



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Key Performance Indicator**

*Key Performance Indicator* (KPI) adalah indikator kunci yang dapat diukur untuk kepentingan perusahaan/organisasi dalam memantau kinerja apakah sesuai dengan ekspektasi atau belum (Lubis & Kusumanto, 2018). KPI sangat dibutuhkan oleh perusahaan/organisasi supaya tetap berada *track* yang tepat serta dapat mempertahankan *quality standard* yang selama ini dipegang.

Secara umum terdapat lima tahapan (Doto & Ikatrinasari, 2018) untuk melakukan penilaian manajemen kinerja antara lain pengetahuan terkait dengan misi dan tujuan strategis organisasi dan pengetahuan tentang pekerjaan yang akan dilakukan, *performance planning*, *performance execution*, *performance assessment*, dan *performance review*

*Key performance indicator* sendiri bertujuan untuk menyelaraskan visi, misi, strategi serta nilai perusahaan untuk mencapai sebuah tujuan dan untuk meningkatkan objektivitas dalam melakukan penilaian kinerja (Gunawan, 2019). Penilaian kinerja karyawan sendiri bergantung kepada kebijakan masing-masing perusahaan, beberapa perusahaan ada yang melakukannya dalam periode per semester namun ada juga yang melakukannya selama tahunan.

## **2.2 Sistem Pendukung Keputusan**

Sistem pendukung keputusan saat ini sudah banyak digunakan untuk membantu baik perusahaan atau organisasi sebagai alat untuk menentukan sebuah keputusan yang strategis, serta untuk menghasilkan sebuah keputusan yang objektif berdasarkan data-data pendukung.

Sistem pendukung keputusan sendiri merupakan suatu sistem berbasis komputer yang bertujuan untuk membantu pengambilan keputusan berdasarkan data dan metode tertentu untuk menyelesaikan permasalahan yang tidak terstruktur (Ningsih, Dedih, & Supriyadi, 2017).

### **2.2.1 Simple Additive Weighting**

Di dalam sistem pendukung keputusan (SPK), terdapat berbagai macam algoritma untuk mendukung SPK yang dapat menghasilkan keluaran yang terpercaya, salah satunya adalah SAW (*Simple Additive Weighting*).

SAW merupakan sebuah algoritma atau metode untuk menjumlahkan nilai dari masing-masing pilihan dan memerlukan proses normalisasi dari masing-masing pilihan tersebut sehingga dapat dibandingkan satu sama lain (Setiawan et al., 2018).

Untuk memperjelas konsep dan langkah-langkah metode SAW (*Simple Additive Weighting*) berikut merupakan langkah-langkahnya:

## 1. Menentukan Kriteria

Menentukan kriteria untuk penilaian alternatif yang terdiri dari dua yakni kriteria yang bersifat benefit dan cost. Benefit memiliki arti bahwa kriteria tersebut merupakan faktor pendukung, sedangkan cost merupakan faktor penghambat.

Berikut merupakan contoh kriteria dalam penilaian terhadap kasus penerimaan beasiswa:

- Prestasi (C2)
- Ekonomi Orangtua (C1)
- Inklusi (C3)

## 2. Mengisi Bobot Dari Alternatif

Pada tahap ini, terdapat tiga alternatif yaitu bernama Lutfi, Budi dan Indah. Dari ketiga alternatif tersebut masing-masing akan diisi bobotnya sesuai kriteria yang berlaku. Asumsikan nilai maksimal dari bobot adalah 1.

**Tabel 2. 1 Tabel kriteria dan alternatif pilihan.**

Nama Pelajar	kriteria		
	C1	C2	C3
Budi S	0,5	1	0,7
Lutfi H	0,8	0,7	1
Indah P	1	0,3	0,4

### 3. Menentukan Bobot Kriteria

Dari kriteria yang ada, haruslah diberikan bobot dari yang paling berpengaruh sampai tidak. Asumsikan bahwa bobot paling besar adalah 1.

