

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Aplikasi

Aplikasi adalah Program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Widarman dan Kumala 2017).

2.1.2 E-Commerce

E-Commerce merupakan aktivitas pembelian dan penjualan melalui jaringan internet dimana pembeli dan penjual tidak bertemu secara langsung, melainkan berkomunikasi melalui media internet Menurut (Ahmadi, 2020). *E-Commerce* adalah suatu proses membeli dan menjual produk-produk secara elektronik oleh konsumen dan dari perusahaan ke perusahaan dengan komputer sebagai perantara transaksi bisnis. *E-commerce* adalah segala bentuk aktivitas transaksi jual beli barang ataupun jasa dengan menggunakan media elektronik, Sandi, E. K. (2022).

2.1.2.1 Komponen E-Commerce

Ada beberapa kelebihan yang dimiliki *E-commerce* dan tidak dimiliki oleh transaksi bisnis secara *offline* yaitu:

1. Produk : banyak jenis produk yang bisa dipasarkan dan dijual melalui internet seperti pakaian, mobil, sepeda dll.
2. Tempat menjual produk : tempat menjual adalah internet yang harus dimiliki dominan dan hosting.
3. Cara menerima pesanan : *email*, telpon, sms dan lain-lain.
4. Cara pembayaran : *credit card*, paypal, tunai
5. Metode pengiriman : menggunakan pos Indonesia, EMS, atau JNE

6. *Customer service* : email, *contact us*, telepon chat jika tersedia dalam *software*.

2.1.2.3 Jenis-jenis E-Commerce

E-commerce memiliki berbagai transaksi dalam menerapkan sistemnya. Jenis-jenis transaksi *e-commerce* diantaranya sebagai berikut:

1. *Collaborative Commurce (C-commerce)*

Collaborative commurce yaitu kerjasama secara elektronik antara rekan bisnis. Kerja sama ini biasanya terjadi antara rekan bisnis yang berada pada jalur penyediaan barang (*supply chain*).

2. *Business To Business (B2B)*

E-commerce tipe ini meliputi transaksi antar organisasi yang dilakukan di *electronic market*.

3. *Business To Consumers (B2C)**Business to consumers* Yaitu penjual adalah suatu organisasi dan pembeli adalah individu

4. *Consumer To Business (C2B)*

Dalam *Consumer to business* memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen.

5. *Costumer To Customer (C2C)*

Costumer to customer yaitu konsumen menjual secara langsung ke konsumen lain atau mengiklankan jasa pribadi di internet. Dalam *Costumer to customer* menjual produk atau jasa ke *costumer* lain. Dapat juga disebut sebagai pelanggan kepelanggan yaitu orang yang menjual produk dan jasa kesatu sama lain.

2.2.3 UMKM

Usaha mikro, kecil dan menengah (UMKM) menurut No. 20 Tahun 2008 tentang UMKM. Di mana pengertian dari usaha mikro, kecil dan menengah yakni usaha mikro adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan yang memiliki kriteria usaha mikro sebagaimana diatur dalam undang-undang tersebut.

2.2.4 Website

Website (situs web) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. Web adalah sistem *hypertext*, terdiri dari jutaan halaman teks yang dihubungkan oleh *hyperlink-hyperlink* (Sanjaya & Hesinto, 2018).

Website adalah sejumlah halaman *web* yang memiliki topik saling terkait antara halaman yang satu dengan halaman lain, yang biasanya ditempatkan pada sebuah *server web* yang dapat diakses melalui jaringan internet ataupun jaringan wilayah lokal (LAN) (Susilowati, 2019) .

1. Adapun Jenis – jenis *web* berdasarkan sifat atau *style*-nya yaitu :
 - a. *Website* Dinamis, merupakan sebuah *website* yang menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, NET dan memanfaatkan *database* MySQL atau MS SQL.
 - b. *Website* Statis, merupakan *website* yang *content*-nya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*.
2. Fungsinya, *website* terbagi atas :
 - a. ***Personal website***, *website* yang berisi informasi pribadi seorang.
 - b. ***Commercial website***, *website* yang dimiliki oleh sebuah perusahaan yang bersifat bisnis.
 - c. ***Government website***, *website* yang dimiliki oleh instansi pemerintahan. pendidikan yang bertujuan memberikan pelayanan kepada pengguna.
 - d. ***Non-Profit Organization website***, dimiliki oleh organisasi yang bersifat *non-profit* atau tidak bersifat bisnis.
3. Segi Bahasa pemrograman yang digunakan, *website* terbagi atas:
 - a. *Server Side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung kepada tersedianya *server*. Seperti PHP, ASP, NET dan lain sebagainya.

