



Article DOI: 10.36350/jbs.v13i2.217
Received: May; Accepted: June; Published: July

Penerapan Metode *Simple Additive Weighting* (SAW) Untuk Rekomendasi Penentuan Penerimaan *Backend Engineer*

Mellyza Rismawan¹, Muhammad Miftahudin^{2*}, Rajib Ghaniy³

¹Sistem Informasi/Universitas Binaniga Indonesia
Email: mellyza724@gmail.com

²Teknik Informatika/Universitas Binaniga Indonesia
Email: m.miftahudin@unbin.ac.id

³Sistem Informasi/Universitas Binaniga Indonesia
Email: rajob@unbin.ac.id

*) *Corresponding Author*

ABSTRACT

Determining the acceptance of candidate backend engineers will go through various stages and an assessment process based on certain predetermined criteria or requirements. However, the problem faced at this time is the result of an assessment that has almost the same score weight, close to the same or even exactly same, of course this will make it difficult for the manager concerned to make a decision. Therefore, the implementation of a Decision Support System (DSS) using the Simple Additive Weighting (SAW) method on this problem is one of the right and effective ways. The decision support system created in this study has successfully implemented the SAW method based on candidate backend engineer data which has been processed using 8 criteria, namely Golang, Test Results, Code Readability, Documentation, Unit Tests, Tech Experience, Design System, Code Scalability. And the ranking results have been tested using Spearman Rank correlation calculations and yielded a value of 0.98, which means that the SAW method can be used to determine recommendations for accepting backend engineer candidates.

Keywords: *Decision Support System, Simple Additive Weighting, Backend Engineer, SAW, Spearman Rank, Recruitment*

ABSTRAK

Penentuan penerimaan kandidat backend engineer akan melewati berbagai tahapan dan proses penilaian berdasarkan kriteria atau persyaratan tertentu yang sudah ditetapkan. Namun permasalahan yang dihadapi saat ini adalah hasil dari penilaian yang memiliki bobot skor yang hampir sama, mendekati sama bahkan sama persis, tentu hal tersebut akan menyulitkan manager yang bersangkutan dalam mengambil keputusan. Oleh karena itu, penerapan Sistem Pendukung Keputusan (SPK) dengan menggunakan metode Simple Additive Weighting (SAW) pada permasalahan ini merupakan salah satu cara yang tepat dan efektif. Sistem Pendukung keputusan yang dibuat pada penelitian ini berhasil mengimplementasikan metode SAW berdasarkan data kandidat backend engineer yang sudah diolah menggunakan 8 kriteria yaitu Golang, Hasil Test, Code Readability, Dokumentasi, Unit Test, Tech Experience, Design System, Code Scalability. Dan hasil perankingan telah diuji dengan

menggunakan perhitungan korelasi Spearman Rank dan menghasilkan nilai 0.98, yang berarti metode SAW dapat digunakan untuk penentuan rekomendasi penerimaan kandidat backend engineer.

Keywords: Sistem Pendukung Keputusan, Simple Additive Weighting, Backend Engineer, SAW, PSSUQ, Spearman Rank, Pemilihan Backend Engineer

A. PENDAHULUAN

1. Latar Belakang

Didalam perusahaan tentunya terdapat sumber daya manusia (SDM) yang merupakan hal yang sangat penting untuk mendukung kemajuan dan kualitas suatu instansi atau organisasi dalam mencapai tujuan. Salah satu kegiatan yang menjadi peran terhadap kesuksesan atau tidaknya suatu perusahaan adalah pada saat penerimaan karyawan baru. Proses penerimaan karyawan merupakan tahap yang paling strategis untuk mencari SDM yang tepat untuk perusahaan. Pada sebuah perusahaan biasanya terdapat banyak jenis profesi yang ada. Beberapa jenis profesi yang ada saat ini adalah HRD, CEO, CMO, *Software Engineer* yang di dalamnya terdapat juga profesi *Backend Engineer*, *Frontend Engineer*, *Mobile Engineer* dan lain-lain. Saat ini untuk memilih dan memilah para calon *Backend Engineer*, suatu instansi harus kompetitif dalam mencari dan menerima para calon *Backend Engineer*, dengan mempertimbangkan segala aspek / kriteria yang sudah ditentukan. Proses penerimaan *Backend Engineer* tentunya tidak dapat berjalan secara cepat karena harus menghitung banyak komponen hasil tes untuk menguji kemampuan dalam membuat program. Dengan terpilihnya *Backend Engineer* yang berkualitas serta terdidik ini, nanti nya *Backend Engineer* ini akan bekerja untuk menunjang nilai suatu instansi dan bisa bersaing dengan instansi lainnya.

