunakan <i>QR Code</i>
2.	"Aplikasi Sistem Perpustakaan-an Berbasis Web Meng-gunakan <i>Codeigniter</i> " (Pratama dan Crishtioko, 2021)	Menggunakan <i>database Mysql</i> , Bahasa Pemrograman <i>PHP</i> dan Menggunakan <i>Framework CodeIgniter</i>	Aplikasi sekarang menggunakan <i>QR Code</i>
3.	"Rancang Bangun Aplikasi Perpustakaan Menggunakan <i>Framework Codeigniter</i> (Studi Kasus Sdn Cibubur 05)" (Yahya, 2020)	Menggunakan <i>database Mysql</i> , Bahasa Pemrograman <i>PHP</i> dan Menggunakan <i>Framework CodeIgniter</i>	Aplikasi sekarang menggunakan <i>QR Code</i>

Pada table 2.1. Pertama dengan judul penelitian "Sistem Aplikasi Perpustakaan Berbasis Web Menggunakan *Framework Codeigniter* pada SMAS *Islamic Centre*". Mempunyai persamaan pada aplikasi yaitu sama-sama menggunakan *database mysql*, bahasa pemrograman *PHP* dan menggunakan *Framework CodeIgniter*. Mempunyai perbedaan yaitu pada aplikasi sekarang menggunakan *QR Code*.

Kedua dengan judul penelitian "Aplikasi Sistem Perpustakaan Berbasis Web Menggunakan *Codeigniter*". Mempunyai persamaan pada aplikasi yaitu sama-sama menggunakan *database mysql*, bahasa pemrograman *PHP* dan menggunakan *Framework CodeIgniter*. Mempunyai perbedaan yaitu pada aplikasi sekarang menggunakan *QR Code*.

Ketiga dengan judul penelitian "Rancang Bangun Aplikasi Perpustakaan Menggunakan *Framework Codeigniter* (Studi Kasus Sdn Cibubur 05)". Mempunyai persamaan pada aplikasi yaitu sama-sama menggunakan *database mysql*, bahasa pemrograman *PHP* dan menggunakan *Framework CodeIgniter*. Mempunyai perbedaan yaitu pada aplikasi sekarang menggunakan *QR Code*.

2.2 Teori Perpustakaan Sekolah

Perpustakaan sekolah merupakan salah satu sarana pendidikan di sekolah yang walaupun keberadaannya masih kurang sekali, telah banyak diusahakan peningkatan dan penggunaannya untuk perkembangan sekolah, pengembangan perpustakaan sekolah sangat penting untuk sarana pendidikan, dimana perpustakaan yang menyimpan buku-buku pelajaran yang menyangkut dengan kurikulum sekolah yang bersangkutan.

Adapun pengertian perpustakaan sekolah menurut Reitz dalam Hasugian (2009:78) juga menjelaskan perpustakaan sekolah adalah "*(School library), A library in a public or private elementary or secondary school that serves the information needs of its students and curriculum needs of its teachers and staff, usually managed by a school librarian or media specialist*". Definisi di atas menyatakan bahwa perpustakaan sekolah adalah suatu perpustakaan yang beradapada jenjang sekolah dasar sampai dengan sekolah lanjutan baik milik pemerintah(negeri) maupun swasta yang melayani kebutuhan informasi siswanya, kebutuhan kurikulum dari guru dan staf, biasanya dikelola oleh pustakawan sekolah ataupun spesialis media.

Berdasarkan pengertian di atas dijelaskan bahwa perpustakaan sekolah adalah bagian integral dari sekolah dan merupakan sumber belajar yang akan dikelola oleh petugas yang menyajikan berbagai jenis bahan pustaka serta melayani kebutuhan informasi siswa dan siswi untuk menunjang proses pendidikan dan

pengajaran yang layak serta mendukung tercapainya tujuan pendidikan sekolah yang bersangkutan.

2.3 Landasan Teori

Landasan teori adalah pernyataan yang disusun secara sistematis dan memiliki variabel yang kuat. Landasan teori biasanya memuat teori-teori dan hasil penelitian, di mana teori dan hasil penelitian digunakan sebagai kerangka teori peneliti untuk menyelesaikan sebuah penelitian.

Secara umum, landasan teori juga dapat diartikan sebagai asumsi secara eksplisit terhadap sebuah teori yang akan dilakukan evaluasi dan penelitian kritis. Kerangka teori berperan untuk menghubungkan pada pengetahuan baru.