**Tabel 2. 2 Tabel penentuan bobot kriteria.**

Kriteria	Bobot
C1	0,3
C2	0,4
C3	0,3
Total	1

Selanjutnya adalah melakukan normalisasi dari masing-masing alternatif beserta dengan bobot yang didapat. Pertama mari lakukan normalisasi pada kriteria C1, karena C1 merupakan benefit maka nilai nilai yang ada di kolom C1 dibagi dengan nilai tertinggi yakni 1, seperti gambar berikut:

$$R11 = 0,5 / 1 = 0,5$$

$$R21 = 0,8 / 1 = 0,8$$

$$R31 = 1 / 1 = 1$$

**Gambar 2. 1 Ilustrasi proses normalisasi kriteria.**

Begitu sampai dengan C3 dan disesuaikan dengan sifatnya antara *benefit* dan *cost*.

#### 4. Membuat Tabel Hasil Normalisasi

Setelah melakukan perhitungan pada langkah ketiga, saatnya memasukan hasil tersebut ke dalam tabel sesuai dengan alternatifnya. Seperti gambar di bawah ini:

**Tabel 2. 3 Ilustrasi hasil penghitungan normalisasi.**

0,5	0,3	0,5
0,8	0,4	0,4
1	1	1

#### 5. Mengalikan Bobot Kriteria

Setelah mendapatkan hasil normalisasi dari langkah sebelumnya, selanjutnya adalah mengalikan nilai normalisasi dari setiap alternatif dengan bobot kriteria. Asumsikan alternatif pertama memiliki nilai 0,5, 0,3, 0,5 dan masing-masing hasil normalisasi tersebut dikalikan dengan bobot kriteria yang terdiri dari 0,3, 0,4, 0,3.

### **2.2.2 Analytical Hierarchy Process**

Sistem pendukung keputusan juga memiliki metode lain selain SAW, yaitu AHP yang juga dapat mendukung dalam menghasilkan keluaran yang dapat dipercaya. AHP merupakan sebuah model pengambil keputusan yang bersifat komprehensif dengan memperhitungkan hal-hal yang bersifat kualitatif dan kuantitatif (Sasongko, Astuti, & Maharani, 2017).

Untuk memperjelas langkah-langkah dan cara kerjanya, berikut merupakan langkah-langkahnya:

#### **1. Menetapkan masalah, kriteria, dan sub-kriteria**

Asumsikan menggunakan contoh kasus dalam pemilihan mahasiswa terbaik menggunakan metode AHP. Maka masalah yang harus diselesaikan adalah bagaimana menentukan mahasiswa terbaik berdasarkan kriteria IPK, nilai TOEFL, dan jabatan organisasi. Sedangkan sub-kriteria merupakan range dari masing-masing kriteria, seperti IPK (3.5-4.00 bermakna sangat baik), nilai TOEFL (500-600 bermakna sangat baik), dan jabatan organisasi yang terdiri dari ketua, coordinator, dan anggota.

#### **2. Membuat matrix Pairwise Comparison**

Penentuan nilai pada matrix pairwise comparison ini bersifat subjektif karena tergantung dari masukan yang diberikan oleh penilai.

**Tabel 2. 4 Matriks Pairwise Comparison Kriteria.**

	IPK	TOEFL	Jabatan
IPK	1	3	4
TOEFL	1/3	1	2
JABATAN	1/4	1/2	1

Tabel 2.4 menunjukkan bahwa IPK 4 kali lebih penting dibandingkan dengan jabatan, 3 kali lebih penting daripada nilai TOEFL. Pada baris kedua menyatakan nilai TOEFL 1/3 kali lebih penting dari IPK, 2 kali lebih penting dari jabatan, dan baris ketiga menyatakan jabatan 1/4 lebih penting dari IPK dan 1/2 lebih penting dari nilai TOEFL.