2. Permasalahan

Berdasarkan data pada Table 1 terlihat hasil perhitungan yang diperoleh dari menjumlahkan masing - masing penilaian kriteria dari setiap alternatif yang sudah disamakan kemudian membaginya dengan banyaknya jumlah alternatif yang ada. Sistem penilaian yang sedang berjalan tersebut dapat dikatakan sudah cukup baik tetapi masih ada kekurangan, yaitu belum ada nya rangking serta masih kurang tepatnya penggunaan kriteria yang di berikan untuk penilaian penerimaan kandidat *Backend Engineer*. Ada pun kriteria lain yang dapat digunakan untuk pengambilan keputusan penerimaan kandidat *Backend Engineer* yakni: pengalaman akan teknologi (*tech experience*), kemampuan dalam merancang sistem (*design system*), serta kemampuan dalam membuat code untuk menanggapi beban yang diberikan (*code scalability*). Penilaian penerimaan kandidat *Backend Engineer* yang sedang berjalan saat ini juga belum tepat karena belum ada nya nilai dari bobot setiap kriteria yang berikan, serta tidak jarang masih ada nilai yang sama dari setiap alternatif seperti yang sudah diberikan seperti pada contoh tabel diatas, kandidat yang ditandai dengan warna merah memiliki nilai yang sama dengan kandidat yang ditandai dengan warna biru. Hal ini dapat menyulitkan manajemen atau manager divisi dalam menentukan kandidat *Backend Engineer* yang akan dipertimbangkan untuk diterima diperusahaan. Dari permasalahan yang sudah disebutkan maka dapat dikatakan pendataan ini belum tepat serta belum efektif dalam pelaksanaannya.

Tabel 1 Daftar Kandidat *Backend Engineer* dan Nilai dari Kriteria

Periode: Januari 2021							
No	Nama	Golang	Hasil Tes	Code Readability	Dokumentasi	Unit Test	Rata-rata
1	Adam	86	90	80	75	50	76.2
2	Ben	86	90	80	80	25	72.2

3	Bobby	85	89	83	80	88	85
4	Dedi	90	80	78	50	80	75.6
5	Felix	83	70	66	70	60	69.8
6	Dadan	78	80	78	60	90	77.2
7	Citra	85	89	83	80	88	85
8	Putri	82	89	79	87	65	80.4
9	Bayu	65	74	70	50	25	56.8

Berdasarkan permasalahan yang telah diuraikan, maka didapatkan identifikasi masalah yaitu sebagai berikut:

- a. Belum tepatnya dalam penetapan *Backend Engineer* di divisi IT perusahaan.
- b. Belum efektifnya proses penilaian dalam pengambilan keputusan penerimaan *Backend Engineer* di divisi IT perusahaan.

3. Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

- a. Mendapatkan *Backend Engineer* yang tepat sesuai dengan kriteria yang ditetapkan.
- b. Mendapatkan proses yang lebih efektif didalam pengambilan keputusan penerimaan *Backend Engineer*.
- c. Mengukur tingkat ketepatan dan efektifitas penerapan metode *Simple Additive Weighting* (SAW) untuk pengambilan keputusan penerimaan *Backend Engineer*.

4. Tinjauan Pustaka

a. Sistem Pendukung Keputusan

Menurut (Warmansyah, 2020, p.112) Sistem Pendukung Keputusan merupakan suatu sistem yang digunakan untuk mempermudah pengambil keputusan dan hasil yang didapat melalui SPK tidak sepenuhnya harus digunakan untuk menyelesaikan sebuah masalah Tahapan-tahapan pengambilan keputusan menurut Simon (1960) adalah:

- 1) *Intelligence*, Merupakan kumpulan informasi untuk mengidentifikasi masalah.
- 2) *Design*, Merupakan tahapan merancang solusi berupa alternatif pada pemecahan masalah.
- 3) *sChoice*, Ini adalah Langkah memilih dari solusi alternatif yang disediakan.
- 4) *Implementation*, Fase implementasi adalah fase dimana keputusan telah diimplementasikan. Pada tahap ini, perlu dikembangkan serangkaian Tindakan yang direncanakan agar hasil keputusan dapat dimonitor dan disesuaikan jika diperlukan kebaikan.

b. Metode SAW

Metode *Simple Additive Weighting* (SAW) adalah salah satu metode yang digunakan dalam proses pengambilan suatu keputusan. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Menurut (Warmansyah, 2020, p.67) metode SAW ini menggunakan pembobotan pada masing-masing kriteria, setiap pembobotan diukur untuk mendapatkan nilai penjumlahan pada setiap alternatif yang akan dipilih.

Menurut (Warmansyah, 2020 pp. 67-68) ada beberapa langkah yang harus diperhatikan dalam algoritma dari metode SAW antara lain adalah:

“Mengevaluasi alternative A terhadap sekumpulan atribut atau kriteria C di mana setiap atribut saling tidak bergantung. Matriks keputusan X dibentuk dari rating kinerja alternative x dan nilai bobot yang menunjukkan kepentingan relatif setiap atribut W. Proses diakhiri dengan perankingan untuk mendapatkan alternative terbaik.”

