

BAB II

TINJAUAN PUSTAKA

2.1. Studi Literatur

Pada tahapan pengumpulan data dalam menunjang penelitian ini, penulis melakukan pencarian informasi dan pengetahuan mengenai materi yang dibahas dengan menggunakan jurnal terdahulu yang menjadi sumber referensi dalam penelitian ini. Hasil referensi penelitian serupa yaitu membuat sistem informasi pemesanan berbasis website dengan menggunakan bahasa pemrograman *PHP* disertai metode *Agile Development* dengan kerangka kerja Kanban.

Tabel 2. 1 Penelitian Jurnal Terdahulu

	Title and Author	Location	Method	Result and Conclusion
1.	Judul: Rancang Bangun Sistem Informasi Pemesanan Kue Kota Pontianak Menggunakan Metode Waterfall Tahun: 2021	Pontianak	Model pengembangan sistem menggunakan model pengembangan <i>waterfall</i> .	Dengan dirancangnya aplikasi pemesanan ini membantu efektifitas waktu pekerja menjadi lebih cepat dan efisien. Serta menghasilkan informasi yang cepat, tepat dan akurat dengan adanya <i>database</i> .
2.	Judul: Rancang Bangun Website Pemesanan Makanan Kedai Twins Menggunakan <i>Laravel PHP Framework</i> Tahun: 2020	Kedai Twins	Pendekatan pengembangan sistem penelitian ini adalah dengan studi pustaka dengan penggunaan bahasa PHP dengan <i>Laravel</i> .	<i>Website</i> ini dirancang untuk memberikan kemudahan kepada pengguna dalam memesan produk dan mendapatkan informasi dari kedai Twins. Selain itu, situs ini juga membantu administrator dalam mengelola data yang berkaitan dengan kedai Twins.
3.	Judul: Pembuatan Website Arsyla Catering	Arsyla Catering	Model pengembangan sistem pada penelitian ini adalah	Diterapkannya <i>website</i> arsyala catering dengan menggunakan bahasa pemrograman <i>php</i> dan <i>mysql</i> dengan pengujian

	Tahun: 2022		menggunakan metode SDLC (System Development Life Cycle)	menggunakan metode black box. Tampilan semua fitur dapat berjalan dengan baik.
4.	Judul: Rancang Bangun Aplikasi Pemesanan Pada Cantika Catering Berbasis Web Tahun: 2019	Cantika Catering	Model pengembangan sistem menggunakan model pengembangan <i>waterfall</i> .	Hasil dari sistem yang dibuat: 1. Aplikasi pemesanan catering menjangkau pangsa pasar yang lebih luas. 2. Kegiatan proses bisnis lebih efektif dan efisien. 3. Penyimpanan data lebih rapih dalam bentuk database.
5.	Judul: Rancang Bangun Sistem Informasi Pemesanan Makanan Berbasis Web Pada Cafe Surabiku Tahun: 2019	Café Surabiku	Model pengembangan sistem menggunakan model pengembangan <i>waterfall</i> . Dengan pengujian sistem black box.	Dengan adanya rancang bangun aplikasi pemesanan berbasis web ini dapat mempermudah antara pihak pembeli dan pihak petugas. Sistem ini mempunyai jenis pelayanan diantaranya pemesanan makanan dan minuman melalui Web yang sudah disediakan. Hal ini akan membantu efektifitas waktu pekerja menjadi lebih cepat dan efisien. Serta dapat menghasilkan informasi yang cepat, tepat dan akurat karena menggunakan sistem database Mysql, sehingga data dapat terhubung satu dengan yang lain.
6.	Judul: Perancangan Sistem Aplikasi Pemesanan Makanan Dan Minuman Pada Cafeteria No Caffe Di Tanjung Balai	Tanjung Balai Karimun	Dalam penelitian ini menggunakan pendekatan metode <i>Prototype</i>	Perancangan sistem aplikasi pemesanan makanan dan minuman pada cafeteria no coffe sangat mempermudah kinerja karyawan karena tidak perlu lagi mengolah data manual dan

