

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Sebelumnya, telah banyak penelitian dan pengembangan aplikasi yang dilakukan untuk meningkatkan kinerja perusahaan. Di antaranya adalah penelitian yang dilakukan oleh (Agustina, 2015) yang berjudul Sistem Informasi Penjualan (Studi Kasus di Counter Ketro). Dalam penelitiannya, disimpulkan bahwa dengan adanya Sistem Informasi Pembayaran Les pengolahan data pembayaran menjadi lebih efektif dan efisien serta penyimpanan data pembayaran tidak membutuhkan berkas banyak dan data bisa digunakan dalam jangka waktu yang relatif lama.

Ada pengembangan aplikasi yang dilakukan oleh (Andalia & Budi Setiawan, 2015) yang diberi judul Pengembangan Sistem Informasi Pengolahan Data Pencari Kerja Pada Dinas Sosial Dan Tenaga Kerja Kota Padang menjelaskan bahwa dengan sistem dapat menjamin keakuratan dan ketepatan dalam pembuatan laporan.

Penulis menggunakan *Programming Microsoft ASP.NET MVC Third Edition* yang ditulis oleh (Esposito, 2014) untuk mempelajari dasar desain perangkat lunak dan cara memprogram pada *ASP.NET MVC*.

2.2 Landasan Teori

2.2.1 Data

Menurut (Andalia & Budi Setiawan, 2015), data merupakan kenyataan yang mendeskripsikan sebuah kejadian dan kesatuan nyata.

2.2.2 Informasi

Menurut (Andalia & Budi Setiawan, 2015), informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan memberikan arti dan nilai bagi penerimanya. Kualitas informasi dinilai dari relevansi, keakuratan, kelengkapan, ketepatan waktu dan bisa dimengerti.

2.2.3 Sistem

Menurut (Masykur, Makruf, & Atmaja, 2015), sistem adalah kumpulan elemen yang saling berinteraksi untuk menyelesaikan suatu sasaran tertentu. Sistem merupakan himpunan komponen yang terorganisasi, saling berinteraksi, bergantung satu sama lain dan terpadu. Sistem digunakan untuk memproses masukan agar dapat menghasilkan keluaran.

2.2.4 Sistem Informasi

Menurut (Iswandy, 2015), sistem informasi adalah sistem yang memiliki tugas untuk mengolah data, mendukung operasi organisasi, serta menyediakan informasi. Sistem informasi juga didefinisikan sebagai kumpulan komponen yang saling berhubungan atau berinteraksi yang berfungsi mengumpulkan atau mendapatkan, mendistribusikan informasi, serta menyimpan informasi untuk mendukung keputusan dan pengawasan di dalam suatu organisasi.

2.2.5 ASP.NET MVC (Model View Controller)

Menurut (Galloway, Wilson, Scott Allen, & Matson, 2014), *ASP.NET MVC* adalah sebuah *framework* untuk membangun aplikasi *web* dengan

menerapkan konsep *Model-View-Controller* pada *ASP.NET framework*. *Model* adalah kumpulan *class* yang memrepresentasikan bentuk data yang digunakan. *View* adalah yang bertanggung jawab untuk menyediakan tampilan ke *user* sedangkan *Controller* adalah kumpulan *class* yang mengatur data yang datang dari *user*, keseluruhan alur aplikasi dan logika aplikasi serta menyediakan data keluar ke *View* yang sesuai.

2.2.6 Entity Framework

Menurut (Singh, 2015), *Entity Framework* adalah *Object Relational Mapper (ORM)* dari Microsoft yang memungkinkan para pengembang aplikasi bekerja dengan model bisnis dengan data yang berelasi. *Entity Framework* berinteraksi antara *.NET* dan relasional *database*.

Keuntungan dari *Entity Framework* adalah waktu pengembangan berkurang karena tidak perlu menulis semua kode *ADO.NET* yang diperlukan untuk akses data, dapat memiliki semua logika akses data yang ditulis dalam bahasa tingkat yang lebih tinggi seperti *C#* daripada menulis *query SQL* dan *stored procedure* (Singh, 2015).

2.2.7 LINQ (Language Integrated Query)

LINQ atau disebut dengan *Language Integrated Query* adalah sintak *query* yang seragam digunakan pada *C#* dan *VB* digunakan untuk menyimpan dan mengambil data dari sumber yang berbeda. *LINQ* terintegrasi dalam *C#* atau *VB*, sehingga menghilangkan ketidakcocokan antara bahasa pemrograman dan *database*, serta menyediakan antarmuka *query* tunggal untuk berbagai jenis sumber data.

LINQ memungkinkan untuk melakukan *query* data dari dalam bahasa pemrograman .NET dengan cara yang sama seperti *SQL* memungkinkan untuk *query* data dalam database (Spaanjaars, 2014).

2.2.8 Definisi Website

Menurut (Akhmad & Noor, 2015), *website* merupakan sekumpulan halaman yang menampilkan informasi seperti teks, gambar atau video, animasi, suara atau gabungan dari semuanya yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.2.9 HTML

Menurut (Nugroho, Ahmad, St, Sun, & St, 2016), *html (hyper text markup language)* adalah bahasa standard yang digunakan untuk mengatur tampilan dan isi dari halaman *web*.

