BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Konsep Pemesanan

2.1.1.1 Pengertian Pemesanan

Pemesanan adalah aktivitas yang selalu dilakukan konsumen sebelum proses membeli. Agar dapat memberikan kepuasan kepada konsumen maka dari itu, tentunya perusahaan harus mempunyai sistem pemesanan yang baik.

(Susanti, 2017:2) Menyatakan Pemesanan adalah suatu perjanjian yang dilakukan oleh 2 (dua) pihak atau lebih yaitu pemberi dan pemakai jasa atau barang untuk memenuhi kebutuhannya dalam mengusahakan barang atau jasa tersebut sehingga dapat digunakan. Perjanjian pemesanan tersebut dapat berupa atas pemesanan ruangan, tempat duduk, kamar dan lainnya pada periode waktu tertentu.

2.1.1.2 Kepuasan Konsumen

Kepuasan konsumen adalah perasaan senang atau kecewa seseorang yang muncul setelah membandingkan antara kinerja (hasil) produk yang diperkirakan terhadap kinerja (atau hasil) yang diharapkan. Jika kinerja dibawah harapan, konsumen tidak puas. Jika kinerja memenuhi harapan, konsumen puas. Jika kinerja melebihi harapan, konsumen amat puas/senang (kotler dan keller, 2017:196). Sedangkan menurut danang sunyoto (2015:140). Kepuasan konsumen merupakan salah satu alasan dimana konsumen memutuskan untuk berbelanja pada suatu tempat. Apabila konsumen merasa puas dengan suatu produk, mereka cenderung akan terus membeli dan menggunakannya serta memberitahukan orang lain tentang pengalaman mereka yang menyenangkan dengan produk tersebut.

2.1.1.3 Ketidakpuasan Konsumen

Maftukhah (2016) menjelaskan bahwa "ketidakpuasan konsumen adalah suatu keadaan dimana pengharapan konsumen tidak sama jika kebutuhan tidak terpenuhi maka konsumen bisa saja beralih kemerek lain".

Solomon (1992) dalam Lestari (2011) menyatakan bahwa konsumen memiliki tiga tindakan yang berbeda dalam menghadapi ketidakpuasan (mungkin saja terjadi lebih dari satu tindakan).

1. Voice response.

   Konsumen dapat menyerukan secara langsung ketidakpuasan mereka kepada penjual untuk mendapatkan perbaikan atau penggantian.

2. Private response.

   Konsumen dapat menceritakan ketidakpuasan terhadap toko atau produk kepada teman dan/atau memboikot toko tersebut. Negative word of mouth (WOM) dapat merusak reputasi perusahaan.

3. Third-party response.

   Konsumen dapat mengambil tindakan hukum untuk menentang pedagang, mendaftarkan keluhan pada lembaga konsumen atau menulis surat pembaca di surat kabar.

2.1.1.4 Alur Kerja Proses Pemesanan Tiket Elektronik pada PT. Sinar Dempo Pagaralam

Berikut proses alur pemesanan Tiket Elektronik pada PT. Sinar Dempo Pagaralam:

1. Konsumen perlu membuka internet dan masuk ke alamat website yang sudah disediakan oleh pihak PT. Sinar Dempo Pagaralam
2. Saat melakukan booking tiket maka nama yang dimasukkan haruslah sama dengan nama pada kartu identitas konsumen.
3. Cek dengan benar rute perjalanan anda.
4. Lihat batas pembayaran tiket anda agar tidak mengalami keterlambatan saat membayar tiket, keterlambatan membayar tiket akan mengakibatkan pembatalan proses pemesanan tiket.

