

BAB II

LANDASAN TEORI

2.1 Teori tentang Topik Skripsi

2.2.1 Rekrutmen

Rekrutmen adalah proses mencari dan menyeleksi calon karyawan untuk mengisi posisi atau jabatan tertentu. Kunci utama menciptakan Sumber Daya Manusia (SDM) yang profesional terletak pada proses rekrutmen, seleksi, *training*, dan *development* calon karyawan [3]. Tidaklah mudah mencari karyawan yang berkualitas. Proses rekrutmen ini penting dalam menentukan baik tidaknya pelamar yang melamar pekerjaan pada suatu perusahaan. Tujuan rekrutmen yaitu agar perusahaan dapat memiliki tenaga kerja yang benar-benar sesuai dengan kebutuhan, baik berdasarkan beban kerja yang ada maupun kebutuhan lain. Agar tujuan tersebut dapat dicapai dengan tepat, tak jarang rekrutmen pun harus melalui berbagai tahapan. Bagaimanapun juga, perusahaan tentu ingin memperoleh tenaga kerja yang mampu menyelesaikan tanggung jawabnya secara efektif dan efisien. Secara garis besar terdapat dua metode rekrutmen yang umum dilakukan suatu perusahaan, yaitu metode rekrutmen tertutup dan terbuka [3].

1. Metode Rekrutmen Tertutup

Metode rekrutmen tertutup adalah proses rekrutmen yang dilakukan dengan mencari calon karyawan dari internal perusahaan. Metode rekrutmen ini dapat dilakukan dengan cara vertikal (demosi) atau cara horizontal (rotasi). Cara vertikal bisa dilakukan dengan langkah demosi (penurunan jabatan). Langkah ini sangat jarang dilakukan oleh perusahaan, terkecuali karyawan tersebut sudah benar-benar tidak mampu mengerjakan beban tugas yang diberikan ataupun sudah melanggar peraturan perusahaan yang mengakibatkan terjadinya penurunan jabatan. Cara horizontal bisa dilakukan perusahaan dengan

langkah rotasi yaitu perpindahan jabatan pada level yang sama. Terdapat beberapa keuntungan dan kelemahan dalam pelaksanaan metode tertutup. Berikut keuntungan dan kekurangannya, antara lain:

2. Metode Rekrutmen Terbuka

Metode rekrutmen terbuka adalah rekrutmen perusahaan yang berasal dari luar lingkungan perusahaan (eksternal). Metode ini diperlukan perusahaan pada saat ekspansi bisnis karena kebutuhan tenaga kerja meningkat.

2.1.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (DSS) mendorong keputusan yang lebih cepat dan lebih cerdas berdasarkan data objektif, bukan berdasarkan kriteria subjektif atau naluri pribadi [4]. Mereka menawarkan wawasan dan tindakan yang diusulkan kepada pembuat keputusan berdasarkan diagnosis masalah, tindakan sebelumnya yang diambil, hasil dari tindakan tersebut dan info kontekstual relevan lainnya.

Sistem pendukung keputusan (DSS) adalah program terkomputerisasi yang digunakan untuk model bahan untuk mendukung penentuan, penilaian, dan tindakan pemilihan solusi dalam organisasi atau bisnis. DSS menyaring dan menganalisis data dalam jumlah besar, mengumpulkan informasi komprehensif yang dapat digunakan untuk memecahkan masalah dan dalam pengambilan keputusan. Pada saat ini organisasi bisnis dan organisasi pemerintahan telah mengaplikasikan berbagai sistem informasi berbasis komputasi dalam berbagai proses manajemennya sehingga menghasilkan gudang data (*Data Warehouse*) hal ini lazim disebut era Big Data.

Para pengambil keputusan biasanya menghadapi situasi yang ada yang memiliki berbagai karakteristik dalam memutuskan solusi terbaik yang pada umumnya berlatar belakang permasalahan yang bersifat unik, tidak pasti, kompleks, memiliki banyak variabel (struktur panjang, sedang, pendek), dan bertingkat. Secara hierarkis model bahan keputusan dalam organisasi bisnis memiliki level dan tingkat secara piramida mengerucut dari sisi operasional hingga tingkat manajerial [4].