### 3. Membuat vektor prioritas

**Tabel 2. 5 Nilai Vektor Prioritas.**

	IPK	TOEFL	Jabatan
IPK	1.000	3.000	4.000
TOEFL	0.333	1.000	2.000
Jabatan	0.250	0.500	1.000
Jumlah	1.583	4.500	7.000

Tabel 2.5 merupakan hasil lanjutan dari tabel 2.4 dengan mengubah angka pada tabel 2.4 menjadi bentuk desimal.

**Tabel 2. 6 Hasil Pembagian Vektor Prioritas.**

	IPK	TOEFL	Jabatan
IPK	0.632	0.667	0.571
TOEFL	0.211	0.222	0.286
Jabatan	0.158	0.111	0.143

Tabel 2.6 merupakan hasil pembagian dari angka yang terdapat pada tabel 2.5. Misalnya untuk IPK dengan IPK mendapat angka 0.632 dari hasil pembagian  $1.000/1.583$ , begitu juga dengan kolom dan baris lainnya menggunakan perhitungan yang sama.

#### 4. Normalisasi Vektor Prioritas

Kemudian diperlukan melakukan normalisasi vektor prioritas atau biasa disebut dengan Eigen vektor normalisasi. Nilai normalisasi didapat dengan cara menjumlah hasil pembagian vektor prioritas yang terdapat pada tabel 2.6 lalu dibagi dengan banyaknya jumlah kriteria yaitu 3 buah.

**Tabel 2. 7 Normalisasi Eigen Vektor.**

	IPK	TOEFL	Jabatan	Jumlah Baris	Eigen vektor Normalisasi
IPK	0.632	0.667	0.571	1.870	0.623
TOEFL	0.211	0.222	0.286	0.718	0.239
Jabatan	0.158	0.111	0.143	0.412	0.137

## 5. Membuat Pembobotan

Untuk nilai pembobotan diasumsikan sama bagi semua kriteria dan sub-kriteria mengikuti hasil pada tabel 2.7

**Tabel 2. 8 Sub-Kriteria IPK.**

	Sangat Baik	Baik	Cukup	Jumlah Baris
Sangat Baik	0.632	0.667	0.571	1.870
Baik	0.211	0.222	0,286	0.718
Cukup	0.158	0.111	0,143	0.412

**Tabel 2. 9 Sub-Kriteria TOEFL.**

	Sangat Baik	Baik	Cukup	Jumlah Baris
Sangat Baik	0.632	0.667	0.571	1.870
Baik	0.211	0.222	0,286	0.718
Cukup	0.158	0.111	0,143	0.412

**Tabel 2. 10 Sub-Kriteria Jabatan.**

	Sangat Baik	Baik	Cukup	Jumlah Baris
Sangat Baik	0.632	0.667	0.571	1.870
Baik	0.211	0.222	0,286	0.718
Cukup	0.158	0.111	0,143	0.412

## **6. Menentukan Ranking**

Langkah terakhir adalah melakukan perankingan berdasarkan nilai pada tabel 2.8, 2.9, dan 2.10. Penghitungan dilakukan dengan cara apabila terdapat mahasiswa bernama Tono memiliki IPK 3.80 yang bermakna sangat baik maka ditulis dengan angka 1, nilai TOEFL cukup maka ditulis dengan angka 3, dan jabatan organisasi sebagai anggota maka ditulis dengan angka 3.

Kemudian kalikan nilai normalisasi Eigen dengan pembobotan masing-masing sub-kriteria sebelumnya. Sebagai contoh, IPK (0.623) x pembobotan IPK (0.632), TOEFL (0.239) x pembobotan TOEFL (0.143), dan

Jabatan (0.137) x pembobotan jabatan (0.143) maka menghasilkan angka 0.447

## 2.3 Pengertian Website

Website atau situs memiliki pengertian yaitu sekumpulan halaman yang mampu menampilkan teks, gambar diam atau gerak, animasi, suara atau yang menggabungkan semua elemen tersebut, baik yang bersifat statis maupun dinamis yang masing-masing dihubungkan oleh berbagai halaman (Nurmi, 2014). Pengertian website tersebut sejalan dengan keluaran dari penelitian ini yakni berupa website yang berisi informasi pemeringkatan karyawan yang dapat diakses baik oleh divisi SDM (Sumber Daya Manusia) ataupun karyawan.