Client Side, adalah *website* yang tidak membutuhkan *server* dalam menjalankannya, cukup diakses melalui browser saja.

2.2.5 XAMPP

Xampp merupakan perangkat lunak berbasis *web server* yang bersifat *open source* (bebas), serta mendukung di berbagai sistem operasi, baik *Windows*, *Linux*, atau *Mac OS*. *Xampp* digunakan sebagai standalone server atau biasa disebut dengan *localhost*. Hal tersebut memudahkan dalam proses pengeditan, desain, dan pengembangan aplikasi. Terdapat banyak manfaat dari penggunaan *Xampp*, berikut merupakan beberapa fungsi utama yang dimiliki oleh *tool web server* ini.

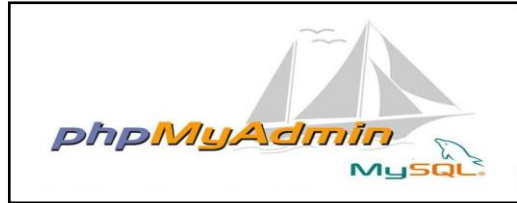
1. Mengkonfigurasi Pengaturan Database pada *PhpMyAdmin* Pertama, mampu mengatur halaman basis data pada *PhpMyAdmin* tanpa perlu khawatir terjadi *error*, dikarenakan anda hanya mengakses pada *server* lokal komputer saja. Dengan *PhpMyAdmin*, anda bebas untuk melakukan beberapa perubahan seperti mengedit, menghapus, mengupdate, dan menambahkan *user* pada *database*.
2. Menjalankan *Laravel* melalui Perangkat Komputer Kedua, *Laravel* merupakan salah satu *framework* milik PHP yang berfungsi untuk mempermudah programmer dalam mengembangkan tampilan *website*. Sehingga, dengan penggunaan *Xampp* akan lebih mudah dalam memodifikasi kode program atau *script*, serta membuat fitur baru dengan lebih cepat. (Noviantoro, 2022).



Gambar 2. 1 *Xampp*

2.2.6 PHP MyAdmin

PhpMyAdmin merupakan aplikasi berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP. Melalui *PhpMyAdmin*, user dapat melakukan perintah *Query* tanpa harus mengetikkan seperti pada MS DOS. Perintah tersebut misalnya administrasi User dan *privileges*, *export* dan *import* database, manajemen *database*, manajemen label dan struktur *table*, dan sebagainya. *PhpMyAdmin* sangat *user friendly*, sangat mudah untuk digunakan walau pengguna baru (*newbie*). (Rumengan, 2021)



Gambar 2. 2 PHP MyAdmin

2.2.7 Database

Database adalah sekumpulan data yang sudah disusun rapi sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi antara satu sama lain sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi yang diinginkan. Selain itu ada pula yang mendefinisikan *database* sebagai beberapa kumpulan *file*, tabel atau arsip yang saling terhubung satu sama lain yang disimpan dalam media elektronik (Arizi, 2017).

Database juga merupakan kumpulan data yang saling terhubung satu dengan yang lainnya, tersimpan di *server* atau didalam komputer sendiri (*localhost*) serta menggunakan perangkat lunak tertentu untuk memanipulasinya. *Database* adalah salah satu komponen yang paling penting dalam sistem informasi. Beberapa *Database* yang ada saat ini adalah : MySQL, *SQL Server*, *Ms.Access*, *Oracle*, dan *PostgreSQL* (Sofwan & Christyono, 2017).