2.2.10 Javascript

Menurut (Wahyudi, Dewi, & Wibowo, 2017), *javascript* adalah bahasa *script* berbasis objek untuk meningkatkan interaksi seperti respon klik, *input* dari *form* dan *event* lainnya antara pengguna dengan halaman *html*. *Javascript* digunakan untuk pembuatan aplikasi web dan program *javascript* diletakkan pada halaman *html*.

2.2.11 *JQuery*

Menurut (Hidayat, Marlina, & Utami, 2017), *jquery* adalah *library javascript* yang didesain untuk menyederhanakan pembuatan aplikasi *web* dengan memanggil kode *javascript* siap pakai yang dapat langsung dipakai tanpa perlu membuat kode secara panjang lebar.

JQuery juga dilengkapi dengan fitur untuk memanipulasi *event* dan element pada dokumen *HTML* (Wahyudi et al., 2017).

2.2.12 *CSS*

CSS atau yang disebut dengan *Cascading Style Sheet* adalah bahasa pemrograman untuk mengendalikan tampilan suatu halaman web (Iriadi & Indrasari, 2017).

Salah satu fitur *CSS* adalah memungkinkan pemisahan presentasi situs dari konten web. Pemisahan ini mengurangi kompleksitas dan pengulangan yang berkaitan dengan informasi tampilan. Pemisahan juga membuat mudah untuk menegakkan konsistensi pada situs web karena seluruh tampilan *web* bisa dikontrol dari satu dokumen *CSS* (Olsson, 2014).

2.2.13 *Bootstrap*

Bootstrap merupakan *framework* untuk mempercepat dan memberi kemudahan dalam mengembangkan *website* (Ariansyah, Fajriyah, & Prasetyo, 2017). *Bootstrap* memiliki template *HTML* dan *CSS* untuk *form*, tombol, tabel dan navigasi pada halaman web. *Bootstrap* juga memberi kemampuan untuk

membangun *website* yang *responsive* yang dapat menyesuaikan tampilan kepada semua perangkat dengan mudah.

2.2.14 Database

Menurut (Nugrahanti, 2015), pengertian *database* adalah kumpulan data (elementer) yang secara *logic* berkaitan dalam mempresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam *system* tertentu. Dari definisi diatas maka dapat disimpulkan bahwa *database* adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, yang kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2.2.15 DBMS (Database Management System)

Menurut (Septa, Hamzah, & Andayati, 2014), sistem manajemen basis data atau *DBMS* merupakan perangkat lunak yang bertugas untuk mendefinisikan, menciptakan, mengelola serta mengendalikan pengaksesan basis data.

2.2.16 SQL Server

Microsoft SQL Server adalah perangkat lunak *relational database management system (RDBMS)* yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas. *Microsoft SQL Server 2000* merupakan bahasa komputer standar yang ditetapkan oleh *ANSI (American National Standar Institute)* untuk mengakses dan memanipulasi sistem database (Suripto & Triyono, 2014).

2.2.17 Entity Relationship Diagram (ERD)

Menurut (Susila & Triyono, 2015), *Entity Relationship Diagram (ERD)* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

2.2.18 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah bahasa pemodelan yang standar untuk lingkungan berorientasi objek, yang berisi notasi-notasi grafis yang relatif sudah dibakukan (*Open Standard*). *UML* mendeskripsikan objek, proses, aturan dan hubungan antar objek yang terjadi pada sistem (Lenti, 2014).

Menurut (Lenti, 2014), Ada tiga karakter penting yang melekat pada *UML* yaitu sketsa, cetak biru dan bahasa pemrograman. Sebagai sebuah sketsa, *UML* bisa berfungsi sebagai jembatan dalam mengomunikasikan beberapa aspek dari sistem. *UML* bisa juga berfungsi sebagai cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi detail tentang bagaimana alur program dan logika yang akan di aplikasikan pada program (*Forward Engineering*) atau bahkan membaca program dan menginterpretasikannya kembali ke dalam diagram (*Reverse Engineering*). Sebagai bahasa pemrograman, *UML* dapat menerjemahkan diagram yang ada di *UML* menjadi kode program yang siap untuk dijalankan. Beberapa diagram yang umum ada dalam pemodelan *UML* adalah sebagai berikut :

1. Diagram *Use Case*.

2. Diagram *Sequence* (Alur).

3. Diagram *Class* (Kelas).

2.2.19 *Flowchart*

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis (Iswanday, 2015). Simbol-simbol yang digunakan untuk menggambar *flowchart* yaitu persegi panjang, lingkaran, jajar genjang, ketupat dan lain-lain. Simbol tersebut digunakan tergantung dengan fungsinya, karena berbeda bentuk berbeda juga fungsinya. Salah satu fungsinya yaitu jajar genjang untuk menggambarkan proses *input* dan *output*.