### 2.3.1 PHP

Dalam pengembangan teknologi *website*, dibutuhkan sebuah bahasa pemrograman supaya *website* bersifat dinamis serta interaktif kepada penggunanya. Dari sekian banyak bahasa pemrograman yang ada, PHP merupakan salah satu yang dinilai sangat cocok untuk pengembangan *website*, terlebih karena bersifat *open source* sehingga mudah untuk dilakukan modifikasi.

Menurut Kustiyaningsih, PHP atau kependekan dari Hyper Preprocessor merupakan sebuah bahasa pemrograman yang berbentuk

skrip-skrip kode dengan tujuan memproses data dan mengirimkan kembali ke dalam bentuk HTML (Firman, Wowor, & Najoran, 2016).

PHP sendiri tidak dapat berjalan tanpa adanya *web server*, karena setiap skrip PHP bersifat *server side*, sehingga dibutuhkan web server untuk mengeksekusi file PHP tersebut.

### **2.3.2 Database**

Salah satu teknologi pendukung dalam pengembangan aplikasi berbasis website adalah penggunaan *database*. Peran *database* sangat vital karena menjadi sumber penyimpanan berbagai bentuk data yang akan digunakan oleh website sebagai konten bagi penggunanya.

*Database* atau basis data merupakan sekumpulan data yang saling berhubungan secara logis yang digunakan secara bersama-sama dengan tujuan untuk memenuhi kebutuhan informasi di suatu tempat (Anisah & Mayasari, 2016).

Dalam pengembangan *website*, salah satu jenis database yang paling sering digunakan adalah *MySQL (My Structure Query Language)* yang merupakan sistem yang bertugas mengatur dan mengelola data pada *database*. *MySQL* merupakan salah satu jenis *database* yang paling banyak digunakan untuk pengembangan *website* karena dikenal efisien dan dapat dipercaya (Hasugian, 2018).

### 2.3.3 HTML

Salah satu komponen dalam pengembangan website adalah HTML (*Hyper Text Markup Language*) yaitu sebuah format penulisan yang dibutuhkan oleh *HTTP Protocol* untuk menampilkan teks dan grafik pada sebuah website (Kassim, Idris, Mohamad, & Suliman, 2018).

### 2.3.4 Bootstrap

Selain HTML, terdapat juga peranan CSS (*Cascading Style Sheet*) yang berfungsi untuk membuat tampilan website semakin menarik dan menambah nilai estetik. Saat ini terdapat sebuah *framework* CSS yang bernama Bootstrap. *Framework* sendiri merupakan sebuah *library* yang digunakan untuk memastikan struktur dan grafis pada website tetap konsisten (*columns, typography, components*) (Laaziri, Benmoussa, Khouilji, Larbi, & Yamami, 2019).

Bootstrap adalah salah satu *framework* untuk CSS yang terdiri dari 12 kolom dengan 4 model penampilan sesuai dengan kebutuhan layar (Laaziri et al., 2019). Dengan menggunakan bootstrap, proses modifikasi CSS dapat dilakukan dengan sendirinya dan sesuai kebutuhan *website*..

### 2.3.5 Message Digest Algorithm

Message digest algorithm juga biasa disebut dengan MD5 yaitu merupakan fungsi hash yang bertujuan untuk mengkonversi string menjadi ke dalam bentuk string yang memiliki panjang tetap. MD5 sendiri merupakan sebuah fungsi matematika yang mengubah variabel dari sebuah ukuran data yang besar menjadi ke ukuran yang lebih sederhana (Sulastri & Putri, 2018).

Penggunaan MD5 dalam website digunakan untuk melakukan hashing pada password sehingga akan menghasilkan keluaran password dengan jenis string yang berbeda dari aslinya guna menghindari bocornya password.

