

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara umum sistem dapat didefinisikan sebagai suatu sistem terdiri dari elemen yang bisa berbentuk individu atau bagian-bagian yang terpisah, kemudian berinteraksi satu sama lain untuk mencapai tujuan. Seiring perkembangan ilmu pengetahuan, sistem menjadi bagian terpenting yang harus ada untuk mengatur jalannya sebuah kegiatan.

2.1.1. Program

Program adalah serangkaian instruksi yang ditulis untuk melakukan suatu fungsi spesifik pada komputer. Komputer pada dasarnya membutuhkan keberadaan program agar bisa menjalankan fungsinya sebagai komputer (Hendriyani & Suryani, 2020).

Program adalah kumpulan instruksi yang digunakan untuk mengatur computer agar melakukan suatu tindakan tertentu. Tanpa program, computer sesungguhnya tidak dapat dibuat apa-apa, itu sebabnya, sering dikatakan bahwa computer mencakup tiga aspek penting, berupa perangkat keras (*hardware*), perangkat lunak (*software*) yang dalam hal ini berupa program dan perangkat akal (*brainware*) atau orang yang berperan terhadap operasi computer maupun pengembangan perangkat lunak (Sarkawi, 2018).

2.1.2. Bahasa Pemrograman

Menurut Kadir menjelaskan “Suatu bahasa pemrograman atau kumpulan intruksi berdasarkan kalimat dapat dianalogikan dengan suatu program. Manusia dapat mengerjakan instruksi berdasarkan kalimat-kalimat dan komputer bisa menjalankan

suatu instruksi menurut program.” (Rahayu et al., 2019). Bahasa pemrograman yang digunakan oleh penulis dalam pembuatan aplikasi berbasis website ini, yaitu :

1. PHP (Hypertext Preprocessor)

“PHP yaitu singkatan dari (*Hypertext Preprocessor*) merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamin” (Fadila et al., 2021).

“PHP merupakan bahasa pemrograman yang dapat digunakan untuk membuat skrip yang lebih interaktif. Skrip ini akan diolah dalam *web server* yang hasilnya dapat dilihat dalam bentuk HTML” (Abdulghani & Gozali, 2020).

Dari kedua pengertian diatas dapat disimpulkan bahwa PHP merupakan salah satu bahasa pemrograman *open source* yang dikhususkan buat pengembangan *web* dan dapat ditanamkan dalam sebuah skripsi HTML.

2. JavaScript

“JavaScript merupakan bahasa pemrograman *client-side* yang dapat digunakan untuk menangani antarmuka aplikasi web dan menghasilkan jalamannya web dinamis melalui *Document Object Model*” (Robani et al., 2021).

“JavaScript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk *web*” (Sahi, 2020).

Dari kedua pengertian diatas dapat disimpulkan *JavaScript* merupakan skrip pemrograman sederhana yang biasa digunakan oleh pengembang web untuk membuat interaksi yang lebih dinamis saat membuat situs web, aplikasi, server, atau game.

3. HTML (*Hypertext Markup Language*)

“HTML merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web* yang berfungsi untuk mempublikasi dokumen online” (Sari et al., 2022).

“HTML adalah singkatan dari *Hypertext Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web internet” (Abdulghani & Gozali, 2020).

Dari kedua pengertian diatas dapat disimpulkan bahwa HTML merupakan bahasa markup yang menggunakan simbol (tag) tertentu untuk menyatakan kode yang harus dipahami browser untuk membuat dan menyusun halaman web dan aplikasi.

4. CSS (*Cascading Style Sheet*)

“*Cascading Style Sheet* (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam” (Abdulghani & Gozali, 2020).

Menurut Madcoms “*Cascading Style Sheet (CSS)* adalah kumpulan kode-kode yang berurutan dan saling berhubungan untuk mengatur format atau tampilan suatu halaman HTML” (Suryadi, 2019).

