BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Invoice

Berdasarkan (Rahmayu, Serli & Hanawi, 2019) “*invoice* ialah sistem pelayanan yang sangat memberikan kemudahan kepada pelanggan dalam melakukan proses pembayaran dan memudahkan perusahaan dalam mengetahui jumlah tagihan”. *Invoice terdiri dari 3* macam :

1. *Proforma invoice* adalah dokumen perincian tentang keterangan barang-barang yang dijual dan harga dari barang-barang tersebut.
2. *Commercial invoice* adalah dokumen perincian tentang keterangan barang-barang yang dijual dan harga dari barang-barang tersebut.
3. *Consular invoice* adalah *invoice* yang dikeluarkan oleh instalasi resmi yakni kedutaan atau konsulat.

2.1.2 Pengertian Sistem

Suatu sistem ialah jaringan kerja dari beberapa prosedur yang saling berkaitan dan berhubungan, berkumpul secara bersama untuk melaksanakan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu” (Dudung, 2018).

Marshall B. Romney menyatakan bahwa “sistem ialah suatu rangkaian komponen yang saling terikat dan berinteraksi untuk mencapai tujuan” (Taufan, 2020).

Dari pendapat diatas dapat disimpulkan bahwa suatu sistem merupakan kumpulan dari prosedur-prosedur dan komponen yang saling berhubungan untuk melakukan kegiatan demi mencapai tujuan tertentu.

1. Sistem Berorientasi Obyek (OOP)

Sebuah model obyek menangkap struktur statis dari sistem dengan menggambarkan obyek sistem, hubungan antara obyek, serta atribut dan operasi yang merupakan karakteristik setiap kelas dan obyek.

Model berorientasi obyek lebih lebih mendekati keadaan nyata, dan dilengkapi dengan penyajian grafis dari sistem yang bermanfaat untuk komunikasi dengan user dan pembuatan dokumentasi struktur dari sistem.

a. Obyek

Didefinisikan sebagai konsep, abstraksi, atau benda dengan Batasan dari arti untuk suatu masalah. Semua obyek mempunyai identitas yang berbeda dengan lainnya. Kadang-kadang obyek berarti suatu barang, maka digunakan istilah *object instance*, dan *object class* untuk menunjukan suatu grup dari barang yang sama.

b. Kelas

Suatu *object class* menggambarkan kumpulan dari obyek yang mempunyai sifat (atribut) perilaku umum (operasi) relasi umum dengan obyek lain dan sematik umum dengan obyek lain dan sematik umum. Obyek dan *object class* sering sama sebagai benda dalam deskripsi masalah.

2.1.3 Website

Website ialah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan gabungan dari semuanya baik yang bersifat dinamis yang membentuk satu rangkaian bangunan yang saling terkait dengan dihubungkan jaringan halaman. Hubungan satu halaman *web* dengan halaman *web* lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext* (Ghaffari, 2017).

1. *Web Browser*

Web browser ialah suatu aplikasi perangkat lunak (*software*) yang biasa digunakan dalam pencarian berbagai macam informasi yang berada di dunia maya atau internet. Aplikasi ini dapat memudahkan penggunaanya dalam melakukan akses data atau mencari informasi yang dibutuhkan (Munah, 2018).

2. *Web Server*

Web server ialah aplikasi perangkat lunak (*software*) yang memiliki fungsi sebagai penerima dari suatu permintaan yang telah dikirimkan melalui *browser* kemudian memberikan suatu tanggapan dari permintaan yang dikirimkan dalam bentuk HTML. Namun, *web server* dapat memberikan dua pengertian yang berbeda, diantaranya menjadi bagian dari suatu perangkat lunak ataupun menjadi perangkat keras (Yasin K, 2018).

3. *Internet*

Internet ialah jaringan komunikasi yang bersifat global yang menghubungkan milyaran jaringan komputer secara terbuka menggunakan standar sistem protocol control transmisi global suite protocol internet (TCP/IP) (Rarasari, 2018).

2.1.4 Basis Data

Berdasarkan (Wahyuni, 2021) “basis data ialah sekumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data ialah komponen penting dalam sistem informasi, karena basis yang menyediakan informasi para pengguna”.

1. MySQL

Menurut Raharjo menyatakan bahwa MySQL ialah *software* RDBMS (*server database*) yang dapat mengolah *database* dengan cepat, dapat menampung data dalam jumlah yang besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara bersamaan (*multi-threaded*) (Rizkiansyah, 2018).

2. XAMPP

Berdasarkan (Supono & Putratama, 2018) “XAMPP ialah terdiri dari dari *apache web server*, MySQL, PHP, Perl, FTP *server* dan *phpMyAdmin*”.