2.2 Teori tentang Framework / Algoritma yang digunakan

2.2.1 TOPSIS

TOPSIS (Technique For Others Reference by Similarity to Ideal Solution) adalah salah satu metode pengambilan keputusan multi-kriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981) [5]. TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal.

Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai.

Semakin banyaknya faktor yang harus dipertimbangkan dalam proses pengambilan keputusan, maka semakin relatif sulit juga untuk mengambil keputusan terhadap suatu permasalahan. Apalagi jika upaya pengambilan keputusan dari suatu permasalahan tertentu, selain mempertimbangkan berbagai faktor/kriteria yang beragam, juga melibatkan beberapa orang pengambil keputusan. Permasalahan yang demikian dikenal dengan permasalahan *multiple criteria decision making* (MCDM). Dengan kata lain, MCDM juga dapat disebut sebagai suatu pengambilan keputusan untuk memilih alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Metode TOPSIS digunakan sebagai suatu upaya untuk menyelesaikan permasalahan *multiple criteria decision making*. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis [5].

2.2.2 *User Acceptance Testing*

UAT adalah salah satu metodologi paling inovatif untuk mencegah kegagalan proyek TI. Dalam pengembangan perangkat lunak, ada tiga hal yang dilakukan dalam proses UAT [6], di antaranya:

1. UAT mengekspos logika / fungsionalitas bisnis yang belum ditemukan, karena unit pengujian dan pengujian sistem tidak fokus pada fungsionalitas / logika bisnis
2. UAT mengukur bagaimana sistem sesuai dengan kebutuhan pengguna
3. UAT membatasi bagaimana sistem telah selesai.

Proses UAT dimulai dengan menyediakan dokumentasi kebutuhan bisnis, kemudian dilanjutkan dengan proses bisnis (alur kerja) atau skenario dan akhirnya pengujian menggunakan data. Keefektifan dalam pengujian diperlukan dalam pengembangan suatu aplikasi atau sistem informasi sehingga produk dapat sampai kepada pengguna tepat waktu dan sesuai dengan kebutuhan pengguna. Pengujian yang dilakukan pada sistem yang dibangun dibagi menjadi dua pengujian mekanisme yaitu pengujian alpha menggunakan metode *black box* dan pengujian beta menggunakan skala Likert. Pengujian *black box* difokuskan pada aktivitas pengujian sistem yang dibangun untuk melihat sejauh mana setiap fungsi dalam aplikasi telah berjalan benar, sesuai harapan dan bebas dari kesalahan. Selain menguji aplikasi dari sisi fungsional, pengujian sistem juga dilakukan oleh pengguna yang akan mengambil keuntungan dari aplikasi. Pengujian beta untuk menentukan penerimaan penggunaan aplikasi menggunakan dilakukan dengan kuesioner untuk mendapatkan kesimpulan tentang penerimaan penggunaan aplikasi dari sisi pengguna [6].

2.2.3 **SDLC**

Sistem yang sedang berjalan atau sedang digunakan oleh organisasi atau perusahaan akan terus dikembangkan untuk ataupun memperbaiki kekurangan-kekurangan pada sistem tersebut [7]. Untuk melakukan pengembangan sistem, metode yang digunakan adalah SDLC. Metode

adalah tahap-tahap ataupun aturan untuk melakukan sesuatu. Demikian pula didefinisikan oleh penulis lain mengatakan metode adalah cara untuk melakukan sesuatu. SDLC adalah sebuah metode yang digunakan untuk mengembangkan sebuah sistem. SDLC adalah sebuah SDLC proses logika yang digunakan oleh seorang *system analyst* logika ya untuk mengembangkan sebuah sistem informasi yang melibatkan *requirements, validation, training* dan pemilik sebuah sistem. SDLC identik dengan teknik pengembangan sistem *waterfall*, karena tahapannya menurun dari atas ke bawah. Berikut adalah tahapan dari SDLC [7]:

1. *Planning*

Pada tahap ini bertujuan untuk mengidentifikasi dan memprioritaskan sistem informasi apa yang akan dikembangkan, sasaran-sasaran yang ingin dicapai, jangka waktu pelaksanaan serta mempertimbangkan dana yang tersedia dan siapa yang melaksanakan.