Istilah yang sering digunakan pada pembahasan *database*, yaitu:

1. DBMS, yaitu sistem perangkat lunak yang menyediakan layanan bagi *user* untuk membuat, mengontrol, dan mengakses *database*.
2. *Table*, merupakan kumpulan data yang diorganisasikan dalam baris (*record*) dan kolom (*field*). Dalam satu *database* biasanya terdiri dari beberapa tabel.
3. *Field*, merupakan kolom pada tabel yang memiliki ukuran dan tipe data tertentu.
4. *Record*, merupakan baris data yang memiliki nilai untuk masing-masing *field* pada tabel. Satu tabel biasanya terdiri dari banyak *record*.
5. *Key*, merupakan suatu *field* (kolom) yang dapat dijadikan kunci dalam operasi tabel.
6. SQL, merupakan singkatan dari *Structured Query Language*, yaitu bahasa yang digunakan untuk mengakses dan memanipulasi *database*.

2.2.8 MySQL

MySQL merupakan suatu jenis database server yang sangat terkenal. *MySQL* termasuk jenis RDBMS (*Relational Database Manajement System*). *MySQL* mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki berberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. *MySQL* merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari *MySQL* yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya. *MySQL* lebih empat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL*.
- b. Didukung oleh berbagai bahasa Database Server *MySQL* dapat memberikan pesan *Error* dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
- c. Mampu membuat tabel berukuran sangat besar. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB sampai dengan ukuran *file* yang dapat ditangani oleh sistem operasi yang dipakai.
- d. Lebih murah *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX platform*, *OS/2* dan *Windows Platform*. Melekatnya integrasi PHP dengan *MySQL*. Keterikatan antara PHP dengan *MySQL* yang sama-sama *Software Open-Source* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya. Modul *MySQL* di PHP telah dibuat *Built-in* sehingga tidak memerlukan konfigurasi tambahan pada File konfigurasi Php ini (Hermiati dkk, 2021).

2.2.9 Visual Studio Code

Visual Studio Code (VS Code) adalah kode editor sumber yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *macOS*. Ini termasuk dukungan untuk *debugging*, kontrol git yang tertanam dan *GitHub*, penyorotan sintaksis, penyelesaian kode cerdas, *snippet*, dan *refactoring* kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan *keyboard*, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan (Joni, 2019).

2.2.10 HTML

Hypertext Markup Language ialah suatu bahasa markah yang dipakai dalam membuat suatu tampilan *web*, berbagai informasi didalam sebuah penjelajahan Internet dan format *hiperteks* sederhana, ditulis dengan berkas ASCII agar bisa menghasilkan tampilan yang terintegrasikan. berkas yang dimuat dalam piranti lunak pengolahan data dan disimpan kedalam sebuah format ASCII agar bisa dapat menjadi halaman sebuah *web* dengan perintah HTML. Berawal dari bahasa yang banyak dipakai sebelumnya didalam dunia percetakan dan juga dunia penerbitan yang biasa disebut dengan *Standard Generalized Markup Language* atau SGML, HTML menjadi sebuah standar yang dipakai secara luas dalam menampilkan halaman web (Putra, 2019).

2.2.11 PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari. PHP merupakan bahasa scripting *server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman *website* oleh *browser*. (Noviantoro, 2022).

2.2.12 CSS

CSS adalah singkatan dari *Cascading Style-Sheet* merupakan pengembangan atas kode HTML yang sebelumnya. CSS dapat menentukan suatu struktur dasar halaman web secara cepat dan mudah, (Hasanah, 2013). CSS merupakan Salah satu bahasa desain web (*Style Sheet Language*) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda markup language. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang CSS bisa di aplikasikan untuk segala dokumen XML, termasuk SVG dan XUL bahkan android (Tampubolon, 2018).

2.2.13 Bootstrap

Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout, Typography, Table, Form, Navigation*, dan lain-lain. Di dalam Bootstrap juga sudah terdapat *JQuery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions, Modal, Dropdown, Scrollspy, Tooltip, Tab, Popover, Alert, Button, Carousel* dan lain-lain. Menurut (Nugroho & Setiyawati, 2019), Bootstrap adalah *framework* CSS untuk membuat tampilan *web*. *Bootstrap* menyediakan *class* dan komponen yang sudah siap dipakai.

Hanafri (2018) “Bootstrap adalah *framework frontend* yang intuitif dan kuat untuk pengembangan aplikasi *web* yang lebih cepat dan lebih mudah, Bootstrap menggunakan bahasa pemrograman berupa HTML, CSS, dan *Javascript*”. Menurut Nur Maulana (2021) “*Bootstrap* adalah *library* (pustaka / kumpulan fungsi-fungsi) dari *Framework* CSS yang dibuat khusus untuk bagian pengembangan *frontend* dari suatu *website*. Didalam *library* tersebut terdapat berbagai jenis *file* yang diantaranya HTML, CSS, dan *Javascript*”.

