

BAB II

LANDASAN TEORI

2.1. Sistem Pendukung Keputusan (SPK)

SPK atau *Decision Support System (DSS)* merupakan suatu sistem yang menyediakan fasilitas untuk melakukan suatu analisis sehingga setiap proses pengambilan keputusan yang dilakukan oleh para pelaku bisnis akan lebih berkualitas dengan melihat keadaan bisnis yang sedang berjalan dan data-data dari luar perusahaan serta data-data privat dari pengambil keputusan (Sarwono, et al., 2018).

2.1.1. Metode *Profile Matching*

Profile Matching adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati (I. G. K. G. Puritan Wijaya Adh, 2017).

Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap). *Gap Analysis* merupakan suatu metode pengukuran untuk mengetahui kesenjangan (*gap*) antara kinerja suatu variabel dengan harapan konsumen terhadap variabel tersebut. *Gap (+)* positif akan diperoleh apabila skor persepsi lebih besar dari skor harapan, sedangkan apabila skor harapan lebih besar daripada skor persepsi akan diperoleh *gap (-)* negatif (Sari, 2018).

Berikut adalah beberapa tahapan dan perumusan perhitungan dengan metode *profile matching* (Pratiwi, 2016):

1. Pembobotan

Pada tahap ini, akan ditentukan bobot nilai masing-masing aspek dengan menggunakan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Dalam penentuan peringkat pada setiap *gap*, diberikan bobot nilai sesuai dengan Tabel 2.1.:

Tabel 2. 1. Bobot Nilai GAP

No	Selisih <i>gap</i>	Bobot Nilai	Keterangan
1	0	5	Kompetensi sesuai dengan yang dibutuhkan
2	1	4.5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kurang 1 tingkat/level
4	2	3.5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kurang 2 tingkat/level
6	3	2.5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kurang 3 tingkat/level
8	4	1.5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kurang 4 tingkat/level

2. Pengelompokan *Core dan Secondary Factor*

Setelah menentukan bobot nilai *gap* kriteria yang dibutuhkan, kemudian tiap kriteria dikelompokkan lagi menjadi dua kelompok yaitu *core factor* dan *secondary factor*.

- a) *Core factor* (faktor utama) merupakan aspek (kompetensi) yang paling menonjol/paling dibutuhkan oleh suatu jabatan yang

diperkirakan dapat menghasilkan kinerja optimal. Untuk menghitung *core factor* digunakan rumus seperti pada Rumus 2.1.

$$NCF = \frac{\Sigma NC}{\Sigma IC}$$

Rumus 2.1. Perhitungan rata-rata *core factor*

Keterangan:

NCF = Nilai rata – rata core factor

NC = Jumlah total nilai core factor

IC = Jumlah item core factor

- b) *Secondary factor* (faktor pendukung) adalah *item-item* selain aspek yang ada pada *core factor*. Untuk menghitung *secondary factor* digunakan rumus seperti pada Rumus 2.2.

$$NSF = \frac{\Sigma NS}{\Sigma IS}$$

Rumus 2.2. Perhitungan rata-rata *secondary factor*

Keterangan:

NSF = Nilai rata – rata secondary factor

NS = Jumlah total nilai secondary factor

IS = Jumlah item secondary factor

3. Perhitungan Nilai Total

Dari perhitungan *core factor* dan *secondary factor* dari tiap-tiap aspek, kemudian dihitung nilai total dari tiap-tiap aspek yang diperkirakan berpengaruh pada kinerja tiap-tiap *profile*. Untuk

menghitung nilai total dari masing-masing aspek, digunakan rumus seperti Rumus 2.3.

$$N = ((X)\% \cdot NCF) + ((X)\% \cdot NSF)$$

Rumus 2.3. Rumus Nilai Total Tiap Aspek

Keterangan:

N = Nilai total tiap aspek

NCF = Nilai core factor

NSF = Nilai secondary factor

$(X)\%$ = Nilai persentase

4. Perankingan

Hasil akhir dari proses *profile matching* adalah ranking dari kandidat yang diajukan untuk mengisi suatu jabatan/posisi tertentu. Adapun rumus perankingan dalam metode *profile matching* seperti Rumus 2.4.

$$Ranking = (x)\% \cdot Na + (x)\% \cdot Nb + \dots + (x)\% \cdot Nn$$

Rumus 2.4. Perhitungan ranking *profile matching*

Keterangan:

Na = Nilai factor a

Nb = Nilai factor b

Nn = Nilai factor ke $- n$

$(x)\%$ = Nilai persentase

2.1.2. Metode *Multi Factor Evaluation Process* (MFEP)

Multi Factor Evaluation Process (MFEP) merupakan metode kuantitatif berbasis “*Weighting System*” (Pratiwi, 2016). Teknik penyelesaian metode ini yaitu dengan penilaian subyektif dan intuitif terhadap *indicator* atau faktor penyebab dari sebuah masalah yang dianggap penting (Latif, Jamil, & Abbas, 2018).

Proses perhitungan menggunakan metode MFEP adalah:

1. Menentukan faktor dan bobot faktor.

Pada tahap ini, setiap kriteria yang menjadi faktor penting dalam pertimbangan diberi pembobotan (*weighting*) sesuai dengan intuisi dan subyektifitas pengambil keputusan. Kita mengurutkan faktor-faktor tersebut berdasarkan tingkatan kepentingannya, mulai dari yang terpenting, kedua terpenting, dan seterusnya dengan cara membandingkan tingkat kepentingan faktor-faktor tersebut. Kegiatan pembobotan ini dinamakan bobot faktor (*weight factor*). Catatan terpentingnya adalah bahwa total bobot faktor harus sama dengan 1.

2. Memberi bobot untuk masing-masing alternatif terhadap faktor-faktor penting yang telah ditentukan. Nilai yang diisikan disini adalah nilai sebenarnya untuk masing-masing alternatif untuk setiap faktor dan dinamakan evaluasi faktor (*factor evaluation*). Nilai ini merupakan data yang akan diproses dan sangat berpengaruh terhadap proses pengambilan keputusan.

3. Selanjutnya proses perhitungan bobot evaluasi (*weight evaluation*) untuk masing-masing alternatif dimana bobot evaluasi merupakan perkalian antara bobot faktor dan evaluasi faktor.

2.1.3. Metode *Analytical Hierarchy Process* (AHP)

Metode ini merupakan metode yang sifatnya persepsional, artinya tingkat kepentingan dari suatu kriteria alternatif tergantung sudut pandang atau perspektif seseorang dalam menilainya (Multi Criteria Decision Making (MCDM) pada Sistem Pendukung Keputusan, 2017). Pada dasarnya prosedur atau langkah-langkah yang perlu diperhatikan dalam metode AHP, antara lain (Pratiwi, 2016):

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
2. Menentukan prioritas elemen.
3. Membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan
4. Matriks bilangan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.
5. Sintesis/prioritas

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah:

1. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
2. Membagi setiap nilai dari kolom dengan total kolom bersangkutan untuk memperoleh normalisasi matriks.
3. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapat nilai rata-rata.
4. Mengukur konsistensi.

Penting untuk mengetahui seberapa baik konsistensi yang ada, hal-hal yang dilakukan dalam langkah ini adalah:

1. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
2. Jumlah setiap baris
3. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
4. Jumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut λ maks (nilai maksimum).
5. Hitung *Consistency Index* (CI) dengan menggunakan Rumus 2.5.

$$CI = \frac{\lambda \text{ maks} - n}{n - 1}$$

Rumus 2. 5. Perhitungan *Consistency Index* (CI)

n = banyaknya elemen berdasarkan sumber kriteria

6. Hitung *Consistency Ratio* (CR) dengan rumus, seperti Rumus 2.6.

$$CR = \frac{CI}{IR}$$

Rumus 2.6. Perhitungan Consistency Ratio (CR)

CR = Consistency Ratio;

CI = Consistency Index;

IR = Indeks random consistency.

7. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10%, maka penilaian data *judgement* harus diperbaiki. Namun jika rasio konsistensi (CI/IR) ≤ 0.1 , maka hasil perhitungan bisa dinyatakan benar. Indeks *random* (IR) bisa dilihat pada Tabel 2.2.

Tabel 2.2. Nilai Indeks Random

Ukuran Matriks	Nilai IR
1, 2	0.00
3	0.58
4	0.90
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56

Ukuran Matriks	Nilai IR
14	1.57
15	1.59

2.1.4. Metode Perbandingan Eksponensial (MPE)

Metode perbandingan eksponensial (MPE) merupakan metode yang digunakan untuk menentukan urutan prioritas alternatif keputusan dengan kriteria jamak. Pada prinsipnya metode ini merupakan metode skoring terhadap pilihan yang ada. Teknik ini digunakan untuk membantu individu pengambil keputusan untuk menggunakan rancang bangun model yang terdefinisi dengan baik pada tahapan proses. Langkah-langkah yang perlu dilakukan dalam pemilihan keputusan dengan menggunakan MPE adalah (Pratiwi, 2016):

1. Menyusun alternatif-alternatif keputusan yang akan dipilih
2. Menentukan kriteria atau perbandingan relatif kriteria keputusan yang penting untuk dievaluasi dengan menggunakan skala konversi tertentu sesuai dengan keinginan pengambil keputusan.
3. Menentukan tingkat kepentingan relatif dari setiap kriteria keputusan atau pertimbangan kriteria. Penentuan bobot ditetapkan pada setiap kriteria untuk menunjukkan tingkat kepentingan suatu kriteria.
4. Melakukan penilaian terhadap semua alternatif pada setiap kriteria dalam bentuk total skor tiap alternatif.

5. Menghitung skor atau nilai total setiap alternatif dan mengurutkannya dengan menggunakan Rumus 2.7. Semakin besar Total Nilai (TN) alternatif maka semakin tinggi urutan prioritasnya.

$$Total\ Nilai\ (TN_i) = \sum_{j=1}^m (RK_{ij})^{TKK_j}$$

Rumus 2.7. Perhitungan Nilai Total MPE

Keterangan:

TN_i = Total nilai alternatif ke – i

RK_{ij} = Derajat kepentingan relatif kriteria ke – j pada pilihan keputusan i

TKK_j = Derajat kepentingan kriteria keputusan ke – j ;

n = Jumlah pilihan keputusan

m = Jumlah kriteria keputusan

2.1.5. Metode *Weighted Sum Model* (WSM)

Weighted Sum Model merupakan salah satu metode yang paling sederhana dan mudah dipahami penerapannya dikarenakan dalam konsep metode ini hanya melakukan perkalian di antara Bobot Kriteria (W_j) dan Nilai Alternatif (X_{ij}). Adapun algoritma penyelesaian dari metode ini adalah (Nofriansyah & Defit, 2017):

1. Langkah I: Mengidentifikasi terlebih dahulu dari kriteria dan alternatif yang digunakan dalam penyelesaian Masalah.
2. Langkah II: Menghitung nilai *WSM-Score* menggunakan rumus seperti pada Rumus 2.8.

$$A_i \text{ WSM - score} = \sum_{j=1}^n w_j x_{ij}$$

Rumus 2.8. Nilai WSM-Score

Keterangan:

n = jumlah kriteria

w_j = bobot dari setiap kriteria

x_{ij} = nilai matrik x

3. Langkah III: Melakukan perangkingan.

2.1.6. Metode *Simple Additive Weighting* (SAW)

Metode *Simple Additive Weighting* (SAW) adalah metode penjumlahan bobot dari kinerja setiap objek-objek yang berbeda dan memiliki kesempatan yang sama pada semua kriteria yang dimiliki. Metode *Simple Additive Weighting* (SAW) memerlukan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua *rating* alternatif yang ada (Pratiwi, 2016).

Adapun langkah-langkah dalam metode SAW adalah sebagai berikut:

- 1) Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, $C_j, j = 1, 2, \dots, m$.
- 2) Menentukan bobot untuk masing-masing kriteria $W_j, j = 1, 2, \dots, m$ dengan catatan penting $\sum W_j = 1$.

- 3) Melakukan normalisasi matriks keputusan dengan melakukan proses perbandingan pada semua nilai *alternative* yang ada, rumus normalisasi dapat dilihat pada Rumus 2.9.:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}}, & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}}, & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

Rumus 2.9. Perhitungan perbandingan antar alternatif

Keterangan:

r_{ij} = Nilai rating kinerja ternormalisasi

x_{ij} = nilai atribut yang dimiliki dari setiap kriteria

Max x_{ij} = nilai terbesar dari setiap kriteria

Min x_{ij} = nilai terkecil dari setiap kriteria

Benefit = jika nilai terbesar adalah terbaik

Cost = jika nilai terkecil adalah terbaik

- 4) Menghitung nilai preferensi untuk tiap alternatif, V_j , diberikan sebagai

$$v_i = \sum_{j=1}^n w_j * r_{ij}$$

Rumus 2.10. Perhitungan nilai preferensi tiap alternatif

Keterangan:

V_i : rangking untuk setiap alternatif

W_j: nilai bobot dari setiap kriteria

Dari hasil perhitungan Rumus 2.10. dapat diketahui bahwa nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.2. XAMPP

XAMPP merupakan server yang paling banyak digunakan (Mengenal Pemrograman PHP7 Database untuk Pemula, 2017). XAMPP adalah singkatan dari X yang menandakan XAMPP bisa di *install* di berbagai *operating system*, Apache Server, MySQL, phpMyAdmin, dan Python. XAMPP dapat di *download* langsung dari situs resminya <http://www.apache-friends.org> (Dadan & Developers, 2015).

2.3. PHP

PHP singkatan dari PHP: *Hypertext Preprocessor*. PHP merupakan *script* yang terintegrasi dengan HTML (Anhar, 2010). PHP merupakan bahasa pemrograman berjenis *server-side*. Dengan demikian, PHP akan diproses oleh *server* yang hasil olahannya akan dikirim kembali ke *browser* (Jubilee Enterprise, 2017).

2.4. HTML

HTML merupakan singkatan dari *Hypertext Markup Language*. HTML merupakan bahasa pemrograman web yang memberitahukan peramban web (*web browser*) bagaimana menyusun dan menyajikan konten di halaman web. HTML disusun dengan bahasa yang sederhana, sehingga sangat

mudah diimplementasikan. Saat ini, HTML dapat menampilkan obyek-obyek seperti teks, tabel, tautan, gambar, *audio* dan *video* (Solichin, 2016).

2.5. phpMyAdmin

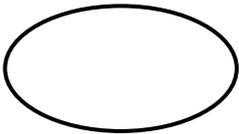
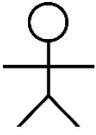
phpMyAdmin adalah aplikasi berbasis web yang digunakan untuk melakukan pengelolaan *database* MySQL dan atau *tool* yang paling populer untuk mengelola *database* MySQL (Panduan Pintar Belajar Singkat phpMyAdmin, 2018). PhpMyAdmin merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirimkan *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL (Hikmah, Supriadi, & Alawiyah, 2015).

2.6. Use Case Diagram

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 saja yang dapat dilakukan terhadap suatu sistem (Indrajani, 2015). Adapun simbol yang digunakan dalam pembuatan *use case diagram* seperti pada Tabel 2.3.

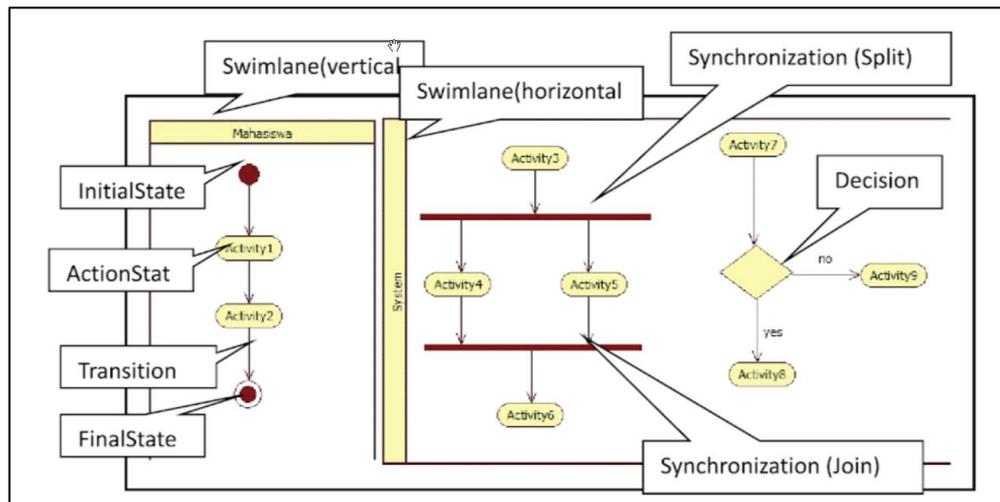
Tabel 2.3. Simbol-simbol *use case diagram*

Sumber: (Harianto, Pratiwi, & Suhariyadi, 2019)

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i>, biasanya dinyatakan dengan menggunakan kata kerja awal frase nama <i>use case</i>.</p>
<p>Aktor</p> 	<p>Berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>.</p>
<p><i>Extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.</p>
<p><i>Include</i></p> 	<p>Relasi <i>case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi atau sebagai syarat dijalankan <i>use case</i> ini.</p>

2.7. Activity Diagram

Activity diagram menjelaskan berbagai aktivitas pengguna (atau sistem), orang atau komponen yang melakukan masing-masing aktivitas, dan urutan aktivitas yang berulang (Satzinger, Jackson, & Burd, 2016). Notasi umum yang sering digunakan dalam *activity diagram* adalah sebagai berikut (Triandini & Suardika, 2012).



Gambar 2.1. Notasi Activity Diagram

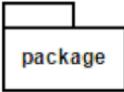
Pada Gambar 2.1. menggambarkan notasi yang sering digunakan pada *activity diagram*. Adapun penjelasan untuk masing-masing notasi:

1. *Swimlane*: mewakili agen yang melakukan aktivitas. Karena dalam alur kerja umumnya mempunyai agen yang berbeda yang melakukan langkah yang berbeda dari proses alur kerja. Simbol *swimlane* membagi aktivitas alur kerja ke dalam kelompok yang menunjukkan agen mana yang menjalankan aktivitas yang mana.
2. *InitialState*: awal dari alur kerja.
3. *Action State*: melambangkan aktivitas tersendiri dalam alur kerja.
4. *Transition*: melambangkan urutan diantara aktivitas.
5. *FinalState*: akhir dari alur kerja.
6. *Synchronization*: membagi alur kerja menjadi beberapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan.
7. *Decision*: titik pengambilan keputusan dimana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya.

2.8. Class Diagram

Class diagram, menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. Tabel 2.4. berisi simbol-simbol yang digunakan dalam *class diagram* sebagai berikut (Harianto, Pratiwi, & Suhariyadi, 2019):

Tabel 2.4. Simbol-simbol *class diagram*

Simbol	Deskripsi
<i>Package</i> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi 	Relasi antara kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
Generalisasi 	Relasi antara kelas dengan makna generalisasi-generalisasi (umum khusus)
Agregasi 	Relasi antara kelas dengan makna semua-bagian (<i>whole part</i>)

2.9. Metode Pengembangan Sistem

2.9.1. Metode *Waterfall*

Waterfall adalah serangkaian proses berurutan yang prosesnya dilihat bertahap melalui fase *requirement, analysis, design, construction, testing, deploy* dan *maintenance*. *Waterfall* tidak memiliki mekanisme untuk menangani perubahan pada *requirements* yang teridentifikasi oleh *feedback* pengguna (Habibi, Jha, Sinha, & Choudhary, 2017).

Tahapan *Waterfall* terdiri dari (The Blokehead, 2014):

1. *Requirements*

Langkah pertama dalam proses pengembangan adalah fase *requirements*. Dalam fase ini, analis akan mengumpulkan data yang diperlukan dan lebih penting; persyaratan di mana hasil akhir dimaksudkan untuk dipenuhi.

2. *Design*

Fase berikutnya, '*design*' adalah tempat kode perangkat lunak dan artefak lainnya diukir dan direncanakan. Ini mengarah ke fase berikutnya, yang merupakan fase '*implementation*'.

3. *Implementation/Construction*

Pada fase berikutnya, kode dan sumber daya yang dibuat dalam fase desain disatukan untuk menciptakan produk fungsional.

4. *Verification/Testing*

Setelah itu fase '*verification* atau *testing*' produk muncul untuk mendeteksi masalah yang ada. Selain itu, fase pengujian melihat

ke belakang dan memastikan hasil memenuhi persyaratan yang ditentukan pada fase pertama.

5. *Maintenance*

Akhirnya, fase pemeliharaan adalah proses yang berkelanjutan untuk memberikan dukungan dan perbaikan yang diperlukan begitu produk dirilis.

2.9.2. Metode Agile

Agile adalah proses untuk mengembangkan perangkat lunak yang memperhatikan individu dan interaksi atas proses dan alat, pengerjaan perangkat lunak atas dokumentasi yang komprehensif, kolaborasi pelanggan atas negosiasi kontrak, dan dapat menanggapi perubahan diluar perencanaan. Metode *Agile* dirancang fleksibel untuk mendukung kebutuhan proyek dan organisasi (Ashmore & Runyan, 2014).

XP (*Extreme Programming*) merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk mengembangkan aplikasi oleh para *developer*. XP sangat cocok untuk pengembangan proyek yang memerlukan adaptasi cepat dalam perubahan-perubahan yang terjadi selama pengembangan aplikasi. XP juga cocok untuk anggota tim yang tidak terlalu banyak dan berada pada lokasi yang sama dalam pengembangan sistem (Suryantara, 2017).

Tahapan Extreme Programming terdiri dari (Mahardika & Siregar, 2018):

1) *Planning*

Pada tahap ini meliputi mendefinisikan ruang lingkup selama penelitian, menganalisis sistem saat ini dan masalah yang terjadi. Kemudian dilanjutkan dengan menganalisis persyaratan sistem yang akan dirancang.

2) *Design*

Pada tahap ini, perancangan proses kerja sistem dan perancangan basis data yang sudah disesuaikan dengan hasil pengumpulan data dan hasil analisis yang telah dilakukan. Dalam merancang proses kerja sistem ini, biasanya menggunakan UML karena menekankan pengembangan sistem berorientasi objek.

3) Pengkodean

Pada tahap ini adalah tahap dimana pembuatan sistem didasarkan pada hasil desain yang telah dibuat sebelumnya. Desain diimplementasikan dalam bentuk kode yang dapat dipahami oleh komputer dengan bahasa pemrograman. Seperti dijelaskan sebelumnya tentang *Extreme Programming* bahwa metode ini melibatkan klien dalam proses pengembangan sistem, proses *coding* dilakukan berulang kali jika ada koreksi dari pengguna sistem (*refactoring*).

4) Pengujian

Pada tahap ini, setiap modul yang sedang dikembangkan akan terlebih dahulu menjalani pengujian. Jika masih tidak sesuai dengan permintaan, itu akan diperbaiki di bagian yang diperbaiki. Jika sesuai permintaan, sistem dapat diterapkan.

2.10. *Blackbox Testing*

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan hal-hal berikut (Mustaqbal, Firdaus, & Rahmadi, 2015):

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.11. Penelitian Terdahulu

Pada Tabel 2.5. berisi penelitian-penelitian terdahulu yang dapat berfungsi sebagai pendukung untuk melakukan penelitian ataupun sebagai pembanding dengan penelitian yang akan dilakukan. Adapun penelitian tersebut diantaranya:

Tabel 2.5. Hasil Penelitian Terdahulu

No	Penulis	Judul	Nama Jurnal	Hasil/Kesimpulan
1	Raymond Heris Zamrudi, Friska Natalia	<i>Decision Support System Based on The Technique For Order Preference by Similarity to Ideal Solution (Topsis)</i>	<i>International Journal of Latest Trends in Engineering and Technology</i> Vol. (12) Issue (4), 2019	Dalam penelitian ini, telah dibuat sebuah situs web yang dapat membantu memilih sepeda motor secara otomatis sesuai dengan kriteria yang diisi oleh pengguna berdasarkan kebutuhan dan selera masing-masing pengguna. Pencarian dan pemilihan dilakukan secara otomatis sesuai dengan kriteria yang dipilih oleh pengguna berdasarkan teori pendukung keputusan TOPSIS sehingga mereka dapat menghitung bobot masing-masing kriteria dan dapat menghasilkan hasil yang akurat sesuai dengan harapan pengguna.
2	Devara Eko Katon Mahardika, Maria Ulfah Siregar	<i>Design and Development of Web Based Employee Payroll Information System Using Codeigniter Framework and Extreme Programmin g Method</i>	<i>International Journal on Informatics for Development</i> 2018 Vol 7:2	Penggunaan metode Extreme Programming dalam pengembangan sistem di penelitian ini menghasilkan tiga fase pengembangan. Dalam setiap fase selalu ada fase pengujian yang melibatkan pemilik proyek untuk mengetahui apakah sistem tersebut sesuai dengan kebutuhan atau tidak. Fase pengembangan sistem dinyatakan selesai ketika semua fungsionalitas sistem yang direncanakan berhasil dilaksanakan dengan benar dan tidak mendapatkan koreksi dari pemilik proyek.
3	Humam Alif Subhi, Yasni Djamain, M. Farid Rifai	Rancang Bangun Aplikasi GATTO Friends dengan Metode	PETIR (Jurnal Pengkajian Dan Penerapan Teknik Informatika) 2017, Vol 10:2	Dengan menggunakan aspek kesehatan dan aspek ideal kucing dalam perkawinan untuk pembobotan kucing jantan dan kucing betina, sehingga nilai yang dirangkingkan menjadikan

No	Penulis	Judul	Nama Jurnal	Hasil/Kesimpulan
		<i>Profile Matching</i> Berbasis Android (Studi Kasus UNDIP Cat Lovers Semarang)		prediksi yang cukup akurat, maka metode <i>Profile Matching</i> efektif digunakan untuk mencarikan pasangan kucing.
4	M. Azman Maricar, Wahyudin, Made Sudarma	<i>Decision Support System of the Employees Acceptance using Analytical Hierarchy Process (AHP) and Multi Factor Evaluation Process (MFEP)</i>	<i>International Journal of Engineering and Emerging Technology</i> , Vol. 1, No. 1 2016	Pada penelitian ini, dalam AHP, MFEP digunakan untuk menghitung bobot setiap kriteria dengan nilai evaluasi masing-masing kandidat. Tetapi, sebelumnya nilai evaluasi diproses dalam perhitungan matriks untuk melakukan perbandingan berpasangan (antar kandidat) dan dilengkapi dengan perhitungan rasio konsistensi. Kedua metode tersebut menghasilkan hasil yang sama dan dapat membantu memberikan rekomendasi kepada pengambil keputusan. Dengan menggunakan aturan dari The Saaty, dimana standar untuk <i>consistency ratio</i> adalah lebih kecil dari 0.1. Pada penelitian ini menunjukkan setiap kriteria memiliki hasil lebih kecil dari 0.1 sehingga seluruh kriteria dapat dikatakan konsisten.
4	Zouhour Chourabi, Faouzi Khedher, Amel Babay & Morched Cheikhrouhou	Multi-criteria decision making in workforce choice using AHP, WSM and WPM	The Journal of The Textile Institute 2018, Vol 109	Dalam tulisan ini, seleksi tenaga kerja dianggap sebagai kriteria keputusan masalah. Metode AHP, WSM dan WPM diimplementasikan untuk mengekspresikan secara obyektif kemampuan operator. Skor MCDM merangkum kriteria dengan cara obyektif membuat penilaian pekerja lebih akurat daripada pendapat para ahli

Dari penelitian terdahulu dapat diketahui beberapa informasi pendukung yang dapat digunakan dalam penelitian ini diantaranya penggunaan metode *profile matching*, *multi factor evaluation process*, dan *analytical hierarchy process* sebagai dasar dalam pembuatan sistem pendukung keputusan serta penerapan *extreme programming* dalam proses pengembangan sistem. Dengan informasi-informasi tersebut, adapun hal-hal yang akan diterapkan pada penelitian ini yaitu:

1. *State of the art*:

Sistem dapat membantu memberikan rekomendasi dari proses penyeleksian dokumen pelamar yang juga sudah dicocokkan dengan kebutuhan untuk masing-masing lowongan.

2. Adopsi

- Penggunaan metode *Profile Matching* untuk proses pencocokan profil pelamar.
- Penggunaan metode *Analytical Hierarchy Process* untuk menghitung konsistensi hasil akhir.

3. Kebaruan

- Penambahan informasi status rekrutmen, baik untuk pelamar yang lolos ataupun tidak.

- Penambahan *dashboard* untuk membantu memonitoring kebutuhan rekrutmen.
- Berlaku untuk kebutuhan berbagai macam lowongan, tidak terbatas hanya untuk posisi tertentu.