2. *Analysis*

Tahap Analisa sistem merupakan kegiatan penguraian dari suatu sistem informasi yang utuh ke dalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

3. *Design*

Desain berkonsentrasi pada bagaimana sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Manfaat desain sistem adalah memberikan gambaran rancang bangun (*blue print*) yang lengkap, sebagai penuntun (*guideline*) bagi *programmer* dalam membuat aplikasi.

4. *Implementation*

Tahap berikutnya adalah implementasi yaitu mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba.

2.3 Teori tentang Tools / Software yang digunakan

2.3.1 Database

Basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai suatu markas atau gudang, tempat bersarang atau tempat berkumpul. Data dapat diartikan merupakan representasi dari fakta dunia yang mewakili suatu obyek (manusia, barang, peristiwa, keadaan dsb.) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Basis data adalah merupakan suatu sistem yang dibangun oleh beberapa komponen diantaranya ada enam komponen pokok [8] :

- 1 Perangkat keras (*hardware*) dalam sistem komputer. Dalam sistem pengolahan basis data digital perangkat utama sebagai pengolah data adalah komputer.
- 2 Perangkat lunak aplikasi (*software*) lain yang mendukung dan bersifat opsional. Perangkat lunak digunakan untuk mendukung proses pengelolaan basis data. Misal: bahasa pemrograman C, *basic pascal*.
- 3 Sistem operasi (*operating system*). Sistem operasi merupakan perangkat lunak yang digunakan untuk mengelola aplikasi basis data dan penggunaan sumber daya komputer.
- 4 Basis data lain yang mempunyai keterkaitan dan hubungan dengan basis data itu sendiri. Berisi atau memiliki objek-objek basis data seperti *file, table, index* . Mempunyai definisi struktur baik untuk basis data maupun objek-objek secara detail.
- 5 Sistem pengelola basis data atau *database management system* (DBMS). Merupakan program aplikasi untuk pengelolaan basis data, seperti *Microsoft Access, Oracle* dan lain-lain
6. Pemakai (*user*), yaitu pengguna yang terlibat dalam pengelolaan basis dan penggunaan basis data.

Database adalah suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (*controlled redundancy*) dengan cara tertentu sehingga mudah digunakan atau ditampilkan kembali, dapat digunakan oleh satu atau lebih program aplikasi secara optimal; data

disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya; data disimpan sedemikian rupa sehingga penambahan, pengambilan, dan modifikasi dapat dilakukan dengan mudah dan terkontrol [27].

2.3.2 *Framework*

Framework adalah sebuah *software* untuk memudahkan para *programmer* untuk membuat sebuah aplikasi *web* yang di dalamnya ada berbagai fungsi di antaranya *plugin*, dan konsep untuk membentuk suatu sistem tertentu agar tersusun dan terstruktur dengan rapi. Dengan menggunakan *framework* bukan berarti pengkodean tidak perlu dilakukan, karena pengguna *framework* haruslah menggunakan fungsi-fungsi dan variabel yang ada di dalam sebuah *framework* yang digunakan [9].

Secara sederhana bisa dijelaskan bahwa *framework* adalah kumpulan fungsi (*libraries*) sehingga seorang *programmer* tidak perlu lagi membuat fungsi-fungsi dari awal dan biasanya disebut kumpulan *library*. *Programmer* cukup memanggil kumpulan *library* atau fungsi yang sudah ada di dalam *framework* yang sudah pasti cara menggunakan fungsi-fungsi itu sudah ditentukan sesuai aturan masing-masing. Beberapa contoh fungsi-fungsi stkitart yang telah tersedia dalam suatu *framework* adalah fungsi *paging*, enkripsi, *email*, SEO, *session*, *security*, kalender, bahasa, manipulasi gambar, grafik, tabel bergaya zebra, validasi, *upload*, *captcha*, proteksi terhadap XSS (*XSS filtering*), *template*, kompresi, XML, dan lain-lain. Contoh dari *framework* PHP adalah *laravel*, *phalcon*, *slim*, *cakePHP*, *code igniter* (CI), *symphony*, *zend*, *yii*, dan *kohana*. Sedangkan, *framework javascript* juga ada, yakni: *Jquery* dan *mootools*. *Framework* untuk *ruby* adalah *ruby on rails* (ROR). Dengan menggunakan *framework*, sebuah aplikasi akan tersusun secara terstruktur dan rapi karena pasti si pembuat *framework* telah menggunakan *pattern stkitart*, misalnya MVC atau sering disebut *model-view-controller* yang telah terkenal di kalangan *programmer* PHP. Logikanya seperti ini, jika *website* dibuat dari 0, biasanya kode yang dibuat dan dipakai juga aka nasal-asalan sehingga di dalam 1 *file* akan