2.3.1 Website

Pengertian *Web* atau Situs *Website* atau situs dapat diartikan sebagai kumpulan halaman - halaman yang digunakan untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, dan atau gabungan dari semuanya itu, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling berkait dimana masing masing dihubungkan dengan jaringan-jaringan halaman / *hyperlink*. (Novendri et al., 2019)

Website atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Josi, 2017)

2.3.2 Basis Data

Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. (Josi, 2017)

Basis data sebagai kumpulan terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. (Novendri et al., 2019)

2.3.3 PHP

PHP singkatan dari *Hypertext Preprocessor* 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*. (Josi, 2017)

Perl Hypertext Preprocessor atau disingkat *PHP* adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan *PHP* harus menggunakan *web server*. *PHP* juga dapat diintegrasikan dengan *HTML*, *JavaScript*, *JQuery*, *Ajax*. Namun, pada umumnya *PHP* lebih banyak digunakan bersamaan dengan *file* bertipe *HTML*. Dengan menggunakan *PHP* anda bisa membuat *website* yang dinamis dengan disertai manajemen *database*-nya. (Hidayatullah, 2017)

2.3.4 XAMPP (X Apache Mysql PHP Perl)

Xampp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah *server*. (Josi, 2017)

Xampp merupakan singkatan dari X (empat sistem operasi apapun), *Xampp* merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Seperti *Apache*, *MYSQL*, *PHP*, dan *Perl*. *Xampp* adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket *Xampp* sudah terdapat *Apache (Web Server)*, *Mysql (Database)*, *PHP (server side scripting)*, *Perl*, *FTP server*, *PhpMyAdmin*, dan berbagai pustaka bantu lainnya. (Novendri et al., 2019)

2.3.5 Qr Code

Qr code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2 dimensi yang tercetak kedalam suatu media yang lebih ringkas. *Qr code* adalah *barcode* 2 dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang Denso Wave pada tahun 1994. *Barcode* ini pertama kali digunakan untuk pendataan inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. *Qr* adalah singkatan dari *Quick Response* karena ditujukan untuk

diterjemahkan isinya dengan cepat. *QR-Code* salah satu tipe dari *barcode* yang dapat dibaca dengan kamera *handphone* (Irawan et al., 2018).

2.3.6 *CodeIgniter*

CodeIgniter merupakan sebuah *framework PHP* yang bersifat *open source* untuk memudahkan *programmer* atau *developer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal dengan menggunakan metode MVC (*Model, View, Controller*). *CodeIgniter* (CI) memiliki *library* yang lengkap untuk mengerjakan operasi - operasi yang umum dibutuhkan oleh aplikasi berbasis *web* misalnya mengakses *database*, memvalidasi *form* sehingga sistem yang dikembangkan mudah (Sallaby dan Kanedi, 2020).

Metode yang digunakan oleh *Framework CodeIgniter* disebut *Model -View-Controller* atau yang disingkat dengan sebutan MVC. MVC memisahkan antara logika pemrograman dengan presentasi. Hal ini dapat terlihat dari adanya minimalisir *script* presentasi (*HTML, CSS, JavaScript*, dan sebagainya) yang dipisahkan dari PHP (*Hypertext Preprocessor*) *script*. Didalam folder *CodeIgniter*, MVC dapat kita temukan dalam folder *application*. *CodeIgniter* juga menjadi salah satu *Framework* pilihan yang memungkinkan *developer* untuk membuat sebuah aplikasi web dengan karakter pengembangan *RAD (Rapid Application Development)*, yang memungkinkan untuk digunakan dan dikembangkan menjadi aplikasi lain yang lebih kompleks. *CodeIgniter* terdiri dari file-file pustaka (*library*), kelas-kelas, dan infrastruktur runtime yang terinspirasi oleh *Framework Ruby on Rails* (Irawan dan Sulistyowati, 2017).

2.3.7 Aplikasi

Menurut Suhartini, Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru (Siregar et al, 2018).

