

## **BAB II**

### **LANDASAN TEORI**

#### **2.1.1. Konsep Dasar Sistem Informasi**

Konsep dasar sistem merupakan sekelompok komponen berbasis komputer yang dibuat oleh manusia dalam mengelola data, menyimpan, menghimpun kerangka kerja serta mengkoordinasikan sumber daya manusia dan komputer untuk mengubah sistem masukan menjadi sistem keluaran untuk mencapai tujuan dan sasaran yang telah ditetapkan sebelumnya. Kemudian informasi adalah data yang diolah sehingga memiliki arti atau makna yang bermanfaat bagi penerimanya (Arafah & Rusmanto, 2024).

#### **2.1.2. Pengertian Sistem**

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran jika dalam sebuah sistem terdapat sebuah elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Kadek Ambali Yasa et al., 2023).

#### **2.1.3. Pengertian Informasi**

Informasi adalah data yang sudah diolah sesuai dengan keperluan tertentu yang memiliki nilai dan manfaat. Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Pengertian informasi sering disamakan dengan pengertian data (Colarika & Zahro, 2023).

#### 2.1.4. Pengertian Sistem Informasi

Sistem informasi merupakan unit jaringan yang bersama-sama melakukan aktivitas tertentu untuk mencapai suatu tujuan. Membahas mengenai sistem informasi tidak terlepas dari *database*. *Database* adalah komponen penting dari sistem informasi dan merupakan dasar untuk menyediakan dikirim melalui Internet dapat diakses di seluruh dunia hanya dalam menit dan detik. *Website* akan mengubah kebiasaan manusia dari yang biasanya membaca buku, koran, majalah secara fisik sekarang berpindah menggunakan *website* (Rahman et al., 2023). Sistem Informasi menjadi salah satu bentuk perkembangan teknologi yang membantu meningkatkan efisiensi kerja. Sistem informasi digunakan sebagai alat untuk mempermudah penggunaan dan pemrosesan data dalam suatu organisasi, institusi, atau badan usaha. Penggunaan sistem informasi memungkinkan pengambilan keputusan yang lebih obyektif (Indah Melyani et al., 2023).

#### 2.1.5. Karakteristik Informasi

Karakteristik informasi diperlukan untuk mendapatkan jenis informasi yang aktual serta dapat meningkatkan kinerja manajerial dalam mencapai tujuan informasi yang baik (Lathiva Gusna Diva & Elsa Meirina, 2023)

Supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu:

1. Komponen atau elemen (*Components*)

Suatu sistem terdiri dari komponen-komponen yang saling yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

2. Batasan Sistem (*Boundary*)

Batasan sistem merupakan daerah yang membatasi antara sistem satu dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah segala sesuatu di luar batas dari sistem yang mempengaruhi operasi suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. lingkungan luar sistem bersifat menguntungkan harus dipelihara dan dijaga supaya tidak hilang pengaruhnya. Sedangkan, lingkungan yang merugikan harus dihilangkan supaya tidak mengganggu operasi dari sistem.

4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media perantara antar subsistem satu dengan subsistem lain. Melalui penghubung ini memungkinkan sumber - sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran (*output*) dari satu subsistem akan menjadi *Input* untuk subsistem yang lainnya dengan melalui penghubung.

5. Masukan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance Input*) dan masukan signal (*signal Input*).

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolahan Sistem (*Process*)

Suatu sistem mempunyai bagian pengolah yang akan mengubah *Input* menjadi output.

#### 8. Sasaran (*Objective*)

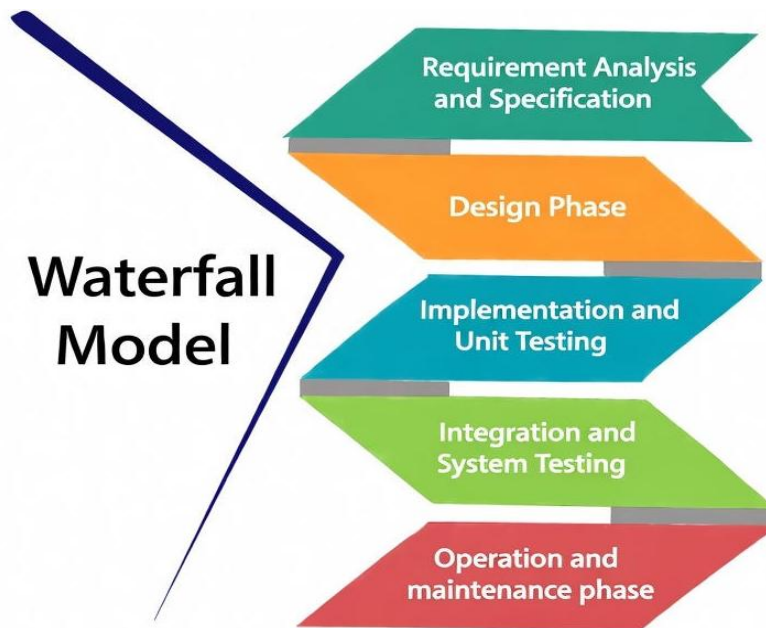
Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