terdapat beribu-ribu kode yang sebenarnya dapat dipisah menjadi beberapa *file* sehingga performa *web* itu tidak menurun [28].

2.3.3 *Hypertext Preprocessor*

PHP (*Hypertext Preprocessor*) adalah bahasa *script* pemrograman yang dapat ditanamkan atau disisipkan ke dalam HTML, PHP banyak dipakai untuk memrogram situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS [10]. PHP merupakan bahasa pemrograman untuk membuat aplikasi berbasis *website*, PHP juga sering dikenal dengan *hypertext preprocessor* yang merupakan sebuah bahasa *scripting* tingkat tinggi yang dipasang pada dokumen HTML [11].

2.3.4 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *general public license* (GPL). Tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan *general public license* (GPL) [10].

MySQL adalah sistem yang berguna untuk melakukan proses pengaturan koleksi-koleksi struktur data (*database*) baik yang meliputi proses pembuatan atau proses pengelolaan *database*. MySQL ini bersifat *open source*, artinya setiap orang dimungkinkan untuk menggunakan dan memodifikasinya. Program ini dapat diperoleh secara gratis pada alamat : <http://www.mysql.com>. Sebenarnya MySQL ini suatu produk yang berjalan pada *linux*, tetapi seiring berjalannya waktu dan pengetahuan maka para pengembang kemudian merilis MySQL yang bisa diakses melalui *windows* [29].

2.3.5 PHPMYAdmin

PHPMYAdmin adalah aplikasi manajemen *database server* MySQL berbasis *web*. Dengan aplikasi *phpMyAdmin* *database* dapat diolah sebagai *root* atau juga sebagai *user* biasa, *database* baru bias dibuat, mengelola *database* dan melakukan operasi perintah-perintah *database* secara lengkap seperti di *MySQL Prompt*. *PhpMyAdmin* adalah *tools* yang dapat digunakan dengan mudah untuk manajemen *database* MySQL secara *visual* dan *Server* MySQL, alhasil tidak perlu lagi harus menulis *query* SQL setiap akan melakukan perintah operasi *database*. *Tools* ini cukup populer, Anda dapat mendapatkan fasilitas ini ketika menginstal paket *triad phpMyAdmin*, karena termasuk dalam XAMPP yang sudah dipasang [12].

PhpMyAdmin merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL [30].

2.3.6 UML (Unified Modeling Language)

UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem [13]. UML singkatan dari (*unified modeling language*) yang berarti bahasa pemodelan standar. UML memiliki sintaks dan semantik. Ketika membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang dibuat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekadar diagram, tetapi juga menceritakan konteksnya [14].

2.3.7 Visual Studio Code

Visual studio adalah *software* untuk mengembangkan aplikasi. Itu artinya, kalau ingin membuat program, *software*, aplikasi, dan bahkan *mobile app*, maka Anda bisa menggunakan *visual studio*. Dengan menggunakan *visual studio* ini, Anda bisa menulis kode-kode program,

menjalankan kode program, melakukan pengujian, *debugging*, mengemas menjadi aplikasi mandiri, dan banyak lagi. Jadi, dapat diibaratkan dengan sebutan yang lebih mudah, *visual studio* adalah pabrik untuk pembuatan *software*. Dari sini muncul istilah yang disebut IDE yang merupakan singkatan dari *integrated development environment*. Jika diterjemahkan secara bebas, maka *visual studio* adalah *software* yang menyediakan lingkungan bagi pengembangan aplikasi yang terintegrasi dari hulu ke hilir. *Visual studio* sendiri dibuat oleh *Microsoft* sehingga berorientasi pada MS *Windows* [15].