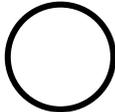
Menurut Hasan Abdurahman dan Asep Ririh Riswaya, Aplikasi adalah program siap pakai yang dapat digunakan untuk menjelaskan perintah – perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai denngan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu

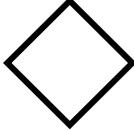
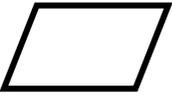
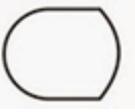
pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. (Novendri et al, 2019)

2.4 Flowchart

Definisi *Flowchart* adalah gambaran dalam bentuk diagram alir dari *algoritma-algoritma* dalam suatu *program*, yang menyatakan arah alur program tersebut. Menurut Jogiyanto, menyatakan pendapat bahwa “bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir atau arus (*flow*) di dalam program atau prosedur sistem secara logika”. Sistem *flowchart* tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk. *Flowchart* sendiri menggambarkan suatu sistem peralatan komputer yang digunakan dalam memproses dan mengolah data serta hubungan antar peralatan yang saling terkait tersebut. (Wahyudi et al., 2019). Simbol-simbol *flowchart* yang biasa digunakan dapat dilihat pada tabel 2.2.

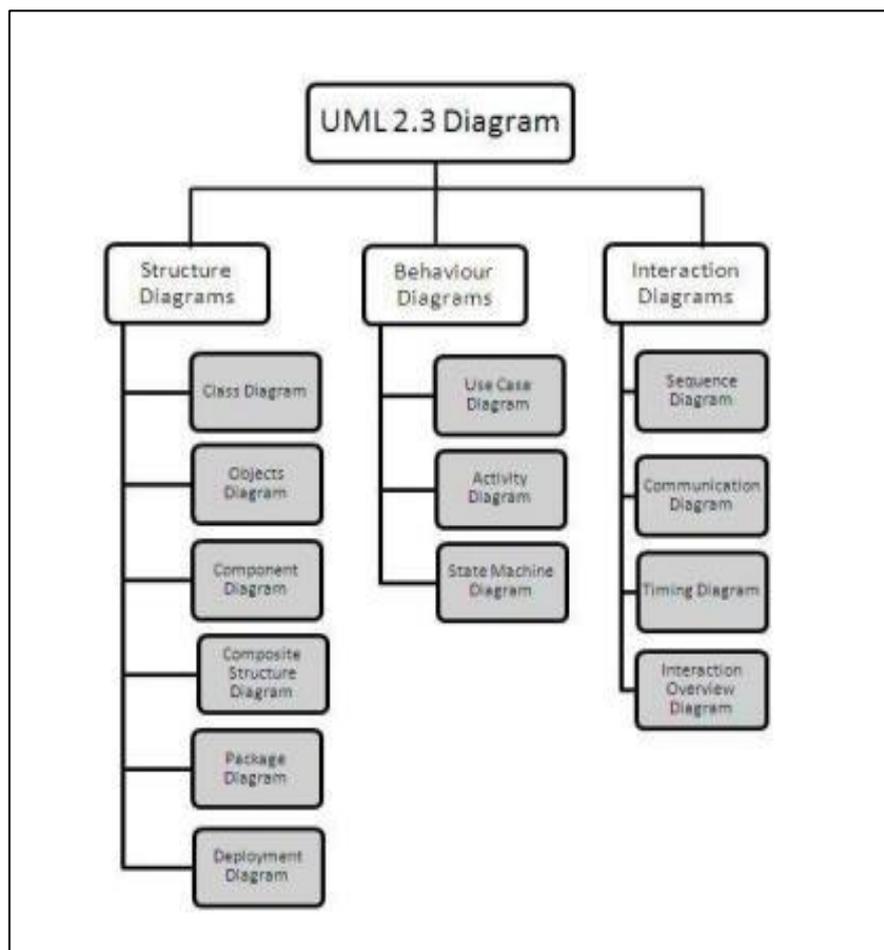
Tabel 2.2 Simbol *Flowchart* dan Keterangan

Simbol	Keterangan
	Flow Direction System yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connecting line</i> .
	Terminator Symbol yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
	Connector Symbol yaitu simbol untuk keluar-masuk atau penyambungan proses dalam lembar / halaman yang sama.
	Symbol Manual Input yaitu simbol untuk pemasukan data secara manual <i>on-line keyboard</i> .
	Symbol preparation yaitu simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .

Simbol	Keterangan
	<p><i>Symbol Predefine Proses</i> yaitu simbol untuk pelaksanaan suatu bagian (<i>sub-program</i>) atau <i>procedure</i>.</p>
	<p><i>Connector Symbol</i> yaitu simbol untuk keluar–masuk atau penyambungan proses pada lembar / halaman yang berbeda.</p>
	<p><i>Processing Symbol</i> yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.</p>
	<p><i>Symbol Manual Operation</i> yaitu simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer</p>
	<p><i>Symbol Decision</i> yaitu symbol pemilihan proses berdasarkan kondisi yang ada.</p>
	<p><i>Symbol Input–Output</i> yaitu simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.</p>
	<p><i>Symbol Display</i> yaitu simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, <i>plotter</i>, <i>printer</i> dan sebagainya.</p>
	<p><i>Symbol Disk And On-Line Storage</i> yaitu simbol yang menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i>.</p>
	<p><i>Symbol Magnetik Tape Unit</i> yaitu simbol yang menyatakan <i>input</i> berasal dari pita magnet atau <i>output</i> disimpan ke pita magnetik.</p>
	<p><i>Symbol Documen</i> yaitu simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.</p>

2.5 *Unifie Modelling Language (UML)*

UML(*Unified Modeling Language*) adalah standar bahasa pemodelan dalam mengembangkan perangkat lunak yang menggunakan teknik pemrograman berbasis objek (A.S, Rosa. & Shalahuddin, 2018). UML dibuat agar dapat memenuhi kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari perangkat lunak atau sistem yang akan dibangun yang memakai diagram dan teks pendukung.



Gambar 2.1 Struktur UML

2.6 *Use Case Diagram*

Use case diagram adalah sebuah diagram yang digunakan untuk menggambarkan hubungan antara aktor dan use case dalam analisis dan desain sistem. Diagram ini memberikan gambaran visual tentang aksi atau tindakan yang dilakukan oleh aktor dalam interaksi dengan sistem (Arianti, et al., 2022) Berikut ini adalah bagian dari sebuah use case diagram :

1. *Use Cases*

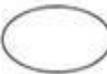
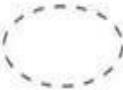
Use cases menjelaskan tentang tindakan/aksi yang dilakukan oleh actors. *Use case* digambarkan dalam bentuk elips yang 9 horizontal.

2. *Actors*

Actors adalah seorang peran yang berinteraksi dengan sistem. Actors meliputi baik manusia maupun organisasi yang saling bertukar informasi.

3. *Relationship*

Relationship adalah hubungan antara use cases dengan actors. *Relationship* dalam *use case* diagram meliputi Asosiasi antara actor dan use case, asosiasi antara 2 use Case dan generalisasi antara 2 actor.

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2.2 Lambang *Use Case* Diagram

2.7 Activity Diagram

Activity diagram, dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity* diagram merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas (Prasetya et al., 2021). *Activity* diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu system Berikut ini merupakan komponen dalam *activity* diagram menurut *Journal of Object Technology: Conrad Bock* yaitu: (Gutama, Arwan, Fanani, 2019).

1. Activity node

Activity node menggambarkan bentuk notasi dari beberapa proses yang beroperasi dalam kontrol dan nilai data

2. Activity edge

Activity edge menggambarkan bentuk edge yang menghubungkan aliran aksi secara langsung ,dimana menghubungkan input dan output dari aksi tersebut .

3. Initial state

Bentuk lingkaran berisi penuh melambangkan awal dari suatu proses.

4. Decision

Bentuk wajib dengan suatu flow yang masuk beserta dua atau lebih *activity node* yang keluar. *Activity node* yang keluar ditandai untuk mengindikasikan beberapa kondisi.

5. Fork

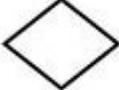
Satu bar hitam dengan satu *activity node* yang masuk beserta dua atau lebih *activity node* yang keluar.

6. Join

Satu bar hitam dengan dua atau lebih *activity node* yang masuk beserta satu *activity node* yang keluar, tercatat pada akhir dari proses secara bersamaan. Semua actions yang menuju join harus lengkap sebelum proses dapat berlanjut.

7. Final state

Bentuk lingkaran berisi penuh yang berada di dalam lingkaran kosong, menunjukkan akhir dari suatu proses.