## 2.2. Perancangan

Perancangan adalah proses pasca-analisis dari siklus pengembangan sistem yang dapat berupa perancangan, perencanaan dan pembuatan sketsa atau pengorganisasian beberapa elemen terpisah menjadi satu kesatuan utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras suatu sistem (Ardi Mandani, 2023).

### 2.2.1. Waterfall

*Waterfall* adalah metode yang digunakan oleh para teknisi Rekayasa Perangkat Lunak. Metode ini tentu saja berfokus pada pendekatan sekuental dan terorganisir dengan baik seperti air terjun yang dimulai dari atas menuju bawah yaitu dari *requirement, design, implementation, verification, maintenance*. Metode Waterfall adalah salah satu model pengembangan perangkat lunak yang berfokus pada pendekatan bertahap dan sekuensial. Model ini sering disebut sebagai model tradisional karena telah digunakan secara luas sejak awal pengembangan perangkat lunak (Albert & Ermatita, 2023).



**Gambar 2.1 Metode Waterfall**

**Sumber:**(Topperworld, n.d.).

### **2.3. Pemesanan Menu**

Pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik. Menurut Kamus Besar Bahasa Indonesia yang dimaksud pemesanan adalah “proses, perbuatan, dan cara memesan (tempat, barang, dsb) kepada orang lain”. Pemesanan adalah keseluruhan proses kegiatan yang berkaitan dengan pengelolaan inventory atau persediaan tempat pendistribusian produk dan catatan keseluruhan transaksi pemesanan tempat(Reny Wahyuning Astuti, Irma Suana, 2025).

Sistem pemesanan menu makanan dan minuman ini dibuat untuk mempermudah proses pemesanan menu oleh pelanggan maupun mempermudah

proses pencatatan pesanan. Adanya *website* pemesanan menu ini diharapkan membantu pelanggan mudah dalam memesan dan membayar makanan tanpa harus datang ke tempat. Selain itu, sistem ini memudahkan karyawan untuk melihat data pemasukan (Muhammad Fadri Adam et al., 2023).

#### **2.4. Cafe**

Cafe adalah suatu usaha di bidang makanan dan minuman yang dikelola secara komersial yang menawarkan pada para tamu makanan atau minuman kecil dengan pelayanan dalam suasana tidak formal tanpa diikuti suatu aturan atau pelayanan yang baku sebagaimana sebuah *exclusive dining room*. Cafe merupakan hal yang menjadi pertimbangan konsumen dalam kunjungan, maka diharapkan pihak manajemen cafe menyediakan menu yang variatif dan tidak monoton agar konsumen tidak bosan dan merasa bahwa cafe ini tidak memiliki keistimewaan dibanding cafe sejenis (Grace Amalia Tondang et al., 2023).

#### **2.5. Unified Modelling Language (UML)**

UML mempunyai kelebihan, diantaranya dapat memberikan bahasa pemodelan visual atau gambar bagi *developer* dari berbagai macam pemrograman maupun proses umum rekayasa, menyatukan informasi-informasi terbaik yang ada dalam pemodelan, memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem, dapat memodelkan sistem berorientasi objek, mempermudah *developer* untuk membaca suatu sistem, serta berguna sebagai blueprint yang nantinya dapat

menjelaskan informasi yang lebih detail dalam perancangan berupa *Coding* suatu program(Hidayati et al., 2023).

