

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Beberapa penelitian yang terkait dengan penelitian ini diantaranya oleh Prakoso (2015) dengan judul “*Sistem Pendukung Keputusan Untuk Menentukan Kelayakan Polisi Dalam Memegang Senjata Api Menggunakan Scoring System*”. Hasil pada tahapan rancangan sistem pendukung keputusan ini adalah, aplikasi ini dapat menentukan kelayakan polisi dalam memegang senjata api berdasarkan kriteria beserta bobot nilai yang telah ditentukan. Serta hasil Penilaian kelayakan polisi dalam memegang senjata api dapat digunakan untuk mengetahui para calon polisi mana yang layak memegang senjata api sesuai dengan kriteria POLRESTA Tanjungpinang.

Wasana (2014) dengan judul “*Sistem Pendukung Keputusan Kelayakan Pengajuan Kredit Sepeda Motor Menggunakan Metode Scoring System*”. Pada hasil sistem pendukung keputusan ini adalah, aplikasi yang dibuat dapat mendukung bagian kredit komite, terutama dalam melakukan penilaian kelayakan kredit dengan kriteria beserta bobot nilai yang telah ditentukan. Kemudian aplikasi yang dibuat dapat memantau perkembangan penilaian kelayakan kredit dari awal pengajuan sampai selesai penilaian kelayakan. Serta hasil Penilaian kelayakan kredit dapat digunakan untuk mengetahui para calon pemohon kredit mana yang layak diterima menjadi nasabah sesuai dengan kriteria PT. HD *FINANCE*.

Berdasarkan penelitian terdahulu dengan penelitian sekarang adalah pada penelitian ini terdapat persamaan dan perbedaan. Dimana persamaannya adalah menggunakan kriteria sebagai tolak ukur untuk menentukan hasil keputusan berdasarkan bobot nilai yang telah ditentukan. Kemudian dari perbedaan dari penelitian terdahulu dan penelitian sekarang adalah penelitian terdahulu sistem yang dibuat menggunakan Bahasa pemrograman *VB.Net*. Sedangkan penelitian sekarang menggunakan bahasa pemrograman berbasis *website*.

2.2 Sistem Pendukung Keputusan

2.2.1 Pengertian Sistem Pendukung Keputusan

Menurut Turban (2005), penelitian tentang Sistem Pendukung Keputusan (SPK) merupakan suatu sistem interaktif yang membantu pengambilan keputusan melalui penggunaan data dan model-model

keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur.

2.2.2 Pengertian Sistem

Menurut Jogiyanto (2005, 2), mengemukakan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi. Dengan demikian sistem merupakan kumpulan dari beberapa bagian yang memiliki keterkaitan dan saling bekerja sama serta membentuk suatu kesatuan untuk mencapai suatu tujuan dari sistem tersebut. maksud dari suatu sistem adalah untuk mencapai suatu tujuan dan sasaran dalam ruang lingkup yang sempit.

2.3 Waralaba

Menurut Munir (2005), menyatakan bahwa *Franchise* atau sering disebut juga dengan istilah waralaba adalah suatu cara melakukan kerjasama di bidang bisnis antara 2 (dua) atau lebih perusahaan, di mana 1 (satu) pihak akan bertindak sebagai *franchisor* dan pihak yang lain sebagai *franchisee*, di mana di dalamnya diatur bahwa pihak-pihak franchisor sebagai pemilik suatu merek dari know-how terkenal, memberikan hak kepada *franchisee* untuk melakukan kegiatan bisnis dari atas suatu produk barang atau jasa, berdasar dan sesuai rencana komersil yang telah dipersiapkan, diuji keberhasilannya dan diperbaharui dari waktu ke waktu, baik atas dasar hubungan yang eksklusif ataupun noneksklusif, dan sebaliknya suatu imbalan tertentu akan dibayarkan kepada *franchisor* sehubungan dengan hal tersebut.