2.1.2 Konsep Tiket Elektronik

2.1.2.1 Pengertian Tiket Elektronik

Tiket elektronik adalah sebuah tiket dalam bentuk digital. Tiket elektronik dapat digunakan untuk penerbangan, perkeretaapian, bus, hotel, bioskop dan juga suatu pertunjukan. Tiket Elektronik berisi data rincian perjalanan anda. Yang tercantum di dalam Tiket Elektronik biasanya berisi nama penumpang, rute perjalanan, waktu perjalanan, kelas tiket, dan harga tiket. semua informasi mengenai tiket elektronik disimpan secara digital dalam sistem komputer milik perusahaan.

Tiket elektronik adalah tiket yang dibuat secara elektronik, dimana semua data perjalanan dan penumpang yang tertulis di dalamnya tersimpan dalam sebuah dokumen elektronik dalam database online atau perusahaan penyedia sistem ticketing. Penumpang hanya perlu membawa print out dari data tersebut yang dapat di print berulang kali (Shofwan dan Dian, 2018:9).

Tiket Elektronik merupakan peluang untuk meminimalkan biaya dan mengoptimalkan kenyamanan penumpang. Tiket Elektronik mengurangi biaya proses tiket, menghilangkan formulir kertas dan meningkatkan fleksibilitas penumpang dan agen perjalanan dalam membuat perubahan-perubahan dalam jadwal perjalanan.

2.1.2.2 Manfaat Tiket Elektronik

Dibawah ini adalah beberapa manfaat Tiket Elektronik untuk kehidupan modern saat ini antara lain:

1. Mengurangi biaya yang berkaitan dengan pencetakan dan surat tiket.
2. Mengurangi tenaga kerja yang berkaitan dengan pencetakan dan surat tiket.
3. Keamanan terjamin, karena terdapat berkode validasi dan menghilangkan kemungkinan tiket palsu atau duplikat.
4. Pemesanan E-tiket oleh konsumen berarti mengetahui berapa banyak konsumen perusahaan, karena perusahaan menyimpan data konsumen di database perusahaan
5. Memberikan informasi tambahan yang perlu diketahui bagi pelanggan.

2.1.2.3 Kelebihan Tiket Elektronik

Dibawah ini adalah beberapa kelebihan Tiket Elektronik dibandingkan dengan tiket model lama, antara lain:

1. Tiket Elektronik tidak mungkin hilang.

   Sekalipun tiket tertinggal di rumah, konsumen dapat dengan mudah mencetaknya lagi atau cukup menunjukkan kode booking yang ada di salinan email konsumen. Hal ini memungkinkan karena data konsumen di kertas hanyalah salinan dari data yang ada di database komputer perusahaan. Hanya orang yang bisa membuktikan identitasnya dengan tanda identitas resmi yang berhak menggunakan Tiket Elektronik untuk bisa pergi dalam perjalanan. Ini jauh berbeda dengan sistem tiket yang lama. Konsumen yang kehilangan tiket diwajibkan membayar biaya lagi untuk mengeluarkan tiket yang baru.
2. Tiket Elektronik fleksibel dan lebih efisien

   Sekiranya ada perubahan entah itu tanggal keberangkatan atau tempat duduk, pihak perusahaan cukup mengubah data yang ada di database.
3. E-ticket lebih mudah diperoleh.

   Penumpang yang sudah booking tidak perlu ekstra menghabiskan waktu di jalan untuk mengambil tiket, karena sudah ada diponsel masing-masing.

2.1.3 Metode Pengembangan Sistem

Fathoroni, dkk (2020:19) menyatakan Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “linear sekuential model”, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (planning).

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang berurut. Tahapan tahapan dari metode *waterfall* adalah sebagai berikut:

1. *Requirement Analisis*

   Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk mcmaham i perangkat lunak yang diharapkan oleh pcngguna dan batasan perangkat lunak tersebu t. Informasi ini biasanya dapat diperoleh melalui

wawancara, diskusi atau survei langsung. lnformasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *System Design*

   Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat *keras(hardware)* dan sistem persyaratan dan juga mem bantu dalam mendefinisikan arsitekntr sistem secara keseluruhan.