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

Dalam proses pengembangan sistem, tentu diperlukan aktivitas perancangan yang mampu menjadi guideline bagi para pemangku kepentingan untuk mencapai goal yang telah ditetapkan sebelumnya. Salah satu caranya adalah dengan menggunakan UML yang bersifat *object-oriented*.

UML sendiri adalah sebuah alat permodelan *software* yang bersifat *object-oriented* dengan tujuan untuk mempermudah dalam pengembangan *software* (Zheng, Feng, & Zhao, 2014). UML juga memiliki tujuan untuk menjaga komunikasi antar *stakeholder* (*developer, designer, analyst* dll) untuk

berkomunikasi dan meningkatkan pemahaman bersama terkait sistem yang tengah dikembangkan.

### 2.4.1 Activity Diagram

Activity diagram adalah bagian daripada UML yang bertujuan untuk menggambarkan proses bisnis secara cepat dan seefektif mungkin (Alaghbari & Akinin, 2015).

Di dalam use case diagram terdapat beberapa objek yang masing-masing memiliki peran, berikut empat objek pada use case diagram (Dennis, Wixom, & Tegarden, 2015).

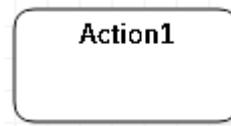
#### 1. Initial



**Gambar 2. 2 Notasi Initial saat ingin memulai pekerjaan.**

Notasi tersebut menandakan bahwa aktivitas dari suatu sistem akan dimulai.

#### 2. Action



**Gambar 2. 3 Notasi Action yang berisikan deskripsi pekerjaan.**

Notasi tersebut menggambarkan aktivitas yang terdapat di dalam sebuah sistem, apabila terdapat banyak aktivitas, maka jumlah ikon tersebut akan menyesuaikan.

3. Decision



**Gambar 2. 4 Notasi Decision yang berfungsi memberikan pilihan.**

Notasi ini menunjukkan apabila terdapat sebuah pilihan antara iya dan tidak, biasanya didahului oleh notasi *Action*.

4. Final



**Gambar 2. 5 Notasi Final yang mengindikasikan bahwa aktivitas telah selesai.**

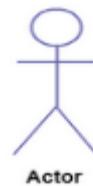
Notasi tersebut menandakan bahwa seluuah aktivitas yang terdapat di dalam sistem telah selesai.

## 2.4.2 Use Case Diagram

*Use case diagram* merupakan seperangkat aktivitas dan fitur yang ditawarkan oleh sistem dan direpresentasikan oleh aktor sehingga dapat dilaksanakan (Alaghbari & Akinin, 2015).

Di dalam *use case diagram* terdapat beberapa objek yang masing-masing memiliki peran, berikut empat objek pada *use case diagram* (Dennis et al., 2015).

### 1. Actor



**Gambar 2. 6** Notasi Actor yang mewakili pemeran aktivitas.

Di dalam sebuah sistem pasti terdapat seseorang atau grup yang melakukan suatu tugas tertentu, maka dari diperlukan penggambaran seperti pada gambar di atas.

## 2. Use Case



**Gambar 2. 7 Notasi Use Case yang berisikan jenis aktivitas.**

*Use case* merupakan penggambaran dari suatu aktivitas yang terdapat di dalam sebuah sistem.

## 3. System



**Gambar 2. 8 Notasi System yang mendefinisikan sistem yang tengah dikembangkan.**

Sistem merupakan gambaran untuk menjelaskan *scope* dari sebuah pengembangan sistem.

#### 4. Association



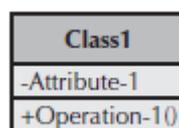
**Gambar 2. 9 Notasi Association yang menunjukkan hubungan antara Actor dan Use Case.**

Notasi panah menunjukkan hubungan pekerjaan antara *actor* dengan masing-masing *use case*.