Dari kedua pengertian diatas dapat disimpulkan bahwa *Cascading Style Sheet (CSS)* merupakan *bahasa pemrograman* atau markup untuk membangun sebuah website.

2.1.3. Website

Firmansyah mengemukakan bahwa “*Website* adalah sebuah kumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan serta dapat diakses secara luas melalui halaman depan (*home page*) menggunakan sebuah *browser* menggunakan URL *website*” (Firmansyah, 2020).

“*Website* merupakan sebuah kumpulan halaman-halaman *web* beserta file-file pendukungnya, seperti file gambar, video, dan file digital lainnya yang disimpan pada sebuah *web server* yang umumnya dapat diakses melalui internet” (Wahyudin & Rahayu, 2020).

Dari kedua pengertian diatas dapat disimpulkan bahwa website adalah kumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi digital berupa gambar, teks, suara, musik dan animasi lainnya yang disediakan melalui koneksi internet.

2.1.4. Basis Data

Basis data adalah kumpulan data yang terintegrasi dan terkait secara logis yang dirancang untuk melengkapi informasi yang diperlukan (Siswanto, 2021). Aplikasi basis data yang digunakan penulis pada website yang dibuat, yaitu :

1. MySQL

“MySQL yaitu sebuah perangkat lunak yang digunakan untuk membuat pangkalan data yang dapat mengatur dan mengumpulkan data yang sangat banyak juga bisa dicapai oleh banyak pengguna lain” (Syahputra Novelan et al., 2020).

2. XAMPP

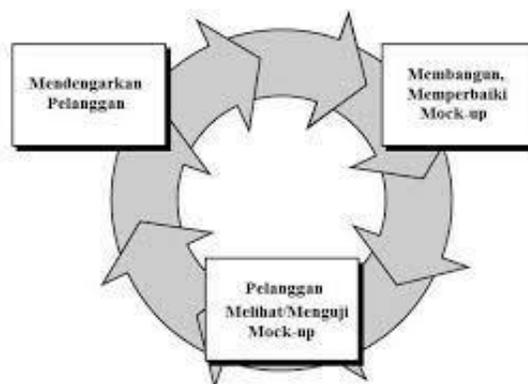
“XAMPP adalah *software web server apache* yang di dalamnya sudah tersedia database mysql dan support php programming. XAMPP merupakan *software* yang mudah digunakan dan gratis dan mendukung instalasi di Linux dan Windows” (Sari et al., 2022).

3. PhpMyAdmin

“PhpMyAdmin adalah sebuah aplikasi pemrograman yang digunakan untuk *managemen database* melalui browser (*web*) untuk mengontrol data mereka dan isi *web* yang akan ditampilkan dalam sebuah *website* yang mereka buat tanpa harus menggunakan perintah (*command*) *SQL*” (Kurniawan et al., 2021).

2.1.5. Metode Pengembangan Perangkat Lunak

Prototyping merupakan proses pengembangan sistem informasi yang bersifat iteratif dan mampu menutupi kekurangan, dengan model dimana *Prototyping Requirement* dengan cepat dikonversi menjadi suatu sistem yang bekerja, sistem direvisi secara terus-menerus, kolaborasi erat antara *user* dan *analyze system*.(Pradipta, 2019).



Sumber: (Pradipta, 2019).

Gambar II 1 Prototyping

1. Mendengarkan Pelanggan

Pada proses ini menjelaskan bahwa metode *prototyping* dimulai dengan mendengarkan kebutuhan dan masukan dari pengguna. Pengembang dan pengguna bertemu dan bersama-sama menentukan tujuan keseluruhan untuk perangkat lunak dan mengidentifikasi apapun persyaratan yang diperlukan.

2. Membangun dan Memperbaiki *Prototype*

Pada proses ini pengembang membuat sebuah gambar tentang aplikasi yang selanjutnya dapat dipresentasikan kepada pemilik toko. Gambaran tersebut berfokus pada representasi aspek-aspek aplikasi yang akan terlihat oleh pengguna.