3. *Implementation*

   Pada tahap i ni, sistem pertama kali dikembangk an di program keci l yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap tmitdikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

4. *Integration & Testing*

   Seluruh unit yang dikembangkan dalam tahap implementasi di integrasikan ke dalam sistem setelah pen gujia n yang dilakukan masing-masing unit. Setelah integrasi selu ruh sistem diuji untu k mengecek setiap kegagalan maupu n kesalahan.

5. *Operation & Maintenance*

   Tahap akhir dalam model *waterfall.* Perangkat lunak yang sudah jadi , dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit system dan peningkatan jasa system sebagai kebutuhan baru.

2.2. Teori Khusus

2.2.1 *Unified Modelling Language* (*UML*)

Rosa dalam Irmayani & Susyatih (2017) “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”.

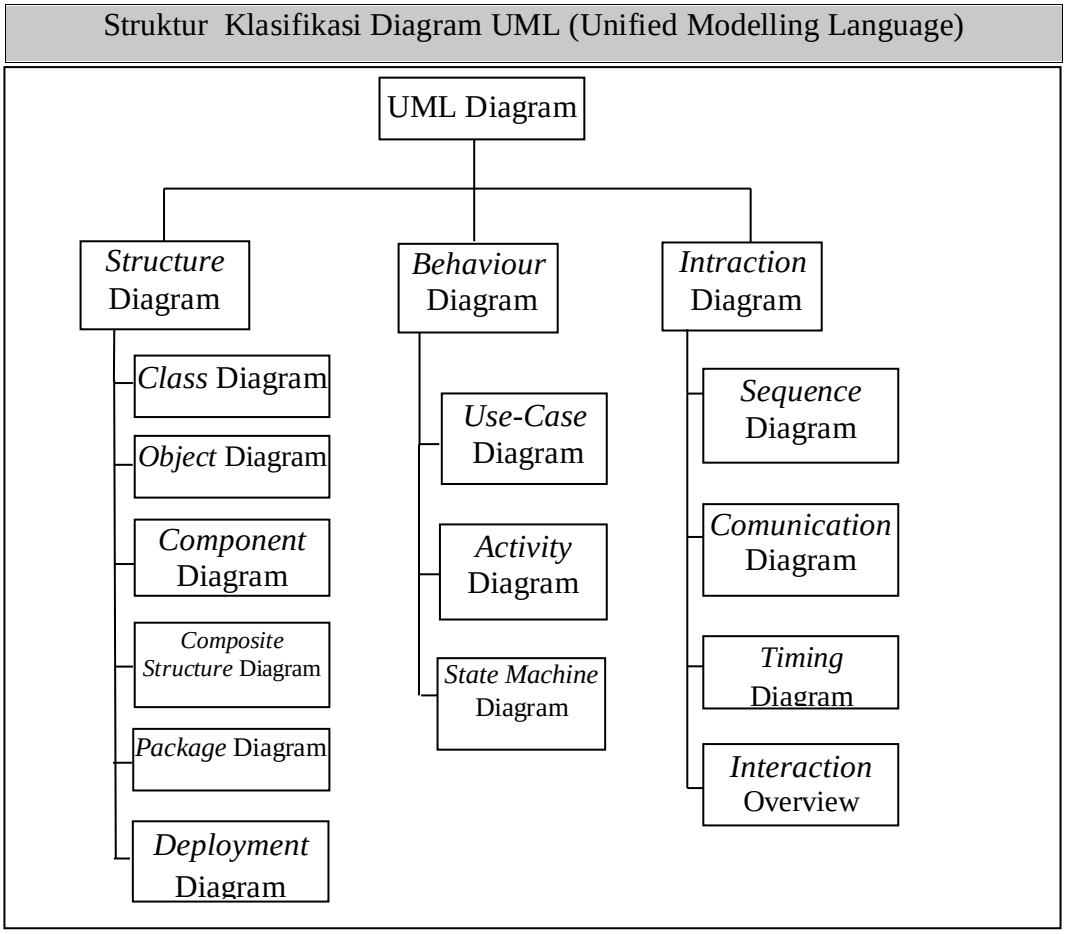
Sukamto dan Shalahuddin (2016:133), “UML (*Unifed Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk

mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek."