#### 2.4.3 Class Diagram

*Class diagram* merupakan model yang bersifat statis bertujuan untuk menampilkan kelas-kelas serta relasinya dengan kelas yang lainnya. *Class diagram* juga memiliki notasi tersendiri dalam pembuatannya, berikut ini merupakan notasi-notasi yang terdapat pada *class diagram* (Dennis et al., 2015)

##### 1. Class



**Gambar 2. 10 Notasi Class Pada Class Diagram.**

Sebuah kelas mewakili orang, tempat, atau apapun yang dibutuhkan oleh sistem untuk menangkap dan menyimpan data.

## 2. Attribute Name

attribute name  
/derived attribute name

**Gambar 2. 11 Attribute Name.**

Atribut mewakili isi dari yang kelas tersebut miliki atau untuk menjelaskan peruntukkan dari *class diagram* tersebut.

## 3. Operation Name

operation name ()

**Gambar 2. 12 Operation Name.**

*Operation name* merupakan sebuah fungsi yang kelas tersebut dapat lakukan.

## 4. Association

AssociatedWith  
0..\* 1

**Gambar 2. 13 Association.**

*Association* merupakan relasi yang menunjukkan keterkaitan antara dua kelas.

#### 5. Generalization



**Gambar 2. 14 Generalization.**

*Generalization* merupakan sebuah relasi yang menunjukkan turunan dari sebuah kelas induk.

#### 6. Aggregation



**Gambar 2. 15 Aggregation.**

*Aggregation* merupakan sebuah relasi yang menunjukkan bahwa sebuah kelas ada apabila kelas lainnya ada.

### 2.5 Black Box Testing

*Black box testing* merupakan bagian dari serangkaian pengujian yang dilakukan terhadap suatu sistem untuk memastikan fungsionalitas sistem berjalan dengan yang diharapkan. *Black box testing* sendiri merupakan teknik pengujian

perangkat lunak yang berfokus pada fungsionalitas suatu perangkat lunak, *black box testing* juga tidak memerlukan keahlian mengenai bahasa pemrograman sehingga hanya berfokus kepada informasi domain saja (Jaya, 2018)

Namun *black box testing* memiliki kekurangan antara lain tidak dapat melakukan pemeriksaan terhadap bagian *back-end* dan memiliki kemungkinan pengulangan tes yang telah dilakukan oleh *programmer*.

## **2.6 Metode Pengembangan Sistem**

Menurut Sommerville (Nugroho, Waluyo, & Hakim, 2017), metode pengembangan sistem merupakan bagian dari software process, yang mana setiap metode pengembangan sistem terdiri dari beberapa proses. Terdapat banyak sekali metode pengembangan sistem saat ini sebut saja yang masih tradisional seperti Waterfall hingga yang tengah populer saat ini seperti Agile/Scrum. Karena pada akhirnya setiap penentuan metode pengembangan sistem adalah tergantung dari kebutuhan akan sistem yang akan dibangun.

Pada penelitian ini, RAD (*Rapid Application Development*) dipilih sebagai metode pengembangan sistem yang akan dibangun.

### **2.6.1 Rapid Application Development**

Jenis metode pengembangan sistem yang akan digunakan adalah metode RAD (*Rapid Application Development*) karena memungkinkan

pengerjaan sebuah sistem dalam kurun waktu yang cukup singkat, yang sejalan dengan durasi pembuatan skripsi sekitar 4 bulan.

RAD sendiri merupakan jenis metode pengembangan *software* yang menitikberatkan pada tahap pengembangan dibandingkan dengan tahap desain, hal ini tentu sangat cocok untuk jenis pengembangan sistem yang berskala kecil hingga menengah dengan durasi pengerjaan yang cukup singkat namun dibarengi dengan kualitas yang baik (Despa, 2014).

RAD sendiri terbagi menjadi dua jenis yaitu RAD Incremental & Iterative dan RAD Prototyping. Pada dasarnya, jenis pertama dan kedua serupa karena melibatkan peran *project sponsor* di setiap pembuatannya, sehingga saat kebutuhan *project sponsor* tidak terpenuhi maka dapat langsung dilakukan perubahan.