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} \rightarrow \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} \rightarrow \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

cost merupakan jenis kriteria yang mengutamakan nilai terendah, sedangkan *benefit* merupakan jenis kriteria yang mengutamakan nilai tertinggi sebagai acuan pemilihan; r_{ij} adalah kinerja ternormalisasi dari alternative A_i pada atribut;

$$C_j: i = 1, 2, \dots, m \text{ dan } j = 1, 2, \dots, n$$

nilai prefrensi untuk setiap alternative (V_i) diberikan sebagai berikut:

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

nilai V_i yang lebih besar mengidentifikasi bahwa alternatif A_i lebih terpilih

W : Bobot (kriteria)

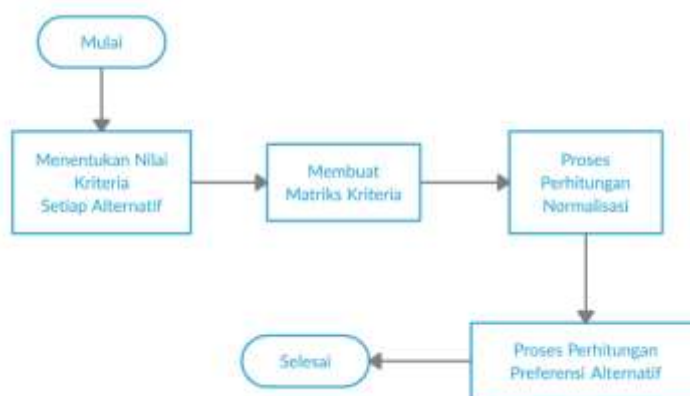
R : Nilai dari setiap peserta untuk tiap kriteria Dengan kata lain antara bobot kriteria (**W**) dikalikan dengan semua nilai tiap peserta (r) untuk tiap kriteria dan dijumlahkan;

B. METODE

1. Metode yang diusulkan

Model konseptual yang diusulkan dalam penelitian ini yaitu metode *Simple Additive Weighting*. Metode SAW merupakan metode yang juga dikenal dengan metode penjumlahan terbobot, dimana konsep dasar SAW mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967). Proses tersebut digambarkan pada alur sebagai berikut:

- Menentukan Nilai Kriteria Setiap Alternatif.
- Membuat Matriks Kriteria.
- Proses Perhitungan Normalisasi.
- Proses Perhitungan Prefrensi Alternatif



Gambar 1. Alur Proses Metode SAW

2. Teknik Analisis Data

Untuk menentukan tingkat keakuratan dan ketepatan pada hasil penelitian ini, maka digunakan uji *spearman rank*. Hasil akhir dari uji korelasi *Spearman* biasanya berupa angka-angka yang kemudian bisa dikategorikan dalam beberapa hubungan. Kemudian dapat dilihat seberapa signifikan hubungan yang terjadi, bagaimana satu variabel sangat mempengaruhi atau bahkan tidak berpengaruh sama sekali terhadap variabel lainnya. Persamaan dari *spearman rank* dapat dilihat sebagai berikut:

$$p = 1 - \frac{6 \sum b_i^2}{n(n^2 - 1)}$$

Nilai hasil uji korelasi antara output SAW dengan hasil para ahli dapat digunakan untuk menilai ketepatan sistem berdasarkan tabel makna *Spearman*.

Tabel 2. Korelasi Rank Spearman

Nilai	Interpretasi
0,00 – 0,25	Hubungan Sangat Rendah
0,26 – 0,50	Hubungan Cukup
0,51 – 0,75	Hubungan Kuat
0,76 – 0,99	Hubungan Sangat Kuat
1	Hubungan Sempurna

Sumber: (Sugiyono, 2016, p. 214)

C. HASIL DAN PEMBAHASAN

1. Hasil

a. Analisis Metode

Dalam penentuan penerimaan *Backend Engineer* menggunakan metode SAW diperlukan kriteria – kriteria dan juga bobot untuk melakukan perhitungan sehingga nantinya mendapatkan alternatif terbaik sesuai dengan penilaian yang diinputkan kedalam sistem.

1) Menentukan Kriteria dan Bobot Kriteria

a) Kriteria

Kriteria merupakan tahapan menentukan ukuran yang menjadi dasar penilaian dalam menentukan rekomendasi penentuan penerimaan *backend engineer*, dimana kriteria penilaian dalam penelitian ini terdiri dari 8 (delapan) kriteria yang dimana bobot dari masing-masing kriteria sudah ditentukan oleh pihak yang memang ahli dan kompeten dari pengguna aplikasi nantinya. Adapun penentuan 3 kriteria tambahan dan besarnya bobot kriteria di dapatkan melalui wawancara kepada *Backend Engineering Manager* selaku ahli pengguna dan menghasilkan kriteria rekomendasi penentuan penerimaan *backend engineer* sebagai berikut:

Tabel 3. Kriteria

ID Kriteria	Keterangan
1	<i>Golang</i>
2	<i>Hasil Test</i>
3	<i>Code Readability</i>
4	Dokumentasi
5	<i>Unit Test</i>
6	<i>Tech Experience</i>

b) Bobot Kriteria

Dari setiap kriteria yang ada akan ditentukan tingkat kepentingan masing – masing kriteria tersebut. Pada penelitian ini akan diberikan nilai 5 – 100.

(1) *Golang*

Tabel 4. Bobot *Golang*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai <i>Golang</i> 1 - 20
25	Buruk	Jika nilai <i>Golang</i> 21 - 40

50	Cukup	Jika nilai <i>Golang</i> 41 - 60
75	Baik	Jika nilai <i