Dari pengertian diatas dapat disimpulkan bahwa *UML* (*Unified Modelling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasikan hingga mendokumentasikan sebuah sistem pengembangan perangkat lunak berorientasi objek.

2.2.2 Klasifikasi Diagram *UML* (*Unified Modelling Language*)

Sukamto dan Shalahuddin (2018:140), "Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori". Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.1 Klasifikasi Diagram UML (*Unified Modelling Language*)

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141) :

1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2) *Behavior diagrams* yaitu kumpulam diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahana yang terjadi pada sebuah sistem.

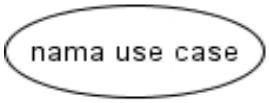
   Interaction diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.3. *Use-Case* Diagram

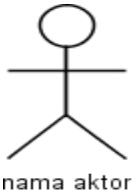
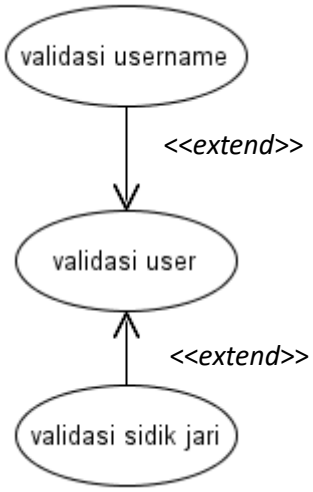
Widodo (dikuip Pranata dkk. 2015:26), *Use-case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *Use-case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Sukamto dan Shalahuddin (2016:155), *"Use-case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use-case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat". Adapun simbol-simbol yang digunakan dalam *use-case* adalah sebagai berikut:

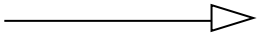
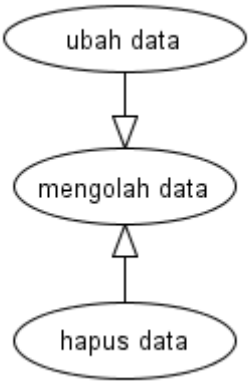
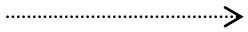
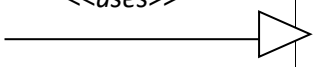
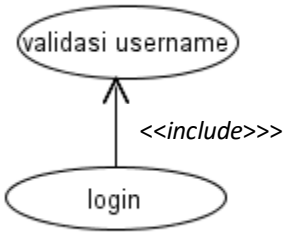
Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
1.	*Use case* 	Fungsionalitas yang disediakan sistem sebagaiunit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frasenama *use case*.

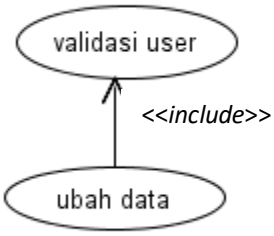
Lanjutan **Tabel 2.1** Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
2.	Aktor / *Actor*  	Orang, proses, atau siatem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.	asosiasi / *association* _____	komunikasi antar aktor dan *use case* yang berpartisioasi pada *use case* atau *use case* memiliki ingeraksi dengan actor.
4.	ekstensi / *extend* .....<<*extend*>>.....>	relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang di tambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misalnya  arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.

Lanjutan **Tabel 2.1** Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
5.	Generalisasi / *generalization* 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,  misalnya: arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).
6.	menggunakan / include / uses  <<*include*>>	Relasi tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini ada dua sudut pandang yang cukup besar mengenai include di *use case*:
	<<*uses*>> 	• *Include* berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan, misalnya pada kasus berikut:  • *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang di tambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:

Lanjutan **Tabel 2.1** Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
		 kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber: Sukamto dan Shalahuddin (2016:156-158)

2.2.4. *Class* Diagram

Sukamto dan Shalahuddin (2016:141), 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.

a. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
b. Operasi atau model adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Tabel 2.2. Simbol-simbol *Class* Diagram

No	Simbol	Deskripsi
1.	Kelas nama_kelas + attribut + operasi ()	Kelas pada struktur sistem.
2.	Antarmuka / *interface* nama_interface	Sama dengan konsep *interface* dalam pemrograman berorientasi objek.