Visual studio adalah IDE (*integrated development environment*) yang dapat Anda gunakan untuk mengembangkan aplikasi-aplikasi *Windows*. *Visual studio* dirancang untuk fokus pada produktivitas. *Tool* ini disebut juga *rapid application development tools* (*RAD tools*) karena dirancang dan dilengkapi untuk meningkatkan produktivitas. Versi baru dari *visual studio* ini mudah digunakan dan mudah dipelajari. Fitur-fitur dalam *visual studio* versi terbaru dibuat lebih sederhana untuk mempermudah pengguna dalam mempelajarinya dan memenuhi kebutuhan para *programmer* [31].


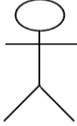



2.3.8 XAMPP

XAMPP merupakan suatu *software* yang bersifat *open source* yang merupakan pengembangan dari *LAMP* (*Linux, Apache, MySQL, PHP* dan *Perl*) [16]. *XAMPP* adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache* HTTP Server, *MySQL* database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* sendiri merupakan singkatan dari X (empat sistem operasi apa pun), *Apache*, *MySQL*, *PHP* dan *Perl* [32].

2.3.9 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut Aktor dan *use case* [17].

Tabel 2.1 *Use Case Diagram*


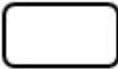




| Simbol | Keterangan |
|--|---|
| <p><i>Use Case</i></p>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> . |
| <p><i>Actor</i></p>  | Orang, proses, 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 <i>frase</i> nama aktor. |
| <p><i>Asosiasi</i></p>  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| <p><i>Generalisasi</i></p>  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> , Berbagi struktur data dari objek dan perilaku antara hubungan objek anak yang ada di atasnya objek induk (<i>ancestor</i>). |
| <p><i>Extend</i></p>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama bisa disisipkan ke dalam elemen yang ada di garis dengan panah. |

Sumber : [17]

2.3.10 Activity Diagram

Activity Diagram adalah Alur kerja, proses, logika dan hubungan antara aktor dengan alur kerja dalam *use case* digambarkan dalam *diagram* yang disebut dengan *diagram* aktivitas atau *activity diagram* [7]. Aktivitas

atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak [17].

| Simbol | Nama | Keterangan |
|---|------------------------|--|
|  | Status awal | Sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan / Decision | Percabangan dimana ada pilihan aktivitas yang lebih dari satu. |
|  | Penggabungan / Join | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu. |
|  | Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
|  | Swimlane | Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Sumber : [17]

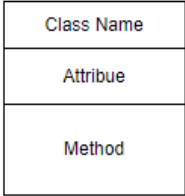





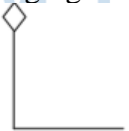
2.3.11 Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [17]. Kelas memiliki apa yang disebut atribut pola dan metode atau operasi ;

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas.

Class Diagram berfungsi untuk memberikan gambaran antara perbedaan yang sangat mendasar dan membawa hubungan antar *class* yang terletak di mana subsistem *class* tersebut [25].

Tabel 2.3 *Class Diagram*

| Simbol | Keterangan |
|--|--|
| <p><i>Class</i></p>  | <p>Kelas pada struktur sistem dan Gabungan tujuan dari berbagai atribut yang saling beroperasi.</p> |
| <p>Antarmuka/<i>interface</i></p>  | <p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p> |
| <p>Asosiasi/<i>association</i></p>  | <p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p> |
| <p>Asosiasi berarah</p>  | <p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p> |
| <p>Generalisasi</p>  | <p>Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus) antara objek anak dengan tingkah laku dan struktur data objek ada diatas objek induk.</p> |
| <p>Ketergantungan</p>  | <p>Relasi antar kelas dengan makna ketergantungan antar kelas dalam suatu komponen mandiri (<i>indepdent</i>) dan mempengaruhi komponen yang tidak bisa berdiri sendiri.</p> |
| <p>Agregasi</p>  | <p>Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).</p> |