3. Pengujian *Prototype* oleh Pelanggan

Pada proses ini setelah aplikasi berhasil dibuat sesuai dengan kebutuhan maka dilakukan proses pengujian aplikasi oleh pemilik toko sebelum aplikasi tersebut diimplementasikan.

2.2. Peralatan Pendukung

Pada sub bab ini, penulis menjelaskan teori tools yang akan digunakan dalam perancangan sistem yang dibahas, diantaranya sebagai berikut.

2.2.1. *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) merupakan salah satu bentuk pemodelan basis data yang sering digunakan dalam pengembangan sistem informasi. Diagram hubungan entitas menunjukkan hubungan dari entitas set disimpan dalam *database*

Menurut para ahli, Brady dan Loonam dalam jurnal (Mulyana et al., 2021) *Entity Relationship diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analysis* dalam tahapan analisis persyaratan proyek pengembangan sistem. ★

Adapun komponen-komponen yang teradapat pada ERD yaitu sebagai berikut:

a. Entitas

suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.

b. Atribut

merupakan ciri umum semua atau sebagian besar instansi pada entitas tertentu.

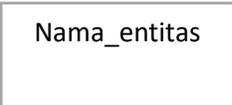
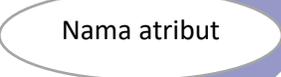
c. Relasi

merupakan hubungan alamiah yang terjadi antara satu atau lebih entitas.

d. Link

meliputi garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Tabel II. 1 Simbol Entity Relationship Diagram

No	Simbol	Keterangan
1	Entitas/ Entity 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
2	Atribut 	Field atau kolom data yang butuh disimpan dalam suatu entitas.
3	Atribut Kunci Premier 	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
4	Relasi 	Relasi yang menghubungkan antar entitas ; biasanya diawali dengan kata kerja .
5	Asosiasi/ Association 	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian , memungkinkan jumlah maksimum keterhubungan antara entitas satu dengan entitas lain disebut dengan kardinalitas .

2.2.2. Logical Record Structure

Logical Relationship Structure (LRS) adalah model permodelan yang mempresentasikan relationship menjadi relasi-relasi atau table-table database (Susliansyah & Frisma, 2019)

LRS merupakan representasi dari pembuatan ERD, dimana setiap entitas dalam ERD akan berubah menjadi kotak atau tabel. LRS ini menggambarkan tabel-tabel yang

digunakan dalam pembuatan aplikasi programnya. Setiap tabel terdiri dari atribut-atribut, yang sudah ditentukan *primary key* maupun *foreign key* nya. (Trisianto, 2018)

Kesimpulan pendapat diatas menjelaskan bahwa dari struktur record-record pada tabel-tabel dibentuk dari hasil relasi antar tabel dan digambarkan kotak persegi panjang dan dengan nama unik yang kelihatan pada kedua link tipe record secara logic yang sesuai dengan konsep.

2.2.3. *Unified Modelling Language*

Unified Modelling Language (UML) adalah sebuah Bahasa yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak (Kurniawan & Syahputra, 2018).

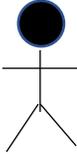
Unified Modeling Language (UML) adalah salah satu standar Bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sari & Istikoma, 2018).

Beberapa digram *Unified Modeling Language* (UML) yang digunakan diantaranya adalah sebagai berikut :

1. *Use Case Diagram*

Tabel II. 2 Simbol Use Case Diagram

GAMBAR	KETERANGAN
--------	------------

	<p>ACTOR</p> <p>Orang proses, atau sistem lain yang berinteraksi dengan system informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
	<p>USE CASE</p> <p>Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pasar antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.</p>
	<p>ASOSIASI/ASSOCIATION</p> <p>Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.</p>
<p><extend></p> 	<p>EKSTENSI/EXTEND</p> <p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan</p>
	<p>GENERALISASI/GENERALIZATION</p> <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p><include></p> 	<p>MENGGUNAKAN/INCLUDE</p> <p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.</p>

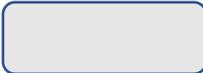
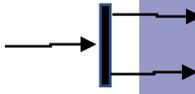
Sumber : (Murdowo & Nugroho, 2018).