Lanjutan **Tabel 2.2** Simbol-simbol *Class* Diagram

No.	Simbol	Deskripsi
3.	Asosiasi / association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4.	Asosiasi berarah / directed association	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
5.	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan/ d*ependency*	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.	Agregasi / *aggregation*	Relasi antarkelas dengan makna semua-bagian *(whole-part)*.

(*Sumber*: Sukamto dan Shalahuddin (2016:144-147)

2.2.5. *Activity Diagram*

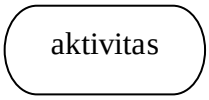
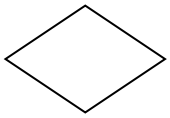
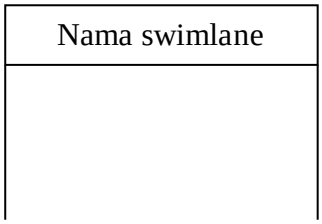
Sukamto dan Shalahuddin (2016:161), *Activity Diagram* atau Diagram Aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana seriap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

4. Rancangan menu yang ditampilkan pada perangkat lunak

Tabel 2.3. Simbol-simbol *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 / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / *join*	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.	*Swimlane* 	*Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

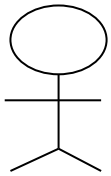
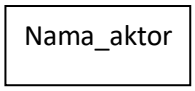
(*Sumber*: Sukamto dan Shalahuddin (2016:162-163))

2.2.6. *Sequence Diagram*

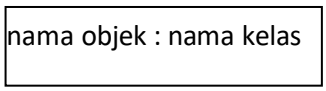
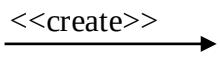
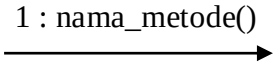
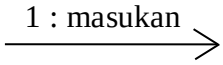
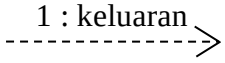
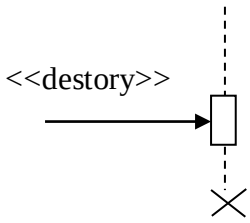
Sukamto dan Shalahuddin (2016:165), "Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karna itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat sekenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka

Tabel 2.4. Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	Aktor  Aktor Atau  tanpa waktu aktif	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 di awal frase nama *actor*.
2.	Garis hidup/ *lifeline* 	Menyatakan kehidupan suatu objek.

Lanjutan Tabel 2.4. Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
3.	Objek 	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7.	Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
8.	Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe *destrory* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

(*Sumber*: Sukamto dan Shalahuddin (2016:165-167))

2.3 Teori Program

2.3.1 Pengertian Basis Data

menurut Saputra (2019:14) “Basis Data merupakan kumpulan data-data yang saling berhubungan satu dengan yang lain yang disimpan dalam perangkat keras komputer dan akan diolah menggunakan perangkat lunak. Basis data sendiri merupakan kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membntuk satu bangunan data”.

Menurut Kristanto (2018:79) “Basis Data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa basis data merupakan kumpulan dari data-data yang terorganisir sebagai aktifitas dari satu atau lebih organisasi yang berelasi dengan cara sedemikian rupa dan memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia.

2.3.2 Pengertian *MySQL*

Gambar 2.1 Logo MySQL

Menurut Sukamto dan Shalahuddin (2018:46) “MySQL adalah bahasa yang digunakan untuk mengelola data pada *Relation Database Management System* (RDBMS) yang dikembangkan berdasarkan teori aljabar relasional dan kalkulus “.