*Unified Modeling Language* atau yang biasa dikenal dengan singkatan UML adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak. Ini merupakan metode dalam pengembangan sistem berbasis objek yang juga berfungsi sebagai alat bantu dalam proses pengembangan sistem. Beberapa alat bantu yang digunakan dalam perancangan berbasis objek dengan menggunakan UML meliputi diagram use case, *activity* diagram, sequence diagram, dan *class* diagram(Elis & Voutama, 2023).

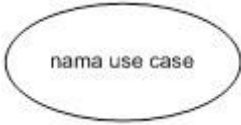
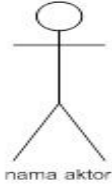
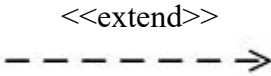
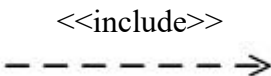
### 2.5.1. Use Case Diagram

*Use case* diagram merupakan pemodelan perilaku (behavior) dari suatu sistem yang dibutuhkan serta diharapkan pengguna. *Use case* digunakan untuk mengetahui fungsi apasaja yang ada di dalam sistem dan siapa saja atau aktor-aktor yang berhak menggunakan fungsi-fungsi tersebut(Suharni et al., 2023). *Use Case* diagram adalah kegiatan-kegiatan atau interaksi yang terjadi di dalam sistem antara aktor dengan sistem(M. I. T. Maulana et al., 2023).

Berikut adalah simbol-simbol yang ada pada *use case* diagram dapat dilihat pada tabel berikut ini :

***Tabel 2.1 Simbol Use Case Diagram***

| Simbol          | Deskripsi                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------|
| <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya |

|                                                                                                                               |                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                              | dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>                                                                                                                                                                                                                             |
| <p>Aktor/<i>actor</i></p>                    | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya menggunakan kata benda di awal frase nama aktor. |
| <p>Asosiasi/<i>assosiation</i></p>           | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor                                                                                                                                                               |
| <p>Ekstensi/<i>extend</i></p>              | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu                                                                                                                                        |
| <p>Generalisasi/<i>generalization</i></p>  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya                                                                                                                                                 |
| <p>Menggunakan/<i>include</i></p>          | <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan                                                                                                                                                                                  |

Sumber:(Ramdany, 2024)

### 2.5.2. Simbol *Activity* Diagram

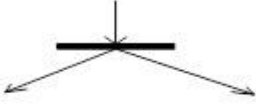
*Activity* diagram menggambarkan aliran fungsionalisme dalam suatu sistem informasi. Secara lengkap, *activity* diagram mendefinisikan dimana workflow dimulai, dimana berhentinya, aktivitas apa yang terjadi selama

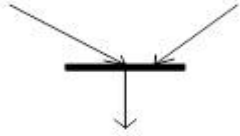
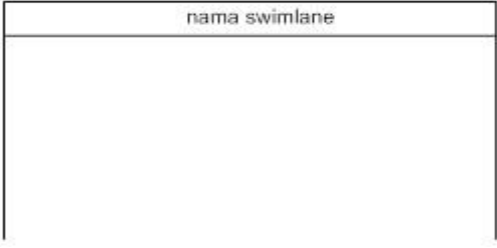


workflow, dan bagaimana urutan kejadian aktivitas tersebut. *Activity* diagram juga menyediakan pendekatan untuk proses pemodelan paralel. Bagi mereka yang akrab dengan analisis dan desain struktur tradisional, diagram ini menggabungkan ide-ide yang mendasari diagram alir data dan diagram alur sistem (Ramdany, 2024).

Berikut adalah simbol-simbol yang ada pada *activity* diagram dapat dilihat pada tabel berikut ini:

**Tabel 2.2 Simbol Activity Diagram**

| Simbol                                                                                                      | Deskripsi                                                                     |
|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <i>Initial State</i><br> | Status awal aktivitas sistem                                                  |
| <i>Activity</i><br>      | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| <i>Control Flow</i><br>  | Urutan perpindahan suatu aktivitas                                            |
| <i>Decision</i><br>      | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu        |
| <i>Final State</i><br>   | Status akhir yang dilakukan sistem                                            |
| <i>Fork</i><br>          | Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel            |
| <i>Join</i>                                                                                                 | Digunakan untuk menunjukkan kegiatan yang digabungkan                         |

|                                                                                                          |                                                                                            |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
|                         |                                                                                            |
| <p><i>Swimlane</i></p>  | <p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p> |

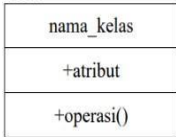
**Sumber:**(Ramdany, 2024)

### 2.5.3. Simbol *Class* Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Diagram ini menjelaskan bagaimana hubungan antara *class* pada aplikasi tersebut terjadi. Yang terdiri dari nama *class*, *attribute*, dan *operation*. Definisi dalam *class* diagram adalah hubungan antar *class* di dalam sebuah sistem(M. I. T. Maulana et al., 2023).