	<p>Karimun Menggunakan Bahasa Pemrograman Php Dan Mysql</p> <p>Tahun: 2020</p>			<p>memudahkan pelanggan mendapatkan informasi yang efektif tentang cafeteria no coffe.</p>
7.	<p>Judul: Perancangan Aplikasi Pemesanan Makanan Dan Minuman Pada Rumah Makan Cepat Saji D'besto</p> <p>Tahun: 2020</p>	<p>Rumah Makan Cepat Saji D'besto</p>	<p>Metode yang digunakan dalam penelitian ini adalah <i>Waterfall</i></p>	<p>Perancangan aplikasi pemesanan makanan dan minuman pada rumah makan cepat saji d'bestO, sehingga bisa lebih memudahkan kasir dalam proses pemesanan, mengurangi kekeliruan pencatatan, pendataan dan membantu menyajikan dalam pembuatan laporan sehingga lebih mudah dan cepat, karena data didalam database telah terintegrasi, dan perancangan aplikasi pemesanan dapat diterapkan pada rumah makan cepat saji d'bestO.</p>
8.	<p>Judul: Aplikasi Pemesanan Makanan pada Restoran 1953 Indonesia Berbasis Web</p> <p>Tahun: 2019</p>	<p>Kebayoran, Jakarta Selatan</p>	<p>Metode yang digunakan dalam metode ini adalah <i>Waterfall</i>.</p>	<p>Penggunaan aplikasi dapat membantu Restoran 1953 Indonesia dalam hal mempromosikan dan menangani pesanan pelanggan melalui fitur reservasi menu dan reservasi tempat. Kedua fitur ini menyediakan informasi yang lengkap mengenai detail menu dan harga serta tempat dan spesifikasinya. Proses menghitung total pembayaran jadi lebih mudah karena prosesnya dilakukan melalui aplikasi dengan mengkalkulasi seluruh pesanan pelanggan.</p>
9.	<p>Judul: Analisa Pemilihan Aplikasi Pemesanan</p>	<p>Universitas Nusa Mandiri, Jakarta</p>	<p>Metode yang digunakan dalam penelitian in</p>	<p>Anomali harga pangan ini lebih disebabkan factor politik-ekonomi</p>

<p>Makanan Online Menggunakan Metode Analytical Hierarchy Process (Ahp)</p> <p>Tahun: 2022</p>		<p>adalah Metode Analytical Hierarchy Process (Ahp)</p>	<p>pangan nasional yang cenderung berpatron pada mekanisme pasar. Signifikansi dan relevansinya mendorong bentuk pasar pangan menjadi cenderung oligopoly bahkan monopoli. Bentuk pasar seperti ini menjadi penyebab utama panjangnya mata rantai distribusi, mahal biaya logistik serta hubungan konsumen dan produsen yang sangat random. Produsen dan konsumen mudah didikte pemilik modal (pelaku usaha besar). Dalam artian, kendali pasokan dan arus utama distribusi bahan pangan berputar pada pusaran pemodal besar, tengkulak, permainan spekulasi dan kartel serta jaringan importer dan perdagangan yang sebagian besarnya dikuasai mafia.</p>
<p>10. Judul: Perancangan Sistem Pemesanan Pada Web E-Commerce Dengan Metode Agile Dan Performance Testing</p> <p>Tahun: 2023</p>	<p>Katering Dapoer Bu Yunus</p>	<p>Metode yang digunakan dalam penelitian ini adalah metode pengembangan Agile</p>	<p>Membangun sistem informasi pemesanan dengan metode pengembangan Agile dengan kerangka kerja model Kanban, menjadikan proses perancangan terstruktur dan efisien. Menghemat waktu serta biaya saat proses membangun sistem guna meningkatkan nilai terhadap pengguna.</p>

2.2. Sistem Informasi

Sistem informasi adalah elemen-elemen yang saling berkaitan dan kerja bersamaan sesuai rangkaian prosedur untuk mengumpulkan, mengolah, menyimpan dan menampilkan informasi sebagai bahan pendukung pengambilan keputusan (Manuhutu & Wattimena, 2019).

2.3. Sistem

Sistem adalah jaringan kerja yang terorganisir yang bekerja sama, berinteraksi dan terintegrasi dengan jaringan lainnya. Ini terdiri dari beberapa subsistem saling bekerja sama untuk mencapai tujuan yang diinginkan (Jimi Asmara, 2019). Jaringan-jaringan dapat juga disebut sebagai komponen sistem informasi yang terdiri *atas hardware, software, brainware*, prosedur, database dan jaringan komunikasi (Wayan Gede Endra Bratha, 2022).