2.1.5 Model Pengembangan Perangkat Lunak

Menurut (Maryani, 2018) “Model *Prototype* merupakan sebuah metode yang wajib pengembang perangkat lunak membuat sebuah *Mockup* berupa model aplikasi, sangat cocok pada kondisi dimana pengguna tidak bisa menyajikan informasi secara jelas mengenai kebutuhan yang sesuai dengan keinginannya”. Hasil dari model *prototype* berupa *mockup* yang akan menjadi rujukan model desain yang digunakan saat berlatih, presentasi, penilaian sebuah desain promosi atau keperluan lain.

Langkah – langkah dalam *Prototyping* adalah sebagai berikut:

1. Pengumpulan kebutuhan.
2. Proses desain.
3. Membangun *Prototype*.
4. Pengujian *Prototype*.

Mengumpulkan kebutuhan, mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sebuah sistem yang akan dibuat. Desain mengutamakan pada representasi dari aspek perangkat lunak dari sudut pengguna, ini mencakup *input*, *proses* dan format *output*, desain cepat menuju pembangunan *prototype* dianalisis oleh pengguna dan bagian analisis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. *Prototype* diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detail apa yang perlu dilakukannya, setelah keempat *prototyping* dijalankan, maka langkah selanjutnya adalah pembuatan atau perancangan produk sesungguhnya.

2.2 Teori Pendukung

Adanya teori pendukung bertujuan untuk mendukung jalannya sistem, mempermudah pekerjaan dan supaya sistem yang didesain menjadi terstruktur karena semuanya didesain secara detail mulai dari rancangan dan lainnya.

2.2.1 Entity Relationship Diagram (ERD)

Mulani menyatakan bahwa “ERD merupakan alat yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antara entitas (*relationship*) secara abstrak (konseptual)” (Susilawati, 2020).

2.2.2 Logical Record Structure (LRS)

Berdasarkan (Rosa & Shalahuddin, 2018) “*Logical Record Structure* (LRS) ialah tranformasi dari penggambaran ERD dalam bentuk yang lebih detail dan mudah untuk dipahami.

2.2.3 Unified Modeling Language (UML)

Berdasarkan (Rosa & Shalahuddin, 2018) “*Unified Modeling Language* (UML) ialah salah satu standar bahasa yang dipergunakan didunia industri untuk mengidentifikasi, *requirement*, membuat analisis & desain, serta mendeskripsikan arsitektur dalam pemrograman berorientasi objek”. Berikut ini beberapa contoh yang terdapat pada *Unified Modeling Language* (UML) :

1. *Activity Diagram*

Activity Diagram ialah diagram status yang menggambarkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini penting dalam pemodelan fungsi-fungsi suatu sistem serta memberi tekanan pada aliran kendali antar objek.

2. *Use Case Diagram*

Diagram ini menunjukkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini sangat penting untuk

mengorganisasi dan memodelkan sikap suatu sistem yang diperlukan dan diharapkan pengguna.

3. *Class Diagram*

Diagram ini bersifat statis, dimana diagram ini menggambarkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Pada umumnya diagram ini dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram ini memuat kelas-kelas aktif.

4. *Sequence Diagram*

Diagram ini bersifat dinamis dan urutan interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

2.2.4 Black Box

Berdasarkan (Cholifah, 2018) “Metode *Black box Testing* ialah salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah serta batas atas dari data yang diharapkan, perkiraan banyaknya data uji bisa dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang wajib dipenuhi serta masalah batas atas dan batas bawah yang memenuhi”. Dan dengan metode ini dapat diketahui bila fungsionalitas masih bisa mendapatkan masukan data yang tidak diperlukan maka mengakibatkan data yang disimpan kurang meyakinkan.

Solusi praktis peningkatan akurasi perlu dilakukan segera guna memperbaiki celah *error* yang telah ditemukan, berikutnya dilakukan pengujian keamanan secara intensif melalui jaringan internal (*whitebox penetration testing*) secara berpola oleh Sistem Administrator atau Pengelola Sistem Informasi, khususnya bagi yang

mengelola perangkat lunak tersebut dan untuk mencapai tingkat akurasi, dimana seluruh parameter akurasi yang terkait aspek kerahasiaan, integritas data, dan avalibilitas data bisa terpenuhi, maka wajib dipertimbangkan metode lain yang dapat dijadikan tolak ukur standar keamanan informasi.

2.2.5 Figma

Figma adalah aplikasi desain antarmuka yang berjalan di *browser*. Figma memberi semua alat yang dibutuhkan untuk fase desain proyek, termasuk alat gambar vektor yang mampu membuat ilustrasi sepenuhnya, serta kemampuan *prototyping* (Bracey, 2018).