Menurut Tampubolon (2018) “Bootstrap adalah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end website*. Bootstrap juga merupakan salah satu *framework* HTML, CSS dan *javascript* yang paling populer di kalangan *web developer* yang digunakan untuk mengembangkan sebuah *website* yang *responsive*” .

Bootstrap adalah *framework web development* berbasis HTML, CSS, dan JavaScript yang dirancang untuk mempercepat proses pengembangan *web responsive* dan *mobile-first*.



Gambar 2. 3 Bootstrap

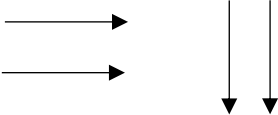
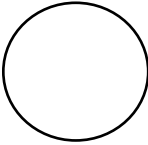
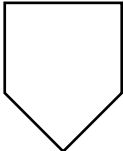

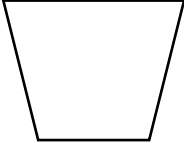
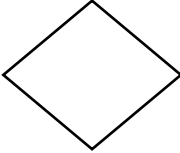

2.2.14 Javascript

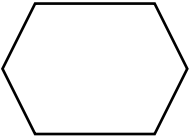

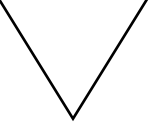


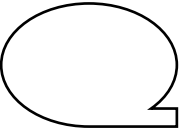

Menurut Sahi (2020), “*Javascript* adalah bahasa yang berupa kumpulan skrip, yang tugasnya dijalankan dalam dokumen HTML. Bahasa ini adalah bahasa skrip pertama di *web*. Bahasa ini merupakan bahasa pemrograman yang memberikan fungsionalitas tambahan pada HTML dengan memungkinkan eksekusi perintah di sisi pengguna, yaitu di sisi *browser*, bukan *server web*”. *JavaScript* adalah bahasa untuk membuat program yang digunakan untuk membuat dokumen HTML yang digunakan, yang ditampilkan di *browser* lebih interaktif. *JavaScript* menyediakan beberapa fungsionalitas ke halaman *web*, sehingga bisa menjadi program yang disajikan melalui antarmuka *web*. *JavaScript* adalah bahasa yang tidak memerlukan kompiler, cukup dengan *interpreter* (Hermiati et al, 2021).


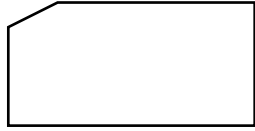
2.2.15 Flowchart

Flowchart atau yang biasa disebut dengan diagram alir merupakan suatu jenis diagram yang mempresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem (Rosaly et al., 2019). *Flowchart* merupakan alternatif lain untuk menggantikan algoritma maupun pseudokode. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun ke *programmer*. Pada dasarnya *flowchart* digambarkan dengan menggunakan simbol-simbol yang mewakili suatu proses tertentu. Untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan menjadi lebih jelas.

Tabel 2. 1 Simbol-simbol Flowchart

NO	SIMBOL	KETERANGAN
1.		<p>Simbol arus / <i>flow</i>, berfungsi untuk menyatakan jalannya arus suatu proses.</p>
2.		<p>Simbol <i>connector</i>, berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.</p>
3.		<p>Simbol <i>offline connector</i>, berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.</p>
4.		<p>Simbol <i>process</i>, berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.</p>
5.		<p>Simbol <i>manual</i>, berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer</p>
6.		<p>Simbol <i>decision</i>, berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak.</p>
7.		<p>Simbol <i>teminal</i>, berfungsi untuk menyatakan permulaan atau akhir suatu program.</p>

8.		<p>Simbol <i>predefined process</i>, berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
9.		<p>Simbol <i>keying operation</i>, berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>.</p>
10.		<p>Simbol <i>offline-storage</i>, berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.</p>
11.		<p>Simbol <i>manual input</i>, berfungsi untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i></p>
12.		<p>Simbol <i>input/output</i>, berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.</p>
13.		<p>Simbol <i>magnetic tape</i>, berfungsi untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis.</p>
14.		<p>Simbol <i>disk storage</i>, berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke disk.</p>

15.		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i>).
16.		Simbol <i>punched card</i> , berfungsi untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.