2.4. Web

Web atau situs web adalah kumpulan halaman yang menampilkan berbagai jenis informasi, termasuk teks, gambar, video, suara, animasi, dan data, dan biasanya merupakan *serangkaian* bangunan yang saling terkait yang dihubungkan dengan jaringan halaman dan hyperlink (Puji Hastanti & Eka Purnama, 2019).

Ada dua tipe website, tipe yang pertama web statis dan tipe yang kedua adalah web dinamis.

- 1) Situs *web* statis mengacu pada situs web yang dikembangkan menggunakan bahasa pemrograman *HTML* atau *CSS*. Secara umum, situs web statis akan menampilkan bentuk yang sama ketika diakses. Jika administrator situs web memperbarui informasi, tampilan situs web jenis ini biasanya tidak berubah saat diakses. Jika administrator ingin mengubah tampilan situs *web* ini, administrator harus melakukan

perubahan melalui kode program, yang mengharuskan administrator untuk memahami prinsip pemrograman situs *web*.

- 2) *Website* dinamis adalah *website* yang menggunakan *database* dan memiliki kemampuan untuk menyesuaikan *website* sesuai dengan situasi saat pengguna mengakses *website* tersebut. Karena *website* ini disusun menggunakan bahasa pemrograman yang terhubung dengan *database*, maka pengembang dapat menampilkan dan membuat halaman dengan konsep visual dan interaktivitas yang tinggi (Febrin Aulia Batubara, 2019).

2.5. Metodologi Pengembangan Sistem

2.5.1. Metode Agile

Metode *Agile* adalah salah satu metodologi untuk pengembangan perangkat lunak yang didasari pada proses pengerjaan berulang yang terdiri dari tata cara dan solusi yang sudah disetujui. Diperlukan adaptasi yang cepat dari pengembang terhadap perubahan atau pembaruan dalam bentuk apapun. *Agile* memiliki prinsip yaitu bersifat cepat, ringan dan bergerak bebas, dimana diperlukan perubahan dan respon cepat antara tim pengembang dan pengguna agar kualitas perangkat lunak yang dihasilkan menjadi optimal (Nyunando & Nasien, 2020).

Tujuan dari *Agile* yaitu tidak memberi instruksi dan perintah, tapi untuk memberikan inspirasi dan pengaruh agar individu-individu yang berada di dalam organisasi dapat bekerja secara mandiri dan berkolaborasi untuk menjadikan produk yang berkualitas (Firdaus et al., 2019).

Agile mengutamakan atas kebutuhan dan kepuasan pengguna (*user*) perangkat lunak. Pengerjaan berulang dilakukan secara inkremental dengan menggabungkan user kedalam tim untuk berkolaborasi bersama pengembang. Reaksi cepat terhadap perubahan atau pembaruan, rencana jangka pendek yang fleksibel dalam mencari solusi dalam penyelesaian

masalah yang dibutuhkan user dilakukan dengan pertemuan tatap muka. *Agile* memiliki alur pengembangan seperti perencanaan, desain/pemodelan, konstruksi, pengujian perangkat lunak, dan perawatan yang berkelanjutan. Semua aktivitas dan prinsip ini menjadi efektif untuk digunakan dalam pengembangan yang berkelanjutan untuk menghadapi perubahan yang cepat. *Agile* memiliki kerangka kerja populer seperti *scrum*, *kanban*, dan *lean* (Gopalkrishna Waja et al., 2021).

2.5.2. Prinsip – Prinsip Metode Agile

Menurut (Al-Saqqa et al., 2020) *Agile* menetapkan dua belas prinsip panduan untuk pengembangan perangkat lunak, Prinsip ini dikenal sebagai *Agile Manifesto* yang mencakup:

- 1) “Prioritas tertinggi adalah memuaskan pelanggan melalui pengiriman perangkat lunak lebih awal dan berkelanjutan”. Proses ini akan menciptakan kepercayaan dan fleksibilitas antara pelanggan dan tim pengembang, dimana tim pengembang akan mendapatkan umpan balik dari pemahaman pelanggan tentang prosedur produk.
- 2) Proses agile menerima apapun perubahan bentuk saat pengembangan yang bermanfaat sebagai keunggulan kompetitif dari pelanggan.
- 3) Pengiriman perangkat lunak yang memiliki kualitas teruji pada skala waktu yang lebih singkat. Prinsip ini membatasi prinsip pertama, merekomendasikan pengiriman awal dan berkelanjutan untuk memenuhi kebutuhan pelanggan.
- 4) Pebisnis dan Pengembang harus bekerja sama setiap hari selama proyek berlangsung. Integrasi ini bertujuan untuk memberikan umpan balik dan menjawab pertanyaan dari tim pengembangan.
- 5) Menciptakan lingkungan kerja dengan individu yang termotivasi tinggi. Memberi mereka lingkungan dan dukungan yang mereka butuhkan dan percayakan mereka untuk menyelesaikan pekerjaan. Anggota tim adalah faktor paling vital dalam kesuksesan, metode *agile* mempercayai

individu-individu ini untuk membuat keputusan yang tepat dalam melakukan pekerjaan mereka, misalnya, mereka dapat mengubah langkah-langkah proses jika mereka pikir itu akan menjadi hambatan bagi tim mereka.

- 6) Metode penyampaian informasi yang sangat efisien dan efektif di dalam tim pengembang adalah komunikasi tatap muka. Prinsip ini menekankan komunikasi langsung dalam tim akan lebih baik dibandingkan menggunakan rencana tertulis.
- 7) Perangkat lunak yang mampu bekerja dengan baik adalah ukuran utama kemajuan. Memecah produk menjadi potongan-potongan kecil dalam pengiriman awal dan berkelanjutan adalah metode pengukuran kemajuan yang lebih baik yang memberikan dampak yang lebih melalui kode yang berjalan.
- 8) Proses *agile* mendukung pembangunan berkelanjutan dengan adanya sponsor, pengembang, dan pengguna. Pengembangan berkelanjutan berarti bahwa tim harus mempertahankan ritme pengerjaan yang stabil. Dengan pengiriman barang berkualitas tinggi dan dapatkan umpan balik sampai tujuan akhir tercapai.
- 9) Perhatian terus-menerus terhadap keunggulan teknis dan rencana yang baik meningkatkan fleksibilitas. Pengiriman kode dengan kualitas tinggi adalah suatu keharusan yang harus selalu dicapai agar cepat dan tepat, bahkan jika mereka harus memperbaiki kode sebagai respon terhadap setiap perubahan prosedur.
- 10) Memaksimalkan jumlah pekerjaan yang belum selesai dengan sederhana. Tujuannya adalah untuk menghasilkan produk yang mampu menangani perubahan mendadak dan pada saat yang sama memenuhi kebutuhan pelanggan.
- 11) Arsitektur, prosedur, dan rencana terbaik muncul dari *Self Organizing Teams*. Setiap tim *agile* adalah tim mandiri yang memiliki tanggung jawab dan menentukan cara terbaik untuk menanganinya melalui

pengetahuan yang mereka peroleh selama proses pengembangan untuk mencapai struktur yang optimal.

- 12) Dalam mengatur pola kerja untuk lebih efektif, tim pengembang melakukan refleksi dan evaluasi secara berkala.

2.5.3. Kanban

Kanban merupakan pendekatan untuk memvisualisasikan alur kerja (workflow) dan pekerjaan yang harus dilakukan pada tim pengembang, dengan menggunakan teori *antrean* untuk mengontrol serta meningkatkan aliran nilai dengan mengarahkan perhatian pada siklus produksi. Keuntungan menggunakan *kanban* yaitu proses komunikasi tim menjadi lebih interaktif, dimana dalam setiap pekerjaan atau proyek melibatkan pihak klien. Penggunaan *kanban* dalam penerapan sistem kerja dapat memaksimalkan bagi proses bisnis, seperti meningkatkan sistem alur kerja, mempersingkat waktu yang dibutuhkan dalam pekerjaan, meningkatkan nilai bagi klien (dos Santos et al., 2019).

Menurut Anderson dalam (dos Santos et al., 2019) *kanban* memiliki lima prinsip:

- 1) Visualisasikan alur kerja: Papan adalah alat utama yang digunakan untuk memvisualisasikan dan kordinasi sesama kerja sama tim. Dimana setiap kolom menunjukkan urutan aktivitas dan kartu merupakan fitur yang sedang dikerjakan.
- 2) Batasan pekerjaan yang sedang berjalan: ini merupakan cara untuk mengelola dan membatasi jumlah pekerjaan yang sedang berjalan serta pemberian tugas yang baru.
- 3) Pengukuran dan pengelolaan aliran kerja: statistik dan diagram yang berbeda dapat digunakan untuk memantau proses *Kanban* seperti siklus/waktu tunggu, ukuran *antrean*, dan diagram aliran kumulatif.