2.4 Lokasi Strategis

Lokasi strategis menurut Lupiyoadi (2009), berhubungan dengan di mana perusahaan harus bermarkas dan melakukan operasi atau kegiatannya. Dalam hal ini ada tiga jenis interaksi yang mempengaruhi lokasi, yaitu:

1. Konsumen mendatangi pemberi jasa (perusahaan): apabila keadaannya seperti ini maka lokasi menjadi sangat penting.

- Perusahaan sebaiknya memilih tempat dekat dengan konsumen sehingga mudah dijangkau, dengan kata lain harus strategis.
2. Pemberi jasa mendatangi konsumen: dalam hal ini lokasi tidak terlalu penting, tetapi yang harus diperhatikan adalah penyampaian jasa harus tetap berkualitas.
 3. Pemberi jasa dan konsumen tidak bertemu secara langsung: berarti penyedia jasa dan konsumen berinteraksi melalui sarana tertentu seperti telepon, komputer, atau surat. Dalam hal ini lokasi menjadi sangat tidak penting selama komunikasi antara kedua pihak terlaksana dengan baik.

2.5 Scoring System

Scoring system disebut juga sebagai skor skala, memerlukan suatu norma pembandingan agar dapat diinterpretasikan secara kualitatif. Pada dasarnya interpretasi skor skala selalu bersifat normative, artinya makna skor diacukan pada posisi relatif skor dalam suatu kelompok yang telah dibatasi terlebih dahulu.

Hal ini dapat dilakukan dengan bantuan statistic deskriptif dari distribusi data skor kelompok yang umumnya mencakup banyaknya subjek (n) dalam suatu kelompok, mean skor skala (M), deviasi standard skor skala (s) dan varians (s^2), skor minimum (X_{min}) dan maksimum (X_{max}), dan statistic - statistik lain yang dirasa perlu. Deskripsi data ini memberikan gambaran penting mengenai keadaan distribusi skor skala pada kelompok subjek yang dikenai pengukuran dan berfungsi sebagai sumber informasi mengenai keadaan subjek pada aspek variable yang diteliti.

Sedangkan untuk menerapkan metode Scoring System ke dalam sistem, suatu skor yang ditentukan melalui prosedur penskalaan akan menghasilkan angka-angka pada level pengukuran interval dan interpretasikan hanya dapat dihasilkan kategori-kategori atau kelompok-kelompok skor pada level ordinal. Skor-skor mentah (*raw score*) yang dihasilkan suatu skala merupakan penjumlahan dari skor item-item dalam skala itu.

Relativitas hasil pengukuran selalu membawa permasalahan mengenai cara-cara pengelompokan (kategorisasi) apabila diperlukan pemisahan subjek ke dalam kelompok diagnosis yang berbeda. Kategori ini didasari oleh suatu asumsi bahwa skor subjek dalam kelompoknya merupakan estimasi dalam skor subjek dalam populasi dan bahwa skor subjek dalam populasinya terdistribusi secara normal.

Kategori jenjang (ordinal) menurut Azwar (2003) kategori ini memiliki tujuan menempatkan individu ke dalam kelompok-kelompok terpisah secara berjenjang menurut suatu kontinum berdasar atribut yang diukur. Kontinum jenjang ini contohnya adalah dari rendah ke tinggi, dari paling jelek ke paling baik, dari sangat tidak puas ke sangat puas, dan sebagainya. Banyaknya jenjang kategori diagnosis yang akan dibuat biasanya tidak lebih dari lima jenjang tetapi juga tidak kurang dari tiga jenjang. Misalnya mengelompokkan individu individu kedalam hanya dua jenjang diagnosis saja, yaitu “semangat kerja rendah” dan “semangat kerja tinggi” akan mengakibatkan resiko kesalahan yang cukup besar bagi skor-skor yang terletak disekitar mean kelompok.

