BAB II

LANDASAN TEORI

2.1 Omzet (Pendapatan)

Omzet merupakan jumlah uang hasil penjualan barang (dagangan) tertentu selama suatu masa jual. Dapat dikatakan omzet dari laba kotor atau pendapatan kotor yang dihasilkan dari usaha.

Nilai omset sebuah perusahaan kerap jadi patokan menilai golongan perusahaan tersebut (Nining Wahyuni, 2020)

2.2 *Website*

2.2.1 HTML

HTML adalah bahasa markup (*markup language*) seperti yang ada di dalam singkatan HTML itu sendiri. Itu artinya, HTML adalah bahasa struktur untuk menandai bagian-bagian dari sebuah halaman (Jubilee, 2016).

2.2.2 CSS

CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti font, color, text dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan (Lewenusa, 2020).

2.2.3 JavaScript

Javascript adalah sebuah program yang berjalan pada client side beda dengan PHP yang berjalan pada server side, *javascript* secara dasar lebih

difungsikan bagaimana seorang untuk mempermudah seseorang yang mengunjungi sebuah web untuk berinteraksi (Kurosaki, 2018).

2.2.4 PHP

PHP merupakan bahasa pemrograman untuk membuat *website* sehingga *website* menjadi dinamis. Bahasa pemrograman ini paling banyak digunakan untuk membuat *website* karena mudah untuk dipelajari dan gratis (Putratama & Supono, 2016).

2.2.5 Bootstrap

Bootstrap adalah sebuah *library framework* CSS yang didalamnya terdapat terdiri dari komponen-komponen seperti *class* yang sudah siap digunakan, sehingga *framework* ini sangat berguna untuk programmer khususnya bagian pengembang *front-end website* karna hanya perlu memanggil *class*nya saja dan tidak harus lagi membuat coding CSS dari awal (Muhammad, 2020).

2.3 Basis Data (*Database*)

2.3.1 Pengenalan Basis Data (*Database*)

Menurut Robi Yanto, M.Kom. (2016:10) Basis Data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas, gudang, tempat berkumpul. Sedangkan data adalah fakta yang mewakili suatu objek seperti manusia, barang, hewan, keadaan, peristiwa dan sebagainya, yang direkam dalam bentuk angka, huruf simbol, teks gambar, bunyi atau kombinasinya.

Menurut Robi Yanto, M.Kom. (2016:11) Basis data sendiri dapat

didefinisikan dalam sejumlah sudut pandang seperti:

a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.

b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi), untuk memenuhi berbagai kebutuhan.

c. Kumpulan file yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.3.2 Komponen Sistem Basis Data (*Database*)

Menurut Robi Yanto, M.Kom. (2016:12)

a. Data

Merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.

b. *Hardware*

Merupakan perangkat keras berupa komputer dengan media penyimpanan yang digunakan untuk menyimpan data karena pada umumnya basis data memiliki ukuran yang besar.

c. Sistem Operasi

Program yang mengaktifkan dan memfungsikan sistem computer mengendalikan seluruh sumber daya dalam komputer, dan melakukan operasi dasar dalam komputer meliputi input, proses dan output.

d. Basis Data

Basis data sebagai inti dari sistem basis data. Basis data menyimpan data

serta struktur sistem basis data baik untuk entitas maupun objek-objek secara detail.

e. *Database Management System*

Merupakan perangkat lunak yang digunakan untuk melakukan pengelolaan basis data. Sebagai contoh Microsoft *access*, Paradox, Sql Server, Mysql, Oracle.

f. *User*

User merupakan pengguna yang menggunakan data yang tersimpan dan terkelola. *User* dapat berupa seseorang yang mengelola basis data yang disebut *Database* administrator (DBA), bisa juga disebut end user.

g. Aplikasi Lainnya

Program yang dibuat untuk memberikan interface kepada *user* sehingga lebih mudah dan terkontrol dalam mengakses basis data.

2.4 MySQL

Menurut Jubilee Enterprise (2018:2) MySQL merupakan server yang melayani *Database*. Untuk membuat dan mengolah *Database*, kita dapat mempelajari pemrograman khusus yang disebut query (perintah) SQL. *Database* sendiri dibutuhkan jika kita ingin menginput data dari user menggunakan form HTML untuk kemudian diolah PHP agar bisa disimpan ke dalam *Database* MySQL.

2.5 UML

2.5.1 *Use Case Diagram*

Menurut Mesran. (2019:9) *Use case Diagram* merupakan suatu pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case Diagram* merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan. *Use case Diagram* dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan dari suatu sistem. Jadi, dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu, bagaimana cara *actor* akan menggunakan sistem, serta apa aja saja yang dapat dilakukan terhadap suatu sistem. Berikut merupakan simbol-simbol yang ada pada *diagram use case.*

Tabel 2.1 Simbol-simbol *Use Case Diagram* (Mesran, 2019)

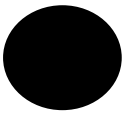
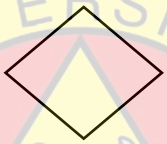
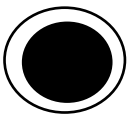
Nama Komponen	Simbol	Deskripsi
Use Case		Gambaran fungsionalitas dari suatu sistem, sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
Actor		Sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi

		dengan sistem.
Association		Komunikasi antara aktor dan *use case* yang berpartisipasi pada use case dengan memiliki interaksi dengan aktor
Extend/ Ekstensi		Relasi use case tambahkan ke seluruh use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan.
Generalization/ Generalisasi		Sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.