- 4) Buat kebijakan proses secara eksplisit: kebijakan merupakan bagian penting untuk memastikan alur tercapai. Mereka menetapkan kondisi untuk membuat sistem terus bekerja. Misal cara menetapkan tugas dan aktivitas ke pengembang dan kapan pekerjaan dapat ditarik dari satu status ke status lainnya.
- 5) Gunakan model untuk mengenali peluang peningkatan: tiga model disarankan (i) Teori kendala, (ii) Bagian kumpulan ide dari Lean Thinking yang mengidentifikasi pemborosan biaya ekonomi, dan (iii) beberapa varian yang berfokus pada pemahaman dan pengurangan variabilitas.

Sistem *kanban* diimplementasikan sebagai papan di dinding dengan kolom yang mewakili tahapan proses pengembangan berurutan. Kartu digunakan untuk mendeskripsikan pekerjaan atau tugas, yang dipindahkan melalui kolom pada papan. Penerapan *kanban* dalam pengembangan perangkat lunak memiliki tahapan kolom pengerjaan seperti tahapan spesifikasi atau *backlog* (spesifikasi), *development* (pengembangan), *testing* (pengujian), dan *deploy* (penerapan) (dos Santos et al., 2019).

2.5.4. Tahapan Agile - Kanban

Berikut tahapan penelitian menggunakan metode *Agile* dengan kerangka kerja *Kanban*:

1. *To Do*

Kolom ini merupakan daftar pekerjaan berbentuk *card task* yang harus dilakukan untuk mencapai tujuan proyek. Didalam kategori ini terdapat beberapa poin, diantaranya:

a) *Product Backlog*

Merupakan rincian daftar tugas secara umum apa saja yang harus dikerjakan berdasarkan keahlian dari tim pengembang, seperti *card task* pembuatan *home page* yang terdapat pada *website*.

b) *Requirements*

Merupakan kebutuhan detail suatu produk yang didapat dari pengumpulan data. Dimana data didapatkan dari *user story* yang dimaksud ialah cerita detail dari pengguna perihal berjalannya proses sistem yang dibutuhkan, selanjutnya ada *feature* yang merupakan hasil rumusan cerita pengguna berupa rincian fitur – fitur untuk menunjang pengembangan aplikasi dan ada *Use Case* dimana rancangan proses berjalannya sistem digambarkan dalam bentuk *chart* pada *Unified Modelling Language* yang akan memudahkan tim pengembang dalam mengetahui alur dari berjalannya proses sistem yang akan dibangun.

c) *Design*

Tahapan ini merupakan perancangan dalam membangun sebuah sistem, dimana didalam tim pengembang terdapat divisi sesuai keahliannya seperti *ui/ux* yang akan membuat usul desain *interface* secara kreatif dan imajinatif, ada *front-end* yang akan mengubah hasil desain kedalam tampilan *browser* serta merancang proses berjalannya sistem dalam bentuk *Unified Modelling Language*, selanjutnya *back-end* yang bertanggung jawab atas basis data sebuah sistem serta dirancang dalam bentuk *Entity Relationship Diagram*.

d) *Implementation*

Tahapan ini merupakan kegiatan integrasi atau menghubungkan fitur – fitur yang sudah dibuat dengan basis data sebagai penyimpanan data.

e) *Test*

Tahapan terakhir ini dilakukan pengujian dalam mencari kesalahan pada sistem. Proses ini akan mengevaluasi sistem baik secara manual maupun otomatis.

2. *In Progress*

Kolom ini merupakan sedang dikerjakannya tugas – tugas oleh tim pengembang. Dimana *card task* dari kolom *to do* pada papan board dipindah ke kolom *in progress* atau biasa disebut WIP (*work in progress*).

3. *Done*

Kolom ini merupakan sudah diselesaikannya pekerjaan oleh tim pengembang, tugas – tugas sebelumnya yang berada pada kolom *work in progress* diimplementasikan lalu dilakukan testing jika berhasil maka dipindahkannya *card task* ke kolom *done*. Jika ada pekerjaan yang tidak bisa diselesaikan atau gagal saat dilakukan testing, maka *card task* akan di pindahkan ke kolom *to do* untuk didiskusikan dan mencari solusi permasalahan.