Keterangan:

X = skor atau nilai

μ = mean teoritis

σ = standar deviasi

Langkah-langkah penentuan kategorisasi berdasarkan jenjang (ordinal) adalah sebagai berikut :

1. Menentukan data statistik secara deskriptif berupa rentang minimum (Xmin), rentang maksimum (Xmax), luas jenjang sebaran, mean teoritis (σ) dan devisiasi standar (μ).
2. Menghitung data statistik secara deskriptif sebagai berikut :
 $Xmin = \text{banyaknya pertanyaan} * \text{nilai minimum}$
 $Xmax = \text{banyaknya pertanyaan} * \text{nilai maksimum}$
 $\text{Luas jarak sebaran} = Xmax - Xmin$
 $\sigma = \text{luas jarak sebaran} / 6$
 $\mu = \text{banyaknya pertanyaan} * \text{banyak kategori}$
3. Menghitung p dengan menggunakan tabel distribusi normal, terlebih dahulu menentukan Zmin dan Zmax dengan rumus :
 $Zmin = (Xmin - \mu) / \sigma$
 $Zmax = (Xmax - \mu) / \sigma$
4. Memilih p dengan nilai yang maksimal sehingga dapat ditemukan rentang skala prioritas dengan 3 kategori, yaitu :
 $X < (\mu - (p * \sigma))$ kategorinya rendah atau tidak layak
 $(\mu - (p * \sigma)) \leq X < (\mu + (p * \sigma))$ kategorinya sedang atau layak
 $(\mu + (p * \sigma)) \leq X$ kategorinya tinggi atau sangat layak.

2.6 Basis Data

2.6.1 Pengertian Basis Data

Menurut Pakereng & Wahyono (2004), basis data merupakan kumpulan data yang dipakai ada dalam suatu lingkup tertentu, misalkan instansi, perusahaan, dan lain-lain atau kasus tertentu. Menurut Pakereng & Wahyono, sebuah konsep database memiliki beberapa hal sebagai berikut:

1. Entitas

Entitas merupakan tempat informasi direkam, dapat berupa orang, tempat, kejadian dan lain-lain. Sebagai contoh dalam kasus Administrasi Siswa misalnya, maka terdapat entitas siswa, matakuliah, guru, pembayaran.

2. Atribut

Atribut dapat juga disebut sebagai data elemen, data field, atau data item yang digunakan untuk menerangkan suatu entitas dan mempunyai harga tertentu, misalnya atribut dari entitas siswa diterangkan oleh nama, tanggal lahir, alamat.

3. Data *value*

Data *value* merupakan suatu informasi atau data aktual yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

4. *File/Table*

Merupakan kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

5. *Record/Tuple*

Merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

2.6.2 MySql

Menurut Nugroho (2004), MySQL (*My Structured Query Language*) adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*Database Management System*), sifat dari DBMS ini adalah *open source*. MySQL sebenarnya produk yang berjalan pada *platform* Linux, dengan adanya perkembangan dan banyaknya pengguna, serta lisensi dari database ini adalah *open source*, maka para pengembang merilis versi Windows.

MySQL merupakan program database yang mengakses datanya bersifat jaringan, sehingga dapat digunakan untuk aplikasi *Multi User* (banyak pengguna). Kelebihan lain dari MySQL adalah menggunakan bahasa *query* (permintaan) standar SQL (*Structural Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur, SQL telah di standartkan untuk semua program pengakses database seperti oracle, PostgreSQL, SQL Server dan lain-lain.

MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti PHP maupun yang tidak *open source* yang ada pada platform windows seperti Visual Basic, Delphi dan lainnya.

2.7 Hypertext Markup Language (HTML)

HTML dirancang untuk digunakan tanpa ketergantungan pada suatu platform tertentu. Dokumen HTML adalah suatu dokumen teks biasa, dan disebut *markup language* karena mengandung tag tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen. Pada dokumen HTML yang termasuk sistem hypertext, kita tidak harus membaca dokumen tersebut secara urut dari atas ke bawah atau sebaliknya, tetapi kita dapat menuju topik tertentu secara langsung dengan menggunakan teks penghubung yang akan membawa anda ke suatu topik atau dokumen lain secara langsung. (Setiawan, 2012).

2.8 Hypertext Preprocessor (PHP)

Menurut Arief (2011), PHP adalah sebuah bahasa pemrograman scripting untuk membuat halaman web yang dinamis. PHP dikatakan sebagai sebuah server-side embedded script language artinya sintak-sintak dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML yang seperti biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada tampilan web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

Terdapat beberapa pandangan dalam mengartikan kata PHP, kurang lebih dapat diartikan sebagai *Hypertext Preprocessor*. PHP merupakan bahasa pemrograman yang hanya dapat berjalan pada *server* dan hasilnya dapat ditampilkan pada *client*.