Sumber : [17]

2.3.12 Black Box Testing

Black box testing adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “Kotak Hitam” yang tidak penting dilihat isinya, tapi cukup dikenali proses *testing* di bagian luar. Jenis *testing* ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan. Sebagai contoh, jika terdapat sebuah perangkat lunak yang merupakan sebuah sistem informasi *inventory* di sebuah perusahaan. Maka pada jenis *white box testing*, perangkat lunak tersebut akan berusaha dibongkar *listing* programnya untuk kemudian dites menggunakan teknik-teknik yang telah dijelaskan sebelumnya. Sedangkan pada jenis *black box testing*, perangkat lunak tersebut akan dieksekusi kemudian berusaha dites apakah telah memenuhi kebutuhan pengguna yang didefinisikan pada saat awal tanpa harus membongkar *listing* programnya [26].

2.4 Penelitian Terdahulu

Berikut adalah beberapa penelitian terdahulu yang menjadi landasan teori dalam penelitian ini yang dapat dilihat pada Tabel 2.1 di bawah ini:

Tabel 2.4 Tabel Literature Review

| Penelitian 1 | |
|---------------------|--|
| Judul | Sistem Pendukung Keputusan Pelanggan Terbaik Dan Pemberian Diskon Menggunakan Metode SAW & TOPSIS [20] |
| Nama Jurnal | Jurnal Ilmiah Elektronika Dan Komputer, Vol. 13, No. 1 |
| Nama Penulis | Taufik Kurnialensya, Rohmad Abidin |
| Tahun | 2020 |
| Metode | SAW dan TOPSIS |
| Hasil | 1. Dengan menggunakan bahasa pemrograman PHP dapat digunakan untuk membantu pihak perusahaan untuk memilih pelanggan terbaik dan penentuan diskon secara valid. 2. Aplikasi yang telah dibuat dapat digunakan sebagai alat bantu bagi pengambil keputusan dengan tetap berbasis pada sistem pendukung keputusan lebih efektif dalam pemilihan pelanggan |

| | |
|---------------------|--|
| | terbaik dan penentuan diskon menggunakan metode SAW (<i>Simple Additive Weighting</i>) dan Topsis (<i>Technique for Order Preference by Similarity to Ideal Solution</i>). |
| Penelitian 2 | |
| Judul | Penerapan Metode TOPSIS Untuk Sistem Pendukung Keputusan Penentuan Keluarga Miskin Pada Desa Panca Karsa II [21] |
| Nama Jurnal | ILKOM Jurnal Ilmiah , Vol. 9, No. 3 |
| Nama Penulis | Irvan Muzakkir |
| Tahun | 2017 |
| Metode | TOPSIS |
| Hasil | <ol style="list-style-type: none"> 1. Sistem yang digunakan pada mampu mengatasi kelemahan-kelemahan yang terdapat pada sistem yang lama dan memberikan hasil yang akurat dalam penentuan Keluarga Miskin 2. Penerapan Metode TOPSIS untuk sistem pendukung keputusan dapat memberikan hasil yang maksimal dalam hal pengambilan keputusan dengan cara mengurutkan alternatif masyarakat miskin mulai dari yang termiskin. 3. Berdasarkan hasil pengujian <i>white box</i> disimpulkan bahwa sistem pendukung keputusan ini bebas dari kesalahan program dengan total <i>Cyclomatic Complexity</i> = 6, <i>Region</i> = 6, dan <i>Independent Path</i> = 6. |
| Penelitian 3 | |
| Judul | Sistem Pendukung Keputusan Perekrutan Karyawan Menggunakan Metode TOPSIS di PT Visionet Data Internasional [22] |
| Nama Jurnal | Jurnal Informatika, Vol. 7, No. 2 |
| Nama Penulis | Ramos Somya, Andre Wahyudi |
| Tahun | 2020 |
| Metode | TOPSIS |
| Hasil | Berdasarkan hasil penelitian dan pengujian yang dilakukan, maka dapat disimpulkan bahwa sistem pendukung keputusan perekrutan karyawan pada PT Visionet Data Internasional dapat membantu perusahaan dalam menyediakan alternatif pengambilan keputusan saat perekrutan karyawan baru. Sistem bekerja dengan mengolah data calon karyawan baru berdasarkan kriteria dengan menentukan alternatif terbaik menggunakan metode TOPSIS. Sistem yang dibangun juga dilengkapi dengan menu riwayat yang akan memudahkan Admin HRD dalam melihat kembali data perhitungan perekrutan karyawan yang sudah pernah dilakukan. Adapun saran yang diperlukan untuk sistem ini adalah mengenai tampilan sistem, |

