

## BAB II

### LANDASAN TEORI

#### 2.1 Kajian Terhadap Penelitian Yang Terkait Sebelumnya

Berikut ulasan beberapa penelitian terkait yang menjadi referensi pada penelitian ini: Hidayatus Sibyan dalam jurnal yang berjudul: “Implementasi Metode SMART Pada Sistem Pendukung Keputusan Penerima Beasiswa” pada judul ini penelitian menggunakan metode SMART sebagai hasil penelitian bahwa metode SMART sangat cocok untuk menentukan beasiswa dengan kriteria Ranking paralel, Hafalan Qur’an, Kondisi Ekonomi, dan Prestasi Siswa.

Berikut ulasan beberapa penelitian terkait yang menjadi referensi pada penelitian ini: Solikhun dalam jurnal yang berjudul: “Perbandingan Metode Weighted Product dan Weighted Sum Model Dalam Pemilihan Perguruan Swasta Terbaik Jurusan Komputer” pada judul ini menggunakan metode Weighted Product dan Weighted Sum Model untuk membandingkan nilai akhir dari kedua metode tersebut, dan dapat menentukan ranking dari perguruan tinggi swasta.

Berikut ulasan beberapa penelitian terkait yang menjadi referensi pada penelitian ini: Aidil Putrasyah dan Sukemi dalam jurnal yang berjudul: “Perhitungan Peramalan Harga Emas Menggunakan Metode Single Exponential Smoothing Dan Moving Average” pada judul ini penelitian tersebut menggunakan metode *Mean Squared Error* untuk menghitung hasil peramalan dan tingkat kesalahannya untuk mengetahui metode mana yang paling sesuai digunakan dalam menganalisis data dengan tingkat kesalahan yang paling kecil.

## 2.2 Kendo

Kendo berasal dari kata ken (剣) yang berarti pedang, dan do (道) yang berarti jalan. Secara harafiah, kendo berarti jalan pedang yang berarti juga jalan hidup seseorang yang berlatih menggunakan pedang (Tokeshi, 2003:3).

Dalam *kendo* sendiri terdapat tiga serangan dasar yaitu serangan *men* ke arah kepala, serangan *kote* ke arah pergelangan tangan, dan serangan *do* ke arah perut. Masing-masing titik pukulan tersebut terbagi lagi menjadi sisi pukulan sebelah kanan dan sebelah kiri. Namun ada juga serangan yang hanya boleh dilakukan oleh para kendoka yang sudah lebih berpengalaman. Serangan tersebut adalah serangan *tsuki*, yaitu ke arah tenggorokan lawan. Keempat serangan tersebut adalah serangan-serangan dasar yang dapat dilakukan dalam kendo.

*Keiko* sendiri merupakan sebutan untuk latihan tanding dalam seni tradisional Jepang seperti *kendo*, *ikebana*, dan lain-lain. Saat melakukan *keiko* pun kita diajarkan untuk tetap waspada, hal ini disebut *zanshin* dalam *kendo*. *Zanshin* adalah gerakan siap atau waspada setelah kita menyerang lawan baik itu kena atau tidak. Saat menyerang seorang *kendoka* juga wajib untuk melakukan *kiai* atau teriakan yang menunjukkan semangat dalam diri.

## 2.3 Aplikasi

Aplikasi berasal dari kata *Application* (Bahasa Inggris) yang diartikan menurut KBBI adalah penerapan; lamaran; penggunaan; program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu.

Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat

digunakan oleh sasaran yang dituju. Beberapa pandangan mengenai aplikasi antara lain sebagai berikut :

1. Menurut Pramana di dalam jurnal (Rohayah, Sasmito, & Somantri, 2015) “Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan dan hampir semua proses kegiatan”.
2. Menurut (Abdurahman, Hasan & Riswaya, 2014) didalam Jurnalnya : “Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut”

Dari pengertian diatas, dapat penulis simpulkan bahwa aplikasi adalah suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan oleh pengguna.

#### **2.4 Sistem Pendukung Keputusan**

Menurut Sibyan (2017) SPK yaitu sebuah sistem yang dipergunakan untuk proses pengambilan sebuah keputusan yang bisa membantu tugas dari pengambil keputusan. Komponen- komponen pembangun Sistem Pendukung Keputusan diantaranya yaitu (1) subsistem data yaitu tempat penyimpanan data dalam sistem; (2) subsistem model adalah model keputusan yang diintegrasikan dengan data; dan (3) subsistem dialog adalah antar muka pengguna sebagai sarana komunikasi antara pengguna dengan sistem secara interaktif.

Menurut Abas (2016:3), Definisi awal dari SPK yaitu mengidentifikasi sebuah sistem sebagai yang dimaksudkan untuk mendukung manajerial pembuat keputusan dalam situasi keputusan semi-terstruktur. SPK dimaksudkan untuk menjadi tambahan untuk para pengambil keputusan untuk memperluas kemampuan mereka tetapi tidak untuk menggantikan penilaian mereka.

Dari pengertian diatas, dapat penulis simpulkan bahwa sistem pendukung keputusan adalah suatu sistem yang memanfaatkan kemampuan *tools* komputer terintegrasi untuk membuat keputusan semi terstruktur dan tak terstruktur untuk menjadi tambahan (referensi yang lebih luas) untuk seorang *decision maker* menggantikan penilaian mereka.

## **2.5 Sistem Informasi**

Menurut Stair dan Reynolds (2010) mendefinisikan sistem informasi sebagai seperangkat elemen atau komponen yang saling berhubungan, dikumpulkan (*input*), manipulasi (*process*), menyimpan, dan menyebarkan (*output*) data dan informasi, memberikan reaksi korektif (*feedback*) untuk memenuhi tujuan.

Selain itu, Laudon dan Laudon (2010) mendefinisikan sistem informasi merupakan komponen yang saling bekerja sama untuk mengumpulkan, mengolah, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengendalian, analisis masalah dan visualisasi dalam sebuah organisasi.

Aktivitas dasar dari sistem informasi menurut Laudon dan Laudon (2010) adalah sebagai berikut :

1. *Input*: Melibatkan pengumpulan data mentah dari dalam organisasi atau dari lingkungan eksternal untuk pengolahan dalam suatu sistem informasi.
2. *Process*: Melibatkan proses mengkonversi input mentah ke bentuk yang lebih bermakna.
3. *Output*: Mentransfer proses informasi kepada orang yang akan menggunakannya atau kepada aktivitas yang akan digunakan.
4. *Feedback*: Output yang dikembalikan ke anggota organisasi yang sesuai untuk kemudian membantu mengevaluasi atau mengoreksi tahap Input.

Jadi dapat disimpulkan bahwa Sistem Informasi adalah kombinasi seperangkat komponen yang terdiri dari orang, *hardware*, *software*, jaringan telekomunikasi dan data yang saling bekerja sama untuk mengumpulkan, mengolah, menyimpan, dan menyebarkan informasi untuk mendukung pengambilan keputusan, pengendalian, analisis masalah dan visualisasi dalam organisasi.

## **2.6 *Simple Multi Attribute Rating Technique***

Metode *SMART* yang merupakan singkatan dari *Simple Multi Attribute Rating Technique* adalah sebuah metode untuk menangani permasalahan multi-kriteria dalam sistem pendukung keputusan yang dikembangkan pada tahun 1997 oleh Edward. Metode pengambilan keputusan ini menangani permasalahan multi-kriteria berdasarkan pada nilai-nilai yang dimiliki oleh setiap alternatif pada masing-masing kriteria yang telah diberi bobot. Bobot setiap kriteria digunakan untuk membandingkan antara tingkat kepentingan antara kriteria satu dengan yang

lain. Perhitungan pembobotan akan menghasilkan nilai untuk masing-masing alternatif untuk memperoleh alternatif yang paling baik (Novianti, Astuti, & Khairina, 2016).

Tahapan-tahapan yang ada dalam metode SMART adalah sebagai berikut (Goodwin and Wright 2004) :

1. Menentukan jumlah kriteria.
2. Menentukan bobot dari masing-masing kriteria.
3. Normalisasi bobot.

$$\frac{W_j}{\sum W_j}$$

Gambar 2.1 Rumus Normalisasi Bobot *SMART*

**Keterangan**

- $w_i$  : bobot kriteria ternormalisasi untuk kriteria ke-i
  - $w'_i$  : bobot kriteria ke-i
  - $w_j$  : bobot kriteria ke-j
  - $j$  : 1,2,3, ... , m jumlah kriteria
4. Memberikan nilai pada setiap kriteria untuk setiap alternatif.
  5. Menghitung nilai utility untuk setiap kriteria.

$$u_i(a_i) = \frac{(c_{out} - c_{min})}{(c_{max} - c_{min})}$$

Gambar 2.2 Rumus Utility *Benefit SMART*

$$u_i(a_i) = \frac{(c_{max} - c_{out})}{(c_{max} - c_{min})}$$

Gambar 2.3 Rumus Utility Cost SMART

**Keterangan**

- $u_i(a_i)$  : nilai utility kriteria ke-i untuk alternatif ke-i
- $c_{max}$  : nilai kriteria maksimal
- $c_{min}$  : nilai kriteria minimal
- $c_{out}$  : nilai kriteria ke-i

6. Menghitung nilai akhir masing-masing alternatif dan melakukan perankingan.

$$u(a_i) = \sum_{j=1}^m w_j * u_j(a_i)$$

Gambar 2.4 Rumus Akhir SMART

**Keterangan**

- $u(a_i)$  : nilai total untuk alternatif ke-i
- $w_j$  : nilai bobot kriteria ke-j yang sudah ternormalisasi
- $u_j(a_i)$  : nilai utility kriteria ke-j untuk alternatif ke-i

**2.7 Weighted Product**

Menurut Manik (2018), Metode *Weighted Product* menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.

Secara umum, terdapat 3 langkah untuk melakukan perhitungan *weighted product*, yaitu :

- Penentuan nilai bobot W
- Penentuan nilai bobot S
- Penentuan nilai bobot V

$$w_j = \frac{W_j}{\sum W_j}$$

Gambar 2.5 Rumus Normalisasi Bobot WP

$$S_i = \prod_{j=1}^n X_{ij}^{W_j}$$

Gambar 2.6 Mencari nilai Vector S

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{W_j}}{\prod_{j=1}^n X_{ij} * W_j} \text{ atau } V_i = \frac{S_i}{\sum S_i}$$

Gambar 2.7 Penentuan nilai Vector V

Keterangan :

V= Preferensi alternative dianalogikan sebagai vector V

X= Nilai kriteria

W= bobot kriteria/sub kriteria

i= alternative

j= Kriteria

n= banyaknya kriteria

\*= **banyaknya kriteria yang telah dinilai pada vector S**

## 2.8 Mean Squared Error

Menurut Suryaningrum (Suryaningrum & W, 2015), *Mean Squared Error (MSE)* adalah metode lain untuk mengevaluasi metode peramalan. Masing-masing



kesalahan atau sisa dikuadratkan. Pendekatan ini mengatur kesalahan peramalan yang besar karena kesalahan-kesalahan itu dikuadratkan. Metode itu menghasilkan kesalahan-kesalahan sedang yang kemungkinan lebih baik untuk kesalahan kecil, tetapi kadang menghasilkan perbedaan yang besar. *MSE* merupakan cara kedua untuk mengukur kesalahan peramalan keseluruhan. *MSE* merupakan rata-rata selisih kuadrat antara nilai yang diramalkan dan yang diamati. Rumus untuk menghitung *MSE* adalah sebagai berikut.

$$MSE = \frac{\sum_{t=1}^n (A_t - F_t)^2}{n}$$

Gambar 2.8 Rumus metode MSE

Penjelasan :

$A_t$  = Nilai awal periode ke- $t$

$F_t$  = Nilai akhir (peramalan) periode ke- $t$

$n$  = Jumlah periode  $t$

$t$  = Periode

## 2.9 Perangkat Lunak yang Digunakan dalam Membuat Aplikasi

### 2.9.1 *HTML (Hypertext Markup Language)*

*Hypertext Markup Language* (HTML) merupakan suatu bahasa markup yang digunakan untuk melakukan markup terhadap sebuah dokumen teks. Dalam dokumen atau skrip tersebut terdapat kode-kode atau perintah-perintah yang nantinya akan ditransfer oleh *http* kedalam web browser.

Web browser seperti *Microsoft Edge*, *Mozilla Firefox*, *Chromium* dan lain-lain berfungsi untuk menjalankan kode-kode *HTML*. Didalam sebuah file *HTML*

terdapat tag-tag *HTML* yang secara umum terbagi ke dalam dua bagian yaitu head (kepala) dan body (tubuh). File *HTML* biasanya diawali dengan tag dan diakhiri dengan tag (Bunafit Nugroho, 2008).

### **2.9.2 CSS (*Cascading Style Sheet*)**

Menurut Jayan (2010:2) mengemukakan bahwa “*CSS* merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen *HTML*, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar.”

*CSS* dikembangkan oleh W3C. organisasi yang mengembangkan teknologi internet. Tujuannya tidak lain untuk mempermudah proses penataan halaman web.

### **2.9.3 PHP**

*Hypertext Preprocessor (PHP)* yaitu bahasa pemrograman web server side yang bersifat open source. *PHP* merupakan script yang terintegrasi dengan *HTML* dan berada pada server. *PHP* adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Semua script *PHP* dieksekusi pada server di mana script tersebut dijalankan.

Ketika dipanggil dari *web browser*, program yang ditulis dengan *PHP* akan di-parsing dalam *web server* oleh interpreter *PHP* dan diterjemahkan dalam dokumen *HTML*, yang selanjutnya akan ditampilkan kembali ke *web browser*. Karena pemrosesan program *PHP* dilakukan dilingkungan *web server*, *PHP* dikatakan sebagai bahasa sisi server (*server-side*) oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode *PHP* tidak akan terlihat pada saat user memilih perintah “*view source*” pada *web browser* yang mereka gunakan. Selain

menggunakan *PHP*, aplikasi *web* juga dapat dibangun dengan *Java* (*JSP*-*JavaServer Pages* dan *Servlet*), *Perl*, maupun *ASP* (*Active Server Pages*).

Kode *PHP* diawali dengan `<?php` dan diakhiri dengan `?>`. Pasangan kedua kode inilah yang berfungsi sebagai tag kode *PHP*. Berdasarkan tag inilah, pihak server dapat memahami kode *PHP* dan kemudian memprosesnya. Hasilnya dikirim ke *browser* (Anhar, 2010).

Selain itu, *PHP* kita juga dapat menggunakan tag gaya *ASP* `<%` dan `%>`, dengan mengubah nilai *asp\_tags* dalam file *PHP* ini menjadi `on`. Berikut beberapa contoh perintah *query SQL* di *PHP*:

1. `mysqli_connect()` untuk koneksi ke *database*.
2. `mysqli_query($sql)` untuk menerjemahkan proses *query* ke *MySQL* dari *PHP*.
3. `mysqli_num_rows($query)` untuk menghitung jumlah baris hasil *query*.

#### **2.9.4 Bootstrap**

*Bootstrap* adalah sebuah *css framework* yang dikembangkan pengembang *Twitter* pada pertengahan tahun 2010. Sebelum menjadi kerangka kerja *open-source*, *Bootstrap* dikenal sebagai *Twitter Blueprint*. Dan terus berkembang sampai saat ini dan *Bootstrap* telah menjadi salah satu *front-end framework* yang paling populer dan merupakan proyek *open-source* di dunia. *Bootstrap* digambarkan sebagai *CSS* sederhana namun dibangun dengan *preprocessor* yang menyediakan lebih banyak daya dan fleksibilitas dibanding *CSS* standar.

#### **2.9.5 Font Awesome**

*Font Awesome* adalah font dan ikon toolkit berdasarkan *CSS* dan *Less*. Dibuat oleh Dave Gandy untuk digunakan dengan *Bootstrap*, dan kemudian

dimasukkan ke dalam *BootstrapCDN*. *Font Awesome* memiliki 38% pangsa pasar di antara situs web yang menggunakan skrip font pihak ketiga di platform mereka, menempatkannya di tempat kedua setelah *Google Fonts*.

### 2.9.6 *Javascript*

*Javascript* adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti *Microsoft Edge*, *Mozilla Firefox*, *Chrome*, dan lain-lain. Kode *javascript* dapat disisipkan dalam halaman web menggunakan *tag script*. Contoh penulisan tag script dalam *HTML*:

```
<script>
    console.log("Hi, ini kode Javascript");
</script>
```

Jika tidak ingin file *HTML* penuh dengan berbagai macam kode, *Javascript* dapat dilakukan pengkodean dengan file yang berbeda dan dapat dipanggil ke dalam file *HTML* sebagai contoh dibawah ini

```
// file-eksternal.js
alert("Kode Javascript dari luar file HTML");
```

Untuk memanggil file diatas maka dapat menggunakan *tag script* di head pada file *HTML* seperti dibawah ini

```
<script src="file-eksternal.js"></script>
```

### 2.9.7 *Jquery*

Menurut Lukman Hakim (2010) *jQuery* adalah *Javascript Library* berupa kumpulan kode/fungsi *javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *javascript*. Secara standar, apabila kita membuat kode *javascript*, maka diperlukan kode yang cukup panjang, bahkan

terkadang sangat sulit dipahami, *jQuery* digunakan untuk menyederhanakan kode javascript. Hal ini sesuai dengan slogannya “*Write less, do more*”, cukup tulis sedikit tapi bisa melakukan banyak hal.

### **2.9.8 MySQL**

Pada perkembangannya, *MYSQL* disebut juga *SQL* yang merupakan singkatan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. *SQL* pertama kali didefinisikan oleh *American National Standards Institute (ANSI)* pada tahun 1986. *MYSQL* adalah sebuah sistem manajemen *database* yang bersifat *open source*.

*SQL* juga dapat diartikan sebagai antar muka standar untuk sistem manajemen relasional, termasuk sistem yang beroperasi pada komputer pribadi. *SQL* memungkinkan seorang pengguna untuk mengetahui dimana lokasinya, atau bagaimana informasi tersebut disusun. *SQL* lebih mudah digunakan dibandingkan dengan bahasa pemrograman, tetapi rumit dibandingkan *software* lembar kerja dan pengolah data. Sebuah pernyataan *SQL* yang sederhana dapat menghasilkan set permintaan untuk informasi yang tersimpan pada komputer yang berbeda diberbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak.

### **2.9.9 UML (Unified Modelling Language)**

*UML* merupakan alat yang sangat baik yang dapat meningkatkan kualitas analisis dan perancangan sistem dengan sangat besar, oleh karena itu dapat membantu menciptakan sistem informasi yang berkualitas tinggi. Dengan menggunakan *UML* dalam siklus iteratif analisis sistem, dapat mencapai pemahaman yang lebih tinggi antara tim bisnis dan tim *IT* berkenaan dengan

kebutuhan sistem dan prosesnya yang perlu terjadi didalam sistem tersebut untuk memenuhi kebutuhan-kebutuhan tersebut. Setiap iterasi mendapatkan suatu kenampakan pada rancangan sistem yang lebih detail sampai benda-benda dan hubungannya dalam sistem terlihat jelas, serta definisi yang lebih tepat dalam dokumen *UML* (Kendall dan Kendall, 2011).

### **2.9.10 Use Case Diagram**

Menurut Kendall dan Kendall (2011), Use Case Diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk use case dan actor. Actor merepresentasikan orang yang akan mengoperasikan atau berinteraksi dengan sistem aplikasi. Tujuan utama pemodelan *use case* adalah:

1. Memutuskan dan mendeskripsikan kebutuhan-kebutuhan fungsional sistem.
2. Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *use case* digunakan di seluruh proses pengembangan untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya yang mengacu sistem harus memberikan fungsionalitas yang dimodelkan pada *use case*.
3. Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.

### **2.9.11 Activity Diagram**

Menurut Kendall dan Kendall (2011), *Activity diagram* menunjukkan urutan aktivitas dalam suatu proses, termasuk aktivitas sekuensial dan paralel, serta

keputusan yang dibuat. *Activity* diagram biasanya dibuat untuk satu kasus penggunaan dan mungkin menunjukkan berbagai kemungkinan skenario.

### **2.9.12 *Sequence Diagram***

Menurut Kendall dan Kendall (2011), *Sequence Diagram* dapat menggambarkan rangkaian interaksi antara kelas atau instance objek. Dalam implementasinya, *Sequence Diagram* diturunkan dari analisis kasus penggunaan dan digunakan dalam sistem desain untuk mendapatkan interaksi, hubungan, dan metode objek dalam sistem. *Sequence Diagram* digunakan untuk menunjukkan pola keseluruhan dari aktivitas atau interaksi dalam use case. Setiap skenario use case dapat membuat satu *Sequence Diagram*.

### **2.9.13 *Deployment Diagram***

Menurut Sukamto dan Shalahuddin (2015) memberikan batasan bahwa, “Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”.

*Deployment Diagram* menggambarkan hubungan antara *software* dan *hardware* terhadap sistem dan apa saja output yang dihasilkan dan menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.