Berikut adalah simbol-simbol yang ada pada *class* diagram dapat dilihat pada tabel berikut ini:

***Tabel 2.3 Simbol Class Diagram***

| Simbol                                                                              | Nama  | Deskripsi                  |
|-------------------------------------------------------------------------------------|-------|----------------------------|
|  | Kelas | Kelas pada struktur sistem |

|                                                                                     |                                              |                                                                                                                                    |
|-------------------------------------------------------------------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|    | Antar muka/<br>interface                     | Sama dengan konsep interface dalam pemrograman berorientasi objek                                                                  |
|    | Asosiasi/<br>association                     | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity                                           |
|    | Asosiasi berarah<br>/directed<br>association | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity |
|    | Generalisasi                                 | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)                                                             |
|  | Kebergantungan /<br>dependency               | Relasi antarkelas dengan makna kebergantungan antarkelas                                                                           |
|  | Agregasi<br>aggregation                      | Relasi antar kelas dengan makna semua – bagian (whole-part)                                                                        |

**Sumber:**(Ramdany, 2024)

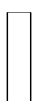
#### 2.5.4. Simbol Sequence Diagram

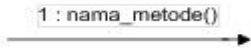
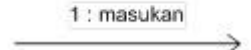
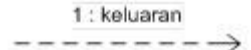
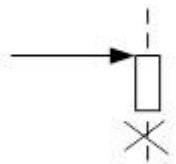
*Sequence diagram* merupakan bentuk kedua dari behavioral model, menunjukkan suatu kejadian (*event*) menyebabkan transisi dari objek ke objek(Suheri et al., 2023). *Sequenc diagrame* adalah diagram UML paling umum kedua yang merepresentasikan bagaimana objek berinteraksi dan bertukar pesan dari waktu ke waktu (*over time*). *Sequence diagram* menggambarkan bagaimana peristiwa (aktivitas) dalam *use case* yang dipetakan ke dalam operasi kelas objek

dalam *class diagram*. Penerimaan umum *Sequence diagram* dapat dikaitkan dengan sifatnya yang relatif intuitif dan kemampuan untuk menggambarkan perilaku parsial(Wayahdi & Ruziq, 2023).

Berikut adalah simbol-simbol yang ada pada *Sequence diagram* dapat dilihat pada tabel berikut ini:

***Tabel 2.4 Simbol Sequence Diagram***

| Simbol                                                                                                                                                                                          | Deskripsi                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Aktor</p>  <p>Atau</p>  | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p> |
| <p>Garis Hidup/<i>Lifeline</i></p>                                                                           | <p>Menyatakan kehidupan suatu objek</p>                                                                                                                                                                                                                                                                                  |
| <p>Objek</p>                                                                                                 | <p>Menyatakan objek yang berinteraksi pesan</p>                                                                                                                                                                                                                                                                          |
| <p>Waktu aktif</p>                                                                                           | <p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya</p>                                                                                                                                                                |
| <p>Pesan Tipe <i>Create</i></p>                                                                              | <p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>                                                                                                                                                                                                                        |

|                                                                                                                 |                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pesan Tipe <i>Call</i><br>     | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri                                                                              |
| Pesan Tipe <i>Send</i><br>     | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim                                              |
| Pesan Tipe <i>Return</i><br>   | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima |
| Pesan Tipe <i>Destroy</i><br> | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>            |

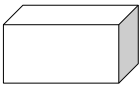
Sumber:(Ramdany, 2024)

#### 2.5.5. Simbol *Component Diagram*

*Component Diagram* memberikan gambaran arsitektur sistem, yang menunjukkan komponen utama dan ketergantungannya. Sistem ini terdiri dari beberapa komponen utama, termasuk modul manajemen produk, modul inventori, dan modul laporan. Setiap komponen terhubung dalam arsitektur yang memastikan efisiensi dan kemudahan dalam pemeliharaan sistem(Hendrawan et al., 2024).