Menurut Rusli, dkk (2019:5), “MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (database) baik yang meliputi proses pembuatan atau proses pengelolaan database”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa MySQL merupakan soyware tempat mengatur jalannya basis data serta mengelola dan membuat basis data tersebut.

2.3.3 Pengertian Xampp

Gambar 2.2 Logo XAMPP

Enterprise (2018:3), "XAMPP merupakan server yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi programmer pemula"

Menurut Joko (2019:11) "Xampp adalah perangkat lunak yang bersifat open source, aplikasi apache yang mudah diinstal dan berisi MariaDB, PHP, dan Perl. Paket *open source* XAMPP telah diatur agar sangat mudah untuk di install dan digunakan".

Menurut Yosef (2017:1), "Xampp adalah perangkat lunak gratis yang bebas digunakan, sebagai server yang berdiri sendiri (localhost) yang ditulis dengan bahasa pemrograman".

Berdasarkan pendapat diatas, dapat disimpulkan bahwa Xampp adalah perangkat lunak gratis yang bebas digunakan programmer pemula bersifat open source, dan berfugsi menjalankan mysql dan phpmyadmin.

2.3.4 Pengertian PHP

Gambar 2.3 Logo PHP

Menurut Rohi (2018:127) "PHP merupakan kependekan dari Hypertext Preprocessor yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server, tujuan dari bahasa ini adalah membantu para

Menurut Anton (2018:1) "PHP merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah web server (server side)".

Berdasarkan pendapat diatas, dapat disimpulkan bahwa PHP atau *Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat dimasukan ke dalam HTML/PHP banyak dipakai untuk membuat situs web dinamis dan membantu para pengembangan web dengan cepat.

2.3.5 Skrip Dasar PHP

Bentuk skrip penulisan PHP, seperti berikut :

```
<?php          // awal tag
……
.....……        } Tuliskan script PHP disini
.....……
?>
```

2.3.6 Pengertian Framework

Menurut Yudho dan Helmi (2018:8), Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan script (terutama kelas dan function) yang dapat membantu developer dalam menangani berbagai masalah-masalah dalam pemrograman, Seperti koneksi ke database, pemanggilan variable, file, dan lain-lain sehingga pekerja developer lebih fokus dan lebih cepat dalam membangun aplikasi

2.3.7 Pengertian Codeigniter

Jubile (2015:1) CodeIgniter merupakan framework aplikasi web yang gratis, ringan, mudah diinstal dan sangat memudahkan anda.

2.3.8 Pengertian HTML

Menurut Faisal dan Abadi (2020:1), “Hypertext Markup Language atau lebih dikenal dengan HTML adalah bahasa yang digunakan untuk membuat antarmuka halaman web”.

Sedangkan, menurut Abdulloh (2018:7), “HTML singkatan dari Hyper Text Markup Language, yaitu bahasa standar web yang dikelola penggunanya oleh W3C (World Wide Web Consortium) berupa tag-tag yang menyusun setiap elemen dari website”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa HTML kependekan dari Hyper Text Markup Language, bahasa yang wajib dipahami jika

anda ingin bisa membuat website yang statis dengan file teks murni yang dapat dibuat dengan editor teks sembarang dan harus menggunakan bahasa lainnya seperti JavaScript, Java,PHP,dll.

2.3.9 Pengertian Sublime Text

Menurut Supono dan Putratama (2016:15), "Sublime Tect merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi".

dapat disimpulkan bahwa Sublime Text adalah software editor berbasis Python, yang memiliki banyak fitur dan mudah digunakan sehingga memudahkan programmer dalam membuat suatu program.

2.3.10 Pengertian PHPMyadmin

Menurut Sarwandi (2016:27), "PHPMyadmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer".

Dari pengertian diatas dapat disimpulkan bahwa PHPMyadmin adalah software berbasis PHP yang digunakan sebagai pengatur untuk manajemen database MySQL pada komputer.