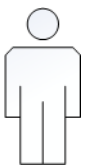

2.2.16 Use Case Diagram



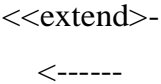
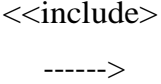
Komponen pembentuk diagram use case adalah : (Yunita, 2018)

- a. Aktkor (*actor*) menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/sarana yang disiapkan oleh bisnis/sistem.

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Pada *Use Case Diagram*, ada beberapa simbol sebagai berikut :

Tabel 2. 2 Daftar Simbol Use Case Diagram

No	Simbol	Keterangan
1.		<i>Actor</i> , mewakili peran orang, sistem lain, atau alat ketika berkomunikasi atau berinteraksi dengan <i>use case</i> .
2.		<i>Use Case</i> , adalah sebuah aksi atau fungsi yang dilakukan oleh sistem.
3.		<i>Association</i> , hubungan antara <i>actor</i> atau objek lain dengan <i>use case</i> tapi



		tidak mengendalikan use case tersebut.
4.		<i>Generalization</i> , menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>usecase</i> .
5.		<i>Extend</i> , menunjukkan bahwa sebuah <i>use case</i> opsional tergantung pada <i>use case</i> lainnya jika suatu kondisi terpenuhi.
6.		<i>Include</i> , menunjukkan bahwa sebuah <i>use case</i> membutuhkan aksi dari <i>use case</i> lain untuk dapat digunakan.

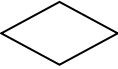


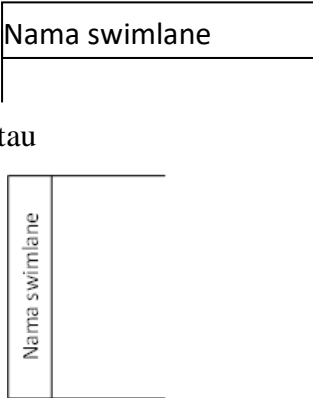

(Sumber: Rosa dan Shalahuddin (2016))

2.2.17 Activity Diagram

Menurut Sucipto dalam (Listiono dan Sintaro, 2021) diagram aktivitas menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktivitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *data flow* diagram pada perancangan terstruktur simbol dalam activity diagram.

Tabel 2. 3 Activity Diagram

No	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i>  atau 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

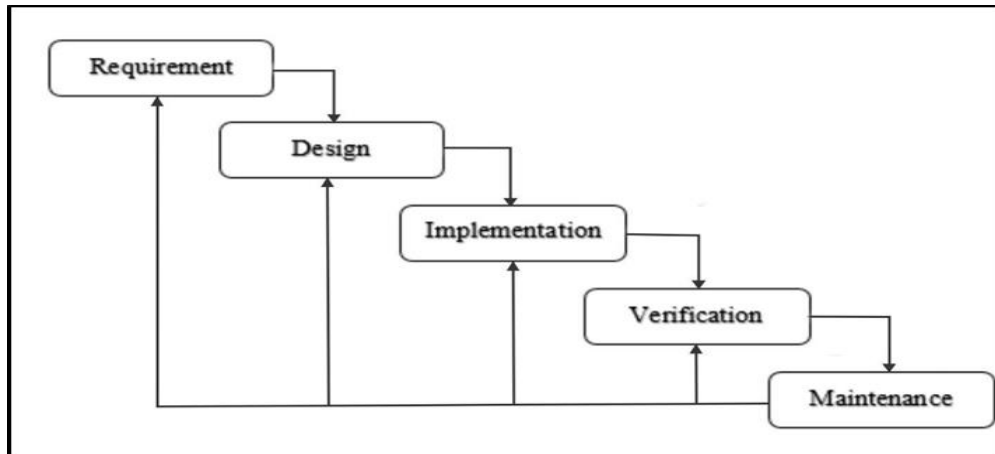
(Sumber: Rosa dan Shalahuddin (2016))