Berikut adalah simbol-simbol yang ada pada *Component diagram* dapat dilihat pada tabel berikut ini:

***Tabel 2.5 Simbol Component Diagram***

| NO | SIMBOL                                                                            | NAMA               | KETERANGAN                                                                                         |
|----|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------|
| 1. |  | <i>Node</i>        | Sumber daya yang digunakan pada saat aplikasi dijalankan.                                          |
| 2. |  | <i>Association</i> | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 3. |  | <i>Component</i>   | Menjelaskan perangkat keras atau objek di dalam sistem.                                            |

**Sumber:**(Ramdany, 2024)

## 2.6. HTML

HTML merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. Halaman ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan pada web browser. Kode HTML

“<HTML>

</HTML>

Masing-masing baris di atas disebut tag. Tag adalah kode yang digunakan untuk *me-mark-up* (memoles) teks ASCII menjadi file HTML. Setiap teks diapik dengan tanda kurung runcing. Ada tag pembuka yaitu <HTML> dan ada tag penutup yaitu </HTML> yang ditandai dengan tanda slash (garis miring) di depan awal penulisan. Tag di atas memberikan kaidah bahwa yang akan ditulis

diantara kedua tag tersebut adalah isi dari dokumen HTML(Pratama et al., 2023).

## **2.7. CSS**

*Cascading Style Sheet* (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda(Nababan et al., 2023).

## **2.8. PHP**

PHP adalah bahasa pemrograman sisi-server (*server-side scripting language*) yang dirancang untuk pengembangan aplikasi web dinamis. Kode PHP dijalankan pada server, kemudian outputnya dikirim ke browser klien sebagai HTML. PHP dapat disematkan dalam markup HTML atau digunakan bersama framework dan sistem manajemen basis data seperti MySQL untuk membuat

sistem informasi berbasis web. Keunggulan PHP tidak hanya terletak pada kemudahan penggunaan, tetapi juga pada fleksibilitas dalam menyesuaikan kebutuhan sistem yang beragam. Menurut statistik global, PHP digunakan oleh lebih dari 77% website yang diketahui teknologi server-nya, menjadikan PHP sebagai bahasa pemrograman web terpopuler di dunia saat ini, PHP dapat berjalan pada berbagai platform termasuk Linux, Unix, Windows, dan macOS serta terintegrasi secara langsung dengan HTML (Commons & Rules, 2025). PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan (Eky Prasetya Hartono, 2023).

## **2.9. Database**

*Database* adalah kumpulan data yang terhubung dan disimpan secara bersama pada suatu media dengan cara-cara tertentu sehingga mudah untuk digunakan dan ditampilkan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan sedemikian sehingga penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (Aryani et al., 2023).

### **2.9.1. MySQL**

MySQL adalah sistem manajemen basis data relasional (Relational Database Management System, RDBMS) bersifat open-source. MySQL memungkinkan pengguna untuk menyimpan, mengelola, dan mengambil data menggunakan bahasa kueri SQL (Structured Query Language). MySQL adalah



sistem manajemen basis data yang handal dan populer untuk menyimpan dan mengelola data(I. Maulana, 2023).

### **2.9.2. XAMPP**

XAMPP adalah sebuah software web server apache yang didalamnya sudah tersedia database server MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa modul lainnya. Dengan XAMPP, pengguna dapat membuat lingkungan server lokal (`localhost`) untuk mengembangkan dan menguji website atau aplikasi web berbasis PHP, tanpa perlu koneksi internet atau instalasi terpisah untuk setiap komponen. XAMPP adalah paket perangkat lunak sumber terbuka (open source), cross-platform (mendukung berbagai sistem operasi seperti Windows, Linux, macOS), yang terdiri dari kumpulan program penting untuk menjalankan server web secara lokal: web server Apache HTTP Server, sistem basis data (sebelumnya MySQL, sekarang umumnya MariaDB), dan interpreter untuk bahasa pemrograman server-side seperti PHP dan Perl(Pratama et al., 2023).

XAMPP adalah paket perangkat lunak open source yang menyediakan lingkungan server lokal untuk pengembangan web. XAMPP mencakup Apache (server web), MariaDB atau MySQL (basis data), serta interpreter untuk skrip PHP dan Perl. Dengan XAMPP, pengembang dapat menguji dan mengembangkan

aplikasi web secara lokal sebelum diunggah ke server produksi(Ramadhan et al., 2025).