2.6. Perangkat Lunak Pendukung

Perangkat lunak yang mendukung penulis dalam pembuatan sistem pemesanan berbasis website, antara lain:

2.6.1. PHP

PHP (Hypertext Pre-processor) merupakan salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pemroses data pada sebuah server (Mubarak et al., 2019). *PHP* juga sebagai pelengkap *HTML* untuk dibuatnya *website* dinamis dengan adanya pengolahan dan pemrosesan data. *Syntax* yang diberikan sepenuhnya dijalankan pada server sedangkan hasilnya dikirimkan ke *browser*. *PHP* dikenal sebagai bahasa *scripting* yang bersanding dengan tag *HTML*, dieksekusi server dan digunakan untuk membuat halaman serta kerangka dalam perancangan *web*. Juga merupakan sebuah *software* yang *open source* bisa digunakan secara bebas dan bersifat gratis.

2.6.2. HTML

HTML adalah bahasa yang perlu dipahami saat membuat situs *web*. *Hyper Text Markup Language (HTML)* adalah bahasa pemrograman standar untuk membuat halaman *web* yang terdiri dari kode tag tertentu. HTML adalah bahasa standar untuk membuat halaman *web* yang

diterjemahkan oleh *browser web* untuk menampilkan halaman *web* yang terdiri dari berbagai format tampilan, termasuk teks, grafik, tautan animasi, dan audio video (Murni et al., 2019).

2.6.3. Javascript

Javascript merupakan bahasa script *website* yang sangat banyak digunakan untuk memanipulasi script HTML dan CSS pada tampilan *browser*. *Javascript* mampu membuat lebih fungsionalitas pada *website*, contohnya yaitu membuat validasi form, berinteraksi pada *server* dengan membuat *website* lebih dinamis dan interaktif. Dengan memanfaatkan *JavaScript* dapat membuat *tools*, aplikasi dan game pada *web* (Firdaus et al., 2019).

2.6.4. MYSQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.6.5. XAMPP

XAMPP adalah paket perangkat lunak komputer yang dinamai berdasarkan singkatan dari *Apache*, *MySQL* (sebelumnya) / *MariaDB* (sekarang), *PHP*, dan *Perl*. "X" di depannya berasal dari istilah cross-

platform, yang berarti bahwa aplikasi ini dapat berjalan pada empat sistem operasi yang berbeda: *Linux OS*, *Windows OS*, *Mac OS*, dan *Solaris*. Jika dijabarkan xampp mempunyai arti sebagai berikut:

X: *XAMPP* berarti dapat digunakan di berbagai *platform*, termasuk *Windows*, *Linux*, *Mac OS*, dan *Solaris*.

A: *Apache* adalah aplikasi *server web*, yang bertanggung jawab untuk menghasilkan halaman *web* yang benar bagi pengguna berdasarkan kode *PHP* yang ditulis oleh penulis halaman *web*. *Apache* juga mengakses database (seperti *MySQL*) terlebih dahulu untuk mendukung halaman *web* yang dihasilkan, jika perlu, berdasarkan kode *PHP* yang ditulis.

M: *MySQL* adalah aplikasi *server database*. Pengembang menggunakan apa yang disebut *Structured Query Language* (*SQL*), yaitu bahasa terstruktur yang digunakan untuk memproses database dan isinya. Pengguna dapat menggunakan *MySQL* untuk menambah, mengubah, dan menghapus data yang ada di dalam database.

P: *PHP*, bahasa pemrograman lain yang serupa, dll.

P: *Perl*, bahasa pemrograman *Perl*, yang sering digunakan untuk memenuhi berbagai kebutuhan. Bahasa ini sangat fleksibel dan banyak digunakan karena dapat digunakan di banyak sistem operasi.

2.7. UML

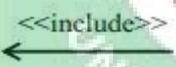
Unified Modelling Language (*UML*) adalah bahasa pemodelan perangkat lunak dan sistem yang menggunakan grafik dan gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis objek berdasarkan konsep pemrograman berorientasi objek.

Didalam *UML* terdapat diagram antara lain:

1) Use Case Diagram

Use Case Diagram menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar.