Use case merupakan sebuah diagram dalam UML yang menunjukkan hubungan antar actor dalam sistem yang akan dikembangkan, para aktor saling berhubungan dalam proses yang terjadi dalam sistem (Murdowo & Nugroho, 2018).

Use case diagram digunakan untuk mengkomunikasikan interaksi manusia (actor) dengan apa yang bisa dilakukan oleh sistem (Abdillah, 2019).

2. Activity Diagram

Tabel II. 3 Simbol Activity Diagram

GAMBAR	KETERANGAN
	STATUS AWAL/INITIAL Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal.
	AKTIVITAS/ACTIVITY Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	PERCABANGAN/DECISION Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	PENGGABUNGAN/JOIN Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
	STATUS AKHIR/FINAL Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
	SWIMLINE Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (Siswidiyanto, 2020).

Activity diagram adalah diagram yang menggambar *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Siswidiyanto, 2020).

Yang perlu di perhatikan disini adalah bahwa digram aktivitas menggambar aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rodianto, 2020).

3. Class Diagram

Tabel II. 4 Simbol Class Diagram

GAMBAR	NAMA	KETERANGAN
	Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
	Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Sumber : (Sari & Istikoma, 2018).

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek (Sari & Istikoma, 2018).

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun istem (Rodianto, 2020).

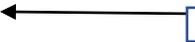
Class Diagram menjelaskan berbagai macam objek dalam sistem dan hubungan statis yang terdapat diantaranya menunjukkan property dan operasi sebuah class serta batasan-batasan dalam hubungan objek.

4. *Sequence Diagram*

Sequence diagram, yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi berdasarkan urutan waktu (Rachman & Marjito, 2020).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian Langkah-langkah yang di lakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu (Sari & Istikoma, 2018). Berikut symbol-symbol sequence diagram.

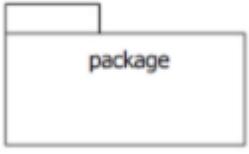
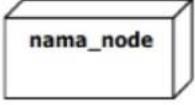
Tabel II. 5 Simbol Sequence Diagram

GAMBAR	NAMA	KETERANGAN
	LifeLine	Objek entity antarmuka yang saling berinteraksi
	Message	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	Message	Spesifikasi dari komunikasi antar objek yang membuat informasi-informasi tentang aktifitas yang terjadi.

Sumber : (Sari & Istikoma, 2018).

5. Diagram Deployment

Tabel II. 6 Diagram Deployment

	<p>Package Package merupakan sebuah bungkus dari suatu atau lebih node.</p>
	<p>Node Menyatakan perangkat keras (Hardware), perangkat lunak yang tidak di buat sendiri (Software) jika di dalam node di sertakan komponen untuk mengkonsistenkan rancangan maka komponen yang di ikut sertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.</p>
	<p>Dependency Keberuntungan antar node, arah panah mengarah pada node yang di pakai.</p>
	<p>Link Relasi antar node.</p>

Sumber: (Ade Hendini, 2018).

2.2.4. Blackbox Testing

Black Box Testing (Penguian kotak hitam) yaitu bertujuan untuk menunjukkan fungsi PL tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan dan apakah informasi yang disimpan secara eksternal selalu dijaga kemuktahirannya (Nurajizah & Aziz, 2019).

Penguian *black-box* berusaha menemukan kesalahan dalam kategori menurut (Suryawan et al., 2019) sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan Interface

3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan Kinerja.
5. Inisialisasi dan kesalahan terminasi