**Gambar 2.2 XAMPP**

**Sumber:** (Rina Noviana, 2022).

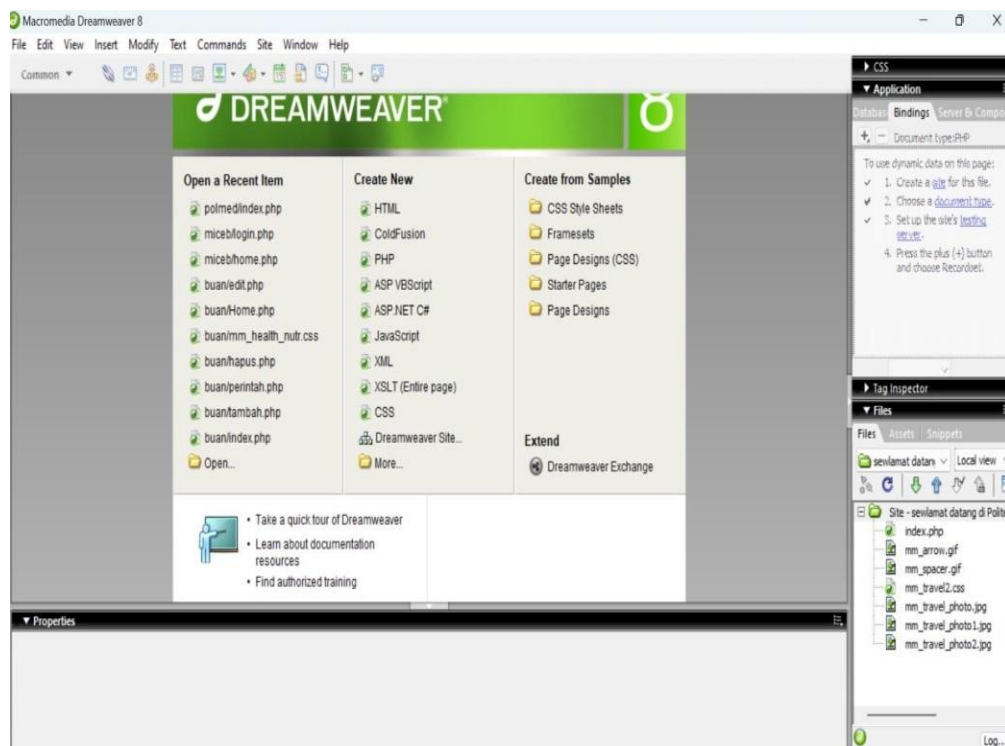
## **2.10. Tools yang Digunakan**

### **2.10.1. Marcomedia DreamWeaver**

Macromedia Dreamweaver memberikan manfaat besar bagi individu maupun organisasi dalam pengembangan *website*. Dengan fitur-fitur unggulan dan antarmuka yang ramah pengguna, Dreamweaver memfasilitasi proses pembuatan dan pengelolaan *website* yang profesional dan responsif. Meskipun memiliki tantangan terkait dengan biaya lisensi dan ketergantungan pada antarmuka grafis, Dreamweaver tetap menjadi pilihan yang baik dalam pelatihan pembuatan *website*, baik untuk pemula maupun pengembang berpengalaman.

Dreamweaver memiliki berbagai keunggulan yang menjadikannya pilihan populer bagi para pengembang web, seperti: Antarmuka Visual yang Mudah Dipahami: Pengguna dapat langsung melihat perubahan yang dilakukan pada

*website* tanpa harus memuat ulang halaman secara terpisah. Dukungan terhadap Berbagai Teknologi Web: Dreamweaver mendukung HTML, CSS, JavaScript, PHP, dan *MySQL* , memungkinkan pembuatan *website* yang lebih dinamis dan interaktif. Kemudahan Integrasi dengan Server: Dreamweaver memungkinkan pengelolaan file di server dengan mudah, sehingga mempermudah proses upload dan update *website*(Nababan & Sinambela, 2025).