Tabel 2. 2 Simbol Use Case Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2) Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel data yang dapat memberikan informasi keadaan dimana setiap objek di dalam kelas memiliki nilai yang unik.
- *Operation*/metode atau sering disebut layanan (*service*) atau operasi adalah prosedur atau fungsi yang menjadi perilaku kelas dan objek untuk menjadi tanggung jawab objek tersebut. Dalam bentuk pemrograman merupakan bentuk subprogram yang digunakan terhadap atribut kelas dan objek.

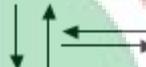
Tabel 2. 3 Simbol Class Diagram

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>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

1) Activity Diagram

Activity Diagram menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem system (Syarif & Nugraha, 2020). Diagram ini hampir sama dengan dengan diagram status karena kegiatannya merupakan status suatu pekerjaan dengan menunjukkan kegiatan yang dilakukan secara berurutan.

Tabel 2. 4 Simbol Activity Diagram

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

2.8. Data Base

Menurut (Andry Andaru, 2019) istilah basis data berasal dari ilmu komputer. Namun, maknanya semakin lama semakin meluas. Ide dasar dari database adalah sekumpulan catatan, sebuah pengetahuan. Basis data memiliki deskripsi terstruktur tentang jenis-jenis fakta yang disimpannya, yang disebut skema. Skema menggambarkan objek-objek yang direpresentasikan oleh *database* dan hubungan di antara mereka.

Data adalah sumber daya yang sangat berharga bagi hampir semua organisasi. Dengan banyaknya data yang tersedia, masalah pengorganisasian data secara efektif menjadi sangat penting dalam pengembangan sistem informasi manajemen.

Suatu bangunan basis data memiliki jenjang sebagai berikut:

- 1) Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau krakter khusus yang membentuk sebuah item data atau *field*.

- 2) *Field/item*, merupakan representasi suatu atribut dan *record* (rekaman) yang sejenis yang menunjukkan suatu item dari data.
- 3) *Record/rekaman*, kumpulan dari *field* membentuk suatu *record* atau rekaman. *Record* menggambarkan suatu unit data individu yang tertentu.
- 4) *File*, merupakan kumpulan dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Mewakili tiap-tiap data.
- 5) *Database*, merupakan kumpulan dari file atau tabel yang membentuk *database*.

2.9. Pengujian Sistem

Pengujian perangkat lunak memiliki peranan yang krusial dalam mendeteksi potensi kesalahan pada suatu perangkat lunak. Proses ini melibatkan evaluasi perangkat lunak baik melalui pendekatan manual maupun otomatis. Dalam kerangka penelitian ini, penulis menerapkan metode load testing, sebagai salah satu cabang dari teknik performance testing, dengan memanfaatkan alat uji perangkat lunak otomatis bernama *Apache JMeter*. *Apache JMeter*, yang merupakan perangkat lunak open-source berbasis bahasa pemrograman *Java* yang dikembangkan oleh *Apache Software Foundation*, memiliki peran penting dalam menjalankan pengujian perilaku fungsional serta pengukuran kinerja perangkat lunak (Suwarsono et al., 2022).

2.9.1. Performance Test

Performance testing, atau pengujian performa, adalah jenis pengujian yang bertujuan untuk mengevaluasi beberapa aspek krusial dari suatu sistem. Ini meliputi kecepatan (waktu respons), skalabilitas (kapasitas pengguna maksimal), stabilitas (respons stabil pada berbagai beban), dan keandalan (kemampuan menangani beban yang berat). Tujuan utama dari pengujian performa ini adalah bukan semata-mata untuk menemukan masalah teknis atau

bug, melainkan untuk memverifikasi kemampuan sebenarnya dari sistem. Tujuan akhirnya adalah untuk meningkatkan kinerja sistem, baik dengan memastikan akses yang lancar atau mengurangi waktu pemuatan (*loading*) yang berlarut-larut (Desy Intan Permatasari et al., 2020).

2.9.2. Load Testing

Load Testing adalah salah satu varian pengujian performa yang menguji sistem dengan beban yang telah ditetapkan untuk mengevaluasi stabilitas dan keandalannya (Suprpto & Sasongko, 2021). Dengan kata lain, pengujian ini bertujuan mengukur performa sistem saat diakses oleh banyak pengguna secara bersamaan, tanpa memerlukan partisipasi pengguna sebenarnya sebagai subjek pengujian.