2.3 Metode Pengembangan Sistem

Metode *Waterfall* adalah salah satu model SDLC (*System Development Life Cycle*) yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model *Waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *Waterfall* antara lain *requirement*, *design*, *implementation*, *verification*, dan *maintenance* (Wahid, 2020). Kelebihan menggunakan metode *Waterfall* adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang diperlukan juga mahal. Adapun kelebihan menggunakan metode *waterfall* diantaranya:

1. Urutan proses pengerjaan menggunakan metode ini menjadi lebih teratur dari satu tahap ke tahap yang selanjutnya.

2. Dari sisi *user* juga lebih menguntungkan karena dapat merencanakan dan menyiapkan seluruh kebutuhan data dan proses yang akan diperlukan.
3. Jadwal menjadi lebih menentu karena jadwal setiap proses dapat ditentukan secara pasti. Dengan adanya urutan yang pasti, dapat dilihat pula progress untuk setiap tahap secara pasti.



Gambar 2. 4 Metode *Waterfall*

Sumber : (Pressman, 2012)

2.3.1 Requirement

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2.3.2 Design

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

2.3.3 Implementation

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

2.3.4 Verification

Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem

sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit *testing* (dilakukan pada modul tertentu kode), sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian.

2.3.5 Maintenance

Ini adalah tahap akhir dari metode *waterfall*. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

2.4 Blackbox Testing

Black-Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. *Black box Testing* memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program (Jaya et al., 2018).

Pengujian *black box* hanya melibatkan antara *input* dan *output*. Pengujian ini menangani kebutuhan pelanggan dari *input* yang valid maupun tidak valid. Pengujian *black box* memiliki peran penting dalam pengujian perangkat lunak yaitu untuk memvalidasi fungsi keseluruhan sistem apakah telah bekerja dengan baik (Parlika et al., 2020).

2.5 Whitebox Testing

Metode pengujian yang biasa diterapkan pada evaluasi sistem adalah Pengujian Blackbox dan Pengujian Kotak Putih. (Andriyadi dkk, 2019). Kedua metode ini memiliki perbedaan. Misalnya, pengujian Blackbox hanya mengevaluasi fungsi sistem luar tanpa mengevaluasi kode pengujian bagian dalam. Pengertian lain, White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan

diperiksa lagi sampai mencapai apa yang diharapkan, singkatnya White Box Testing ini menguji dengan cara melihat Pure Code dari suatu aplikasi/software yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut.

2.6 ERD

Menurut Utomo (2010) “ERD merupakan tool analisis sistem pertama yang memusatkan pada data dan keterkaitan antar data serta pengorganisasian data”. Menurut Ladjamudin (2005) “*Entity Relationship Diagram (ERD)* adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam secara abstrak.”

2.7 Komponen ERD

Berikut ini komponen-komponen ERD menurut Ladjamudin(2005), hubungan entitas terdiri dari :

1) *Entity*

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

2) *Relationship*

Pada E-R diagram, *relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3) *Relationship Degree*

Relationship degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.

4) *Atribut*

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship*.

5) *Kardinalitas*

Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. *Kardinalitas* relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya.

6) *Derajat Relationship (Relationship Degree)*

Definisi derajat *relationship* menurut Ladjamudin (2005) “*Relationship Degree* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.”

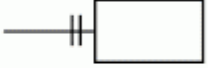
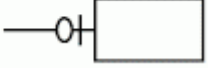
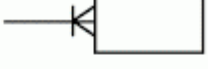


Berikut ini derajat *relationship* yang sering dipakai didalam ERD:

- a) *Unary Relationship* .
- b) *Binary Relationship* .
- c) *Tenary Relationship*.

7) Logical Relationship Structure (LRS)

Menurut Frieyadie (2007) “LRS merupakan hasil dari pemodelan *Entity Relational Ship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Dalam pembuatan LRS terdapat 3 hal yang dapat mempengaruhi (Frieyadie, 2007) yaitu:

- a. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka di gabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
- b. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
- c. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*),
maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

Interpretasi	Min	Max	Notasi
Satu dan hanya satu (<i>exactly one</i>)	1	1	
Nol atau satu (<i>zero or one</i>)	0	1	
Satu atau lebih (<i>one or more</i>)	1	>1	
Nol, satu atau lebih (<i>zero, one, or more</i>)	0	>1	
Lebih dari satu (<i>more than one</i>)	>1	>1	

Gambar 2. 5 Simbol-Simbol ERD