**Gambar 2.3 Macromedia Dreamweaver 8**

**Sumber:** (Nababan & Sinambela, 2025)

### 2.10.2. Visual Studio Code

Visual Studio Code merupakan sebuah code editor yang berfungsi sebagai perangkat lunak pendukung dalam pembuatan aplikasi. Code editor ini dirancang untuk membantu pengembang dalam menulis, mengedit, dan mengelola source code dengan lebih efisien. Visual Studio Code dapat dijalankan di berbagai perangkat desktop, termasuk sistem operasi Windows, macOS, dan Linux, sehingga memberikan fleksibilitas tinggi kepada para pengembang. Selain itu, Visual Studio Code juga merupakan code editor yang bersifat open source. Artinya, perangkat lunak ini tersedia secara gratis dan dapat dimodifikasi sesuai kebutuhan pengguna. Keunggulan lainnya adalah dukungannya terhadap berbagai bahasa pemrograman seperti JavaScript, Python, C++, Java, dan masih banyak lagi. Dengan fitur extensions yang luas, pengembang dapat menambahkan alat dan fungsi khusus untuk memenuhi kebutuhan proyek mereka (Halifah & Subrata, 2025).



**Gambar 2.4 Visual Studio Code**

**Sumber:**(Romzi & Kurniawan, 2023).

### 2.11. Penelitian Terdahulu

Perkembangan teknologi informasi mendorong usaha kuliner menerapkan sistem pemesanan menu berbasis web guna meningkatkan kualitas pelayanan. Penelitian terdahulu menunjukkan bahwa teknologi web seperti PHP dan *MySQL* mampu mempercepat pemesanan, meningkatkan akurasi data, dan mempermudah pengelolaan pesanan.

| N<br>o | Peneliti &<br>Tahun                         | Judul<br>Penelitian                                                                  | Metode /<br>Teknolog<br>i                         | Hasil<br>Penelitian                                                                                                                                   | Relevansi<br>dengan<br>Penelitian                                                                                                    |
|--------|---------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Muhamma<br>d Fadri<br>Adam et<br>al. (2023) | Perancangan<br>Sistem<br>Pemesanan<br>Menu<br>Makanan dan<br>Minuman<br>Berbasis Web | Web-<br>based<br>system,<br>PHP &<br><i>MySQL</i> | Sistem<br>mampu<br>meningkatkan<br>efisiensi<br>pemesanan,<br>mempercepat<br>pelayanan,<br>dan<br>meminimalka<br>n kesalahan<br>pencatatan<br>pesanan | Menjadi acuan<br>dalam<br>perancangan<br>sistem<br>pemesanan<br>menu berbasis<br>web untuk<br>meningkatkan<br>pelayanan<br>pelanggan |

|   |                         |                                                      |                                                      |                                                                                          |                                                                                            |
|---|-------------------------|------------------------------------------------------|------------------------------------------------------|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 2 | Astuti et al. (2025)    | Sistem Informasi Pemesanan pada Usaha Kuliner        | Sistem informasi berbasis web, <i>Database MySQL</i> | Sistem membantu pengelolaan data pesanan, stok, dan laporan penjualan secara terstruktur | Relevan dalam penerapan sistem terkomputerisasi pada usaha kuliner seperti Angkringan Nook |
| 3 | Sotar (2023)            | Analisis Sistem Pemesanan terhadap Kepuasan Konsumen | Sistem pemesanan digital                             | Sistem pemesanan yang baik berpengaruh terhadap kepuasan pelanggan dan ketepatan pesanan | Mendukung pentingnya sistem pemesanan berbasis web untuk meningkatkan kepuasan pelanggan   |
| 4 | Wulandari et al. (2024) | Pengembangan Sistem Informasi Menggunakan            | Metode Waterfall                                     | Metode waterfall menghasilkan sistem yang terstruktur                                    | Digunakan sebagai metode pengembangan sistem pada penelitian ini                           |

|   |                                         |                                                                       |                                   |                                                                                                                |                            |
|---|-----------------------------------------|-----------------------------------------------------------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------|----------------------------|
|   |                                         | Metode<br>Waterfall                                                   |                                   | dan sesuai<br>kebutuhan<br>pengguna                                                                            |                            |
| 5 | Penelitian<br>ini di<br>Angkringan NOOK | Perancangan<br>Sistem<br>Informasi<br>dengan<br>Bahasa<br>Pemrograman | <i>Website &amp;<br/>Database</i> | <i>Website</i><br>mampu<br>meningkatkan<br>kecepatan<br>akses data<br>dan efisiensi<br>pengolahan<br>informasi | Dalam Proses<br>Penelitian |