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.3 Lambang *Activity Diagram*

2.8 *Class Diagram*

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi class, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelaskelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. Diagram kelas ini sesuai jika diimplementasikan ke proyek yang menggunakan konsep *objectoriented* karena gambaran dari class diagram cukup mudah untuk digunakan. (Prasetya et al., 2021).

SIMBOL CLASS DIAGRAM

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2.4 Lambang Diagram

2.9 Metode Waterfall

Waterfall biasanya disebut dengan *model sequential linier* atau *classic cycle*. *Waterfall* ini memberika nsaluran *software* dengan cara sekuensial yang diawali dengan, *design*, pengkodean, pengujian dan *support* (Susilo, 2018). Tahapan dari metode waterfall, sebagai berikut:

1. Analisa Kebutuhan Perangkat Lunak

Tahap ini, mengumpulkan kebutuhan untuk mengetahui apa saja kebutuhan dari perangkat lunak sehingga mudah dipahami oleh pemakai.

2. Desain

Membuat rancangan program perangkat lunak seperti desain data, rancang bangun perangkat lunak, desain antarmuka, dan aturanaturan dalam pengodeannya. Dalam penelitian ini menggunakan *logical* data mode luntuk membuat model basis data. Sedangkan untuk memodelkan kebutuhan perangkat lunak menggunakan use case diagram dan *activity* diagram.

3. Pembuatan Kode Program.

Pada tahap ini mengubah dari desain kedalam bahasa pemrograman dengan menggunakan bahasa pemrograman PHP, HTML, dan CSS.

4. Pengujian

Memastikan semua perangkat lunak sudah berjalan sesuai dengan kebutuhan dengan melakukan pengujian, sehingga meminimalisir kesalahan.

2.10 Black Box Testing

Pengujian *black-box* merupakan salah satu metode pengujian sistem, dimana metode ini berfokus pada pencarian kesalahan dan kekurangan pada fungsional perangkat lunak seperti fungsi, antarmuka, model data, dan akses ke sumber informasi eksternal (Mulyati, Kusyadi, Ashara, Widodo, & Wahyudin, 2022).

Uji *black box*, melakukan pengecekan terhadap fungsi sistem sudah dapat berjalan dengan baik atau tidak. Pengujian menggunakan *black box* hanya mampu menguji sistem dari segi fungsional dan tidak menguji model dan kode program untuk mengetahui fungsi input dan output dari sistem. Uji *black box* dilakukan berdasarkan masukan dan luaran tanpa memperhatikan kode program, sehingga penguji tidak perlu memiliki pengetahuan terkait pemrograman (Shaleh, Yogi, Pirdaus, Syawal, & Saifudin, 2021).

Pengujian *black box* merupakan pengujian yang paling mudah dilakukan, karena pengguna hanya perlu mengisikan batas atas dan batas bawah dari data yang diharapkan. Banyaknya data yang di uji dihitung dari banyak field data entri pada form dalam sistem, aturan entri yang harus dipenuhi serta harus memenuhi batas bawah dan batas atas. Dengan metode *black box* akan mampu mendeteksi fungsionalitas masih dapat menerima masukan data yang tidak diharapkan, sehingga menyebabkan data yang tersimpan kurang valid (Cholifah, Yulianingsih, & Sagita, 2018).

2.11 White Box Testing

Metode pengujian *white box* menekankan pengujian berdasarkan *pada source code* program yang dibuat. Terdapat beberapa pendekatan yang dilakukan

dalam metode pengujian *white box*, dengan berbagai macam metode atau teknik yang dilakukan dalam penelitian sebelumnya. Hal tersebut menjadi sebuah dasar atau petunjuk untuk penulis dalam melakukan sebuah penelitian untuk pengujian perangkat lunak dengan menggunakan metode *white box testing*. Beberapa penelitian yang membahas tentang pengujian perangkat lunak adalah pengujian pada aplikasi inventory Taufan Fish Farm yang menggunakan metode *white box testing* dan *black box testing*. Teknik yang dipakai dalam pengujian *white box testing* yaitu *statement coverage* dan *branch coverage*. Pengujian dilakukan dengan mengambil sampel dari salah satu form di dalam aplikasi Taufan Fish Farm yaitu *login user* yang digunakan untuk memeriksa sebuah masukan *username* dan *password* dari pengguna. Kemudian membuat sebuah skenario uji yang akan menjadi acuan pada proses pengujian (Kurniansyah, Sempati, & Hidayat, 2016).