PHP merupakan produk *Open Source* yang dapat digunakan secara gratis tanpa harus membayar untuk menggunakannya. PHP merupakan

bahasa standar yang digunakan dalam dunia *Website*, PHP adalah bahasa pemrograman yang berbentuk skrip yang diletakkan di dalam *server web*.

2.9 XAMPP

Menurut Nugroho (2004), XAMPP merupakan paket PHP yang berbasis open source yang dikembangkan oleh sebuah komunitas open source. XAMPP merupakan suatu program yang di dalamnya terdapat beberapa paket program yang sudah dapat langsung dijalankan yaitu Apache, MYSQL, PHP, File Zila, Phpmyadmin dan lain-lain.

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Menurut Alan Nur Aditya (2011), fungsi dari XAMPP adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2.10 DFD (*Data Flow Diagram*)

Menurut Fatta (2008, 119), *Data Flow Diagram* (DFD) merupakan diagram yang digunakan untuk menambahkan proses-proses yang terjadi pada sistem yang akan dikembangkan. Dengan model ini, data-data yang terlibat pada masing-masing proses dapat diidentifikasi. Pengembangan DFD biasanya menggunakan cara berjenjang. Dimulai dari *Context Diagram*, DFD level 0, DFD level 1, DFD level 2, dan seterusnya sesuai dengan kompleksitas dari sistem yang akan dikembangkan. Berikut adalah penjelasan dari *Context Diagram*, DFD, DFD level 0, level 1, dan DFD level 2 menurut Fatta (2008, 109).

1. *Context Diagram*

DFD pertama dalam proses bisnis. Menunjukkan *context* dimana Proses bisnis berada. Menunjukkan semua proses bisnis dalam satu proses tunggal atau proses nol. *Context diagram* juga menunjukkan sebuah entitas luar yang menerima informasi dari atau memberikan informasi ke sistem.

2. Level 0 Diagram

Menunjukkan semua proses utama yang menyusun keseluruhan sistem. Level ini juga menunjukkan komponen *internal* dari proses dan menunjukkan bagaimana proses proses utama direalisasikan menggunakan *data flow*. Pada level ini juga ditunjukkan bagaimana proses-proses utama terhubung dengan entitas *eksternal*. Pada level ini juga dilakukan penambahan *data store*.

3. Level 1 Diagram

Umumnya diagram level 1 diciptakan dari setiap proses utama dari level 0. Level ini menunjukkan proses proses internal yang menyusun setiap proses-proses utama dalam level 0, sekaligus menunjukkan Bagaimana informasi berpindah dari satu proses ke proses yang lainnya. Jika misalnya proses induk dipecah, katakanlah menjadi tiga proses anak, maka tiga proses anak ini secara utuh menyusun proses induk.

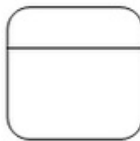
4. Level 2 Diagram

Menunjukkan semua proses yang menyusun sebuah proses pada level 1. Bisa saja penyusunan DFD tidak mencapai level 2 ini. Atau mungkin harus dilanjutkan ke level berikutnya.

Menurut Fatta (2008, 107), ada empat elemen yang menyusun suatu DFD yaitu sebagai berikut:

1. Proses

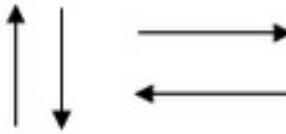
Aktivitas yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.



Gambar 2.1 Simbol Proses Menurut Gene dan Serson

2. *Data Flow*

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.



Gambar 2.2 Simbol *Data Flow* Menurut Gene dan Serson

3. *Data Store*

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam *data store*. Aliran data di *update* atau ditambahkan ke data store.



Gambar 2.3 Simbol *Data Store* Menurut Gene dan Serson

4. *External Entity*

Orang organisasi atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.