2.5.2 *Activity Diagram*

Menurut Mesran. (2019:10) *Activity Diagram*, yaitu *Diagram* yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang digunakan untuk menggambarkan alur kerja (aktivitas) pada *use case* (proses), logika, proses bisnis dan hubungan antara aktor dengan alur-alur kerja *use case*. Berikut merupakan simbol-simbol yang ada pada *Activity Diagram*.

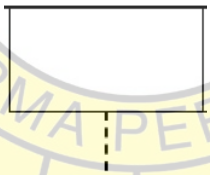
Tabel 2.2 Simbol-simbol *Activity Diagram* (Mesran, 2019)

Nama Komponen	Simbol	Deskripsi
Start State/ Status awal		Sebagai tanda awal proses dari *activity Diagram.*
Activity/ Aktivitas		Memiliki fungsi yang sama dengan *state*. Menampung *event* atau aktifitas pada proses sistem.
Decision/ Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Join/ Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
State Transition		Berfungsi untuk menunjukan aliran atau urutan dari event atau aktifitas pada *Diagram.*
End State		Sebagai tanda akhirdari *activity Diagram.*

2.5.3 *Sequence Diagram*

Sequence Diagram memodelkan kolaborasi objek berdasarkan urutan waktu. Ini menunjukkan bagaimana objek berinteraksi dengan orang lain dalam skenario tertentu dari *use case* (Harry & Lusia, 2020). Beberapa komponen dalam *sequence* antara lain:

Tabel 2.3 Simbol-simbol *Sequence Diagram* (Harry & Lusia, 2020).

Nama Komponen	Simbol	Deskripsi
Aktor		Sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem.
Boxes		Sebuah kotak yang tampil pada posisi paling atas *Diagram*, yang mewakili *object*, *use case*, *class* dan *actor*.
Return Message		Menggambarkan pesan atau hubungan antar objek yang menunjukan urutan kejadian yang terjadi.

Lifeline		Eksekusi objek selama sekuensial (*message* dikirim atau diterima dan aktivasinya).
Message to Self		Menggambarkan pesan atau hubungan objek itu sendiri yang menunjukan urutan kejadian yang terjadi.
Object Message		Menggambarkan pesan atau hubungan antar objek yang menunjukan urutan kejadian yang terjadi.

2.6 Metodologi Pengembangan Sistem

Model *waterfall* ini sebenarnya adalah "*Linear Sequential Model",* yang sering juga disebut dengan "*classic life cycle"* atau model waterfall. Metode ini muncul pertama kali sekitar tahun 1970 sehingga dianggap kuno, tetapi merupakan model/metode yang paling banyak dipakai di dalam *Software Engineering* (SE). Metode ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, testing/verification, and maintenance. Disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan

(Muharto dan Arisandy Ambarita, 2016). Metode *waterfall* terbagi menjadi lima tahapan, yaitu sebagai berikut:

1. *Requirements Definition (Analisis Kebutuhan)*

Pada tahap ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Seseorang sistem analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menerjemahkan kedalam bahasa pemrograman.

2. *System and Software Design (design sistem)*

Pada tahap ini, proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. *Coding & Testing (Penulisan Sinkode Program/Implementation)*

Pada tahap ini, coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan

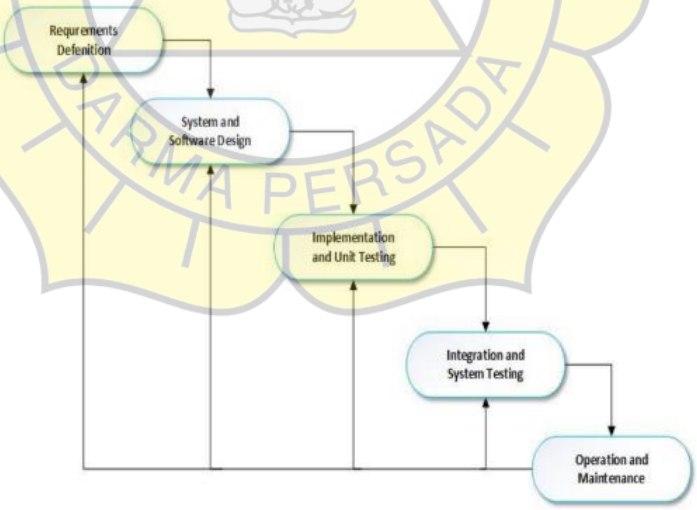
dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. *Integration and System Testing (Penerapan /Pengujian Program)*

Pada tahap ini, bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadikan digunakan oleh *user*.

5. *Operation and Maintenance (Pemeliharaan)*

Pada tahap ini, perangkat lunak yang susah disampaikan kepada user pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena user membutuhkan perkembangan fungsional.



Gambar 2.1 Model Waterfall

2.7 Metode

2.7.1 *Clustering*

Clustering adalah suatu metode pengelompokan berdasarkan ukuran kedekatan atau (kemiripan). *Clustering* beda dengan group, kalau group berarti kelompok yang sama, kondisinya kalau tidak pasti bukan kelompoknya. Namun kalau cluster tidak harus sama hanya kelompok yang kedekatannya berdasarkan dari suatu karakteristik sampel yang ada, yaitu dengan menggunakan rumus jarak euclidean distance (Prianto & Sulpadianti, 2020).

2.7.2 *Exponential Smooth*

Exponential smoothing adalah suatu peramalan rata-rata bergerak yang melakukan pembobotan menurun secara eksponensial terhadap nilai-nilai observasi yang lebih tua. Metode *exponential smoothing* merupakan pengembangan dari metode moving average (Fachri & Esi, 2020).