Perbedaan mendasar antara *incremental & iterative* dan *prototyping* adalah apabila *incremental & iterative* telah dilakukan desain atas sebuah sistem dan akan dikembangkan sesuai dengan urutannya (*design, build, implement, test*). Sedangkan *prototyping*, merupakan siklus metodologi yang terdiri dari *prototyping, testing, analyzing, dan refining*.

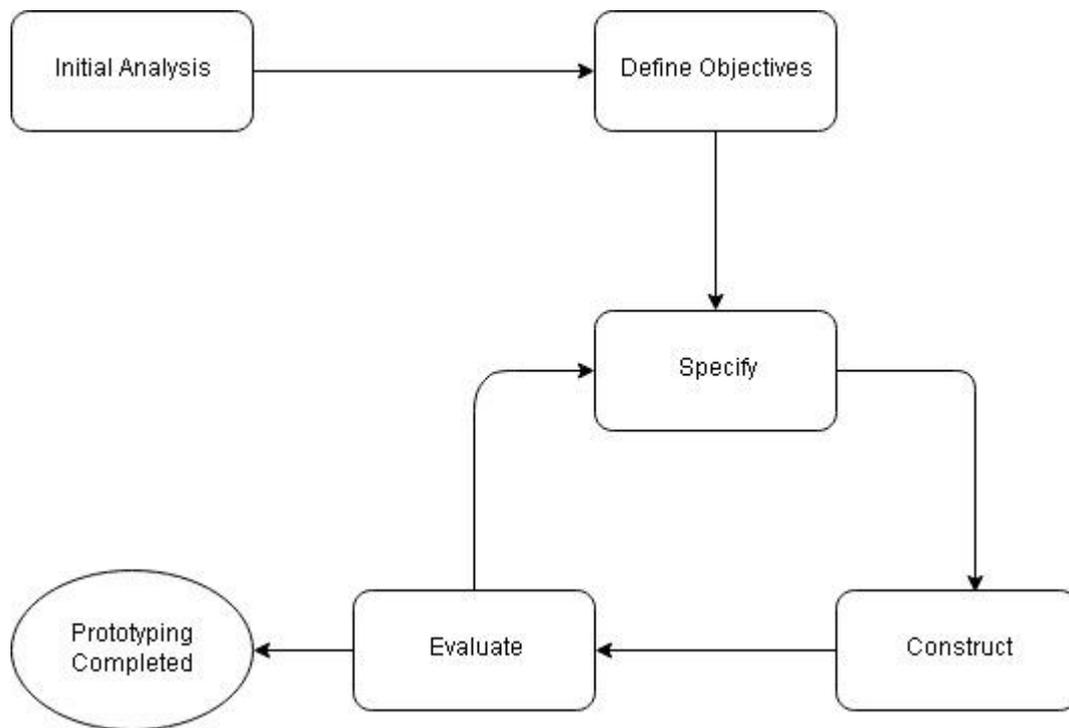
Pada dasarnya, RAD memiliki empat tahapan dalam pengembangan sistem. Berikut foto siklus yang terdapat pada metode pengembangan sistem RAD (Kosasi, 2015).



**Gambar 2. 16 Model RAD secara umum.**

**Sumber: (Kosasi, 2015)**

Namun pada penelitian ini akan digunakan metode RAD Prototyping dengan langkah-langkah sebagai berikut (Bennett, McRobb, & Farmer, 2010):



**Gambar 2. 17 Model RAD Prototyping.**

**Sumber:** (Bennett et al., 2010)

### 1. Initial Analysis

Pada tahap ini, *developer* dan *client* akan bertemu membahas *requirements* seperti apa yang dibutuhkan untuk pengembangan sistem. Namun harus diingat, quick design di sini berbeda dengan design katakanlah pada metode waterfall, karena pada RAD akan bersifat lebih singkat.

## 2. Define Objectives

Selanjutnya adalah baik *developer* dan *client* harus menentukan target apa yang harus dicapai seperti fitur-fitur apa yang diharapkan dari sistem yang akan dibangun.