| | |
|---------------------|---|
| | penambahan otentikasi keamanan dan penambahan atau pembaharuan fungsi yang ada untuk mengoptimalkan kerja dari sistem. |
| Penelitian 4 | |
| Judul | Metode TOPSIS Dalam Sistem Pendukung Keputusan Penentuan Penerimaan Beasiswa Di STMIK Pringsewu [24] |
| Nama Jurnal | Jurnal Manajemen Sistem Informasi dan Teknologi, Vol. 9, No. 1 |
| Nama Penulis | Riki Renaldo, Elisabet Yunaeti Anggraeni, Elieser Rudi HC |
| Tahun | 2019 |
| Metode | TOPSIS |
| Hasil | Metode Topsis Dalam Sistem Pendukung Keputusan Penentuan Beasiswa di STMIK Pringsewu dapat diterapkan untuk membantu dan menentukan penerimaan beasiswa pada STMIK Pringsewu. |
| Penelitian 5 | |
| Judul | Implementasi Metode Entropy dan Topsis Dalam Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik [33] |
| Nama Jurnal | Ultima Computing : Jurnal Sistem Komputer ,Vol 10 ,No. 1 |
| Nama Penulis | Mitha Anggreani Rupang , Adhi Kusnadi |
| Tahun | 2018 |
| Metode | Entropy dan TOPSIS |
| Hasil | Pegawai peringkat tertinggi akan direkomendasikan oleh manajer untuk mendapatkan penghargaan dari perusahaan. Indeks tingkat kepuasan responden terhadap sistem pendukung keputusan berkisar antara 70% -80%. |
| Penelitian 6 | |
| Judul | Decision Support System for Home Selection in South Tangerang City Using TOPSIS Method [34] |
| Nama Jurnal | IJNMT (INTERNATIONAL JOURNAL OF NEW MEDIA TECHNOLOGY), Vol 2 , No 20 |
| Nama Penulis | Haris Widjaja , Ririn Ikana Desanti |
| Tahun | 2020 |
| Metode | TOPSIS |
| Hasil | Penelitian menggunakan metode perhitungan TOPSIS yang memberikan hasil dengan solusi ideal sesuai dengan kriteria yang dipilih. Aplikasi berbasis web ini dapat digunakan oleh agen properti untuk membantu mereka dalam menjual rumah dan oleh |

| | |
|--|--|
| | pengguna dalam hal memberikan rekomendasi rumah yang sesuai dengan kriteria. |
|--|--|

Pada penelitian 1, aplikasi yang telah dibuat dapat digunakan sebagai alat bantu bagi pengambil keputusan dengan tetap berbasis pada sistem pendukung keputusan lebih efektif dalam pemilihan pelanggan terbaik dan penentuan diskon menggunakan metode SAW (*Simple Additive Weighting*) dan Topsis (*Technique for Order Preference by Similarity to Ideal Solution*). Pada penelitian 2, Penerapan Metode TOPSIS untuk sistem pendukung keputusan dapat memberikan hasil yang maksimal dalam hal pengambilan keputusan. Pada penelitian 3, sistem pendukung keputusan perekrutan karyawan pada PT Visionet Data Internasional menggunakan metode TOPSIS dapat membantu perusahaan dalam menyediakan alternatif pengambilan keputusan saat perekrutan karyawan baru. Pada penelitian 4, Metode Topsis Dalam Sistem Pendukung Keputusan Penentuan Beasiswa di STMIK Pringsewu dapat diterapkan untuk membantu dan menentukan penerimaan beasiswa pada STMIK Pringsewu. Pada penelitian 5, Indeks tingkat kepuasan responden terhadap sistem pendukung keputusan menggunakan metode TOPSIS berkisar antara 70% -80%. Pada penelitian 6, metode perhitungan TOPSIS yang memberikan hasil dengan solusi ideal sesuai dengan kriteria yang dipilih.