Gambar 2.4 Simbol *External Entity* Menurut Gene dan Serson

2.11 ERD (Entity Relationship Diagram)

Menurut Ladjamuddin (2005, 84), ERD (*Entity Relationship Diagram*) adalah diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis.

Simbol standar yang digunakan dalam membuat *Entity Relationship Diagram* diantaranya:

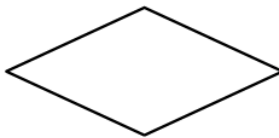
1. *Entity* / Entitas

Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan.



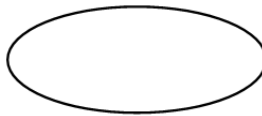
Gambar 2.5 Simbol *Entity* / Entitas

2. *Relationship* / Relasi
Hubungan yang dapat terjadi antara satu entitas atau lebih.



Gambar 2.6 Simbol *Relationship* / Relasi

3. *Attribut* / Atribut
Karakteristik entitas atau *relationship* yang menyediakan penjelasan detail entitas atau *relationship*.



Gambar 2.7 Simbol *Attribut* / Atribut

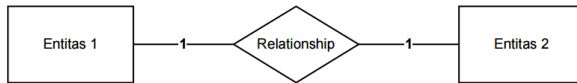
4. *Line* / Garis
Baris sebagai penghubung antara himpunan, relasi dan himpunan entitas dan atributnya.

Gambar 2.8 Simbol *Line* / Garis

Definisi kardinalitas relasi menurut Ladjamuddin (2005), Kardinalitas Relasi menunjukkan jumlah maksimum *tupel* yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat 3 macam kardinalitas relasi yaitu sebagai berikut:

1. Relasi *One to One*

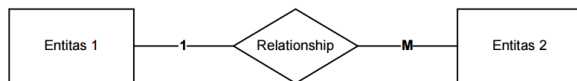
Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Gambar 2.9 Relasi *One to One*

2. Relasi *One to Many* atau *Many to One*

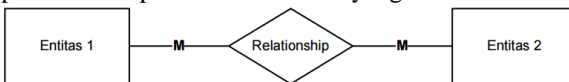
Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.



Gambar 2.10 Relasi *One to Many* atau *Many to One*

3. Relasi *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.



Gambar 2.11 Relasi *Many to Many*

2.12 Blackbox Testing

Menurut Simarmata (2010, 316), klasifikasi *black box testing* mencakup beberapa pengujian salah satunya adalah pengujian fungsional (*functional testing*). Pada jenis pengujian ini, perangkat lunak diuji untuk persyaratan fungsional. Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang

diharapkan. Walaupun pengujian fungsional sudah sering dilakukan di bagian akhir dari siklus pengembangan, masing-masing komponen dan proses dapat diuji pada awal pengembangan, bahkan sebelum sistem berfungsi, pengujian ini sudah dapat dilakukan pada seluruh sistem.

Pengujian fungsional meliputi seberapa baik sistem melaksanakan fungsinya, termasuk perintah-perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fungsional juga meliputi permukaan yang jelas dari jenis fungsi-fungsi, serta operasi *back-end* (seperti keamanan dan bagaimana meningkatkan sistem). Pengujian pada *black box* berusaha menemukan kesalahan seperti:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database eksternal*
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

2.13 Event List

Menurut Kristanto (2003, 64), *Event List* adalah daftar kejadian (*Event List*) yang digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan dan respon dari sistem. Suatu kejadian mewakili satu aliran data atau proses dalam diagram konteks serta deskripsi penyimpanan yang digunakan untuk memodelkan data harus diperhatikan dalam kaitannya dengan daftar kejadian.

Secara umum setiap aliran data dalam *event list* adalah kejadian atau event, tepatnya aliran data mengindikasikan terjadinya kejadian, atau aliran data dibutuhkan oleh sistem untuk melakukan proses. Aturan-aturan dalam *event list* antara lain daftar kejadian yang kita buat dan digambarkan dalam bentuk tekstual sederhana yang berfungsi memodelkan kejadian dalam lingkungan dimana sistem harus memberikan respon. Ketika membuat *event list*, maka kita harus yakin perbedaan antara kejadian (*event*) dan kejadian yang berelasi dengan aliran (*event-related flow*).