## 3. Prototype Cycles (Specify, Construct, Evaluate)

Setelah menerima *requirements* dari *client*, selanjutnya adalah membangun aplikasi yang diinginkan dalam bentuk modul dan disajikan kepada *client* apabila telah rampung. Jika *client* merasa belum puas alias belum memenuhi *requirements*, maka siklus akan berulang dari awal lagi.

## 4. Prototyping Completed

Sistem telah selesai dikembangkan dan siap untuk dilakukan implementasi pada *client*.

### **2.7 Mean Opinion Score**

Menurut International Telecommunication Union (ITU) bahwa *Mean Opinion Score* merupakan nilai pada skala yang telah ditetapkan bahwa subjek memberikan pendapatnya mengenai hasil dari sebuah sistem. Penilaian *Mean Opinion Score* sendiri terdiri dari 5 rentang yaitu *bad*, *poor*, *fair*, *good*, dan *excellent* (Fitriyanti, Lindawati, & Aryanti, 2018).

## 2.8 Jurnal Sebelumnya

**Tabel 2. 11 Tabel Daftar Penelitian Terdahulu.**

<b>Nama Peneliti</b>	<b>Nama Jurnal</b>	<b>Judul Penelitian</b>	<b>Hasil Penelitian</b>	<b>Hasil Yang Diambil</b>
I Made Arya Budhi Saputra, Retantyo Wardoyo.	IJCCS, Volume XI, 2017.	Sistem Pendukung Keputusan Kelompok Penentuan  Karyawan Ter  baik Menggunakan Metode Topsis dan  Borda (2017)	Sebuah sistem yang dapat menentukan karyawan terbaik pada Lombok Garden Hotel serta sistem ini bersifat dinamis yang memungkinkan adanya <i>action</i> untuk CRUD.	Bahwa kebutuhan SPK untuk penentu karyawan terbaik masih dibutuhkan bagi beberapa perusahaan.
Ririn Ikana Desanti, Carolyn Feiby Supit, Andree E. Widjaja	Ultima Infosys, Volume VIII, 2017.	Aplikasi Perekrutan dan Penilaian Karyawan Berbasis Web Pada PT. XYZ (2017)	Aplikasi berbasis website yang <i>online</i> dalam menentukan karyawan terbaik dan perekrutan, sehingga dapat dilihat secara <i>real- time</i> karena bersifat <i>online</i> .	Pada jurnal ini yang akan diaplikasikan pada penelitian ini adalah sistem berbasis website untuk penentu karyawan terbaik dan prediksi reward.
Nashrudin Setiawan, MDTP Nasution, Yossie Rossanty, et al.	International Journal of Engineering & Technology, Volume VII, 2018.	Simple Additive Weighting as Decision Support System for  Determining Employees Salary	Hasil penelitian tersebut menghasilkan sebuah keputusan untuk menentukan besaran gaji karyawan menggunakan metode SAW dengan memerlukan beberapa kriteria	Metode penghitungan SAW akan digunakan di dalam penelitian ini karena kasus yang diambil mengenai penentu karyawan terbaik yang juga memerlukan

			pendukung sebagai tolok ukur penilaian.	kriteria tertentu untuk menghasilkan keputusan terbaik.
--	--	--	---	---

Berdasarkan hasil penelitian sebelumnya mengenai topik berkaitan, dalam penelitian ini akan mengadopsi beberapa yang telah dilakukan pada penelitian sebelumnya seperti aplikasi berbasis website, permasalahan mengenai kinerja karyawan serta metode sistem pendukung keputusan yakni SAW (Simple Additive Weighting).

Adapun hal-hal yang belum ada pada penelitian sebelumnya akan diperbaharui pada penelitian ini. Penelitian ini berfokus pada penilaian karyawan menggunakan sistem pendukung keputusan, penelitian ini akan mendapatkan perbedaan seperti fitur prediksi *reward* yang dapat diakses oleh para karyawan supaya dapat meningkatkan motivasi mereka.