Berdasarkan dari perbandingan jurnal yang dijabarkan di atas, maka akan digunakan metode TOPSIS untuk sistem pendukung keputusan, karena hasil metode TOPSIS cukup baik dari segi hasil dan bermanfaat pada kasus-kasus yang terdapat pada jurnal di atas. Berikut ini adalah perbandingan kekurangan metode Topsis dengan metode yang sejenis:

Tabel 2.4 Tabel Perbandingan Kekurangan Metode Topsis Dengan Metode Yang Sejenis

| Fuzzy Logics | Technique for Order of Preference by Similarity to Ideal Solution | Analytical Hierarchy Process | Simple Additive Weighting |
|---------------------|--|---|----------------------------------|
| | Pada metode TOPSIS, seringkali digunakan asumsi | Ketergantungan model AHP pada input utamanya. | Pembuat keputusan |

| | | | |
|--|---|---|--|
| Belum adanya metode umum untuk mengembangkan dan implementasi pengendali fuzzy. | pada tingkat kepentingan masing-masing respon dan digunakan kombinasi dengan metode lain untuk menyelesaikan asumsi tersebut. | Input utama ini berupa persepsi seorang ahli sehingga dalam hal ini melibatkan subyektifitas sang ahli selain itu juga model menjadi tidak berarti jika ahli tersebut memberikan penilaian yang keliru. | harus membuat matriks keputusan. |
| Susah untuk menentukan kemiringan pada fungsi tertentu dan menentukan aturan fuzzy | | Orang yang dilibatkan adalah orang-orang yang memiliki pengetahuan ataupun banyak pengalaman yang berhubungan dengan hal yang akan dipilih dengan menggunakan metode AHP | Pembuat keputusan harus menentukan bobot di tiap atribut. |
| Sulit dalam menentukan fuzzy set membership function-nya | Belum adanya bentuk untuk penilaian alternatif terhadap kriteria, biasanya bentuk ini diinterpretasikan dalam sebuah bilangan fuzzy | Untuk melakukan perbaikan keputusan, harus di mulai lagi dari tahap awal. | Pembuat keputusan harus menginput data dengan tepat dan benar sehingga tidak terjadi kesalahan dalam pembobotan serta perangkian kriteria. |
| Para engineer dan ilmuwan masih banyak | Metode TOPSIS menentukan solusi berdasarkan jarak | Metode AHP ini hanya metode matematis tanpa ada | Kriteria yang ditetapkan harus |

| | | | |
|---|---|---|--|
| <p>yang tidak mengenal teori kendali fuzzy, meskipun secara praktis mereka memiliki pengalaman untuk menggunakan teknologi dan perkakas yang sudah ada.</p> | <p>terpendek menuju solusi ideal dan jarak terbesar dari solusi yang ideal. Namun, metode ini tidak mempertimbangkan kepentingan (relative importance) dari masing-masing jarak tersebut.</p> | <p>pengujian secara sehingga tidak ada batas kepercayaan dari kebenaran model yang terbentuk.</p> | <p>memiliki cakupan yang luas dan dinamis.</p> |
| <p>Belum banyak terdapat kursus/balai 24 dan buku-buku teks yang menjangkau setiap tingkat (undergraduate, postgraduate, dan on site training) Hingga kini belum ada pengetahuan sistematis yang baku dan seragam tentang metodologi pemecahan problema kendali menggunakan pengendali fuzzy.</p> | <p>Pada metode TOPSIS, alternatif dengan ranking tertinggi merupakan solusi yang terbaik, namun belum tentu ranking tertinggi tersebut adalah yang terdekat dari solusi ideal. Sehingga perlu dilakukan perhitungan lagi untuk memastikannya.</p> | | |

Sumber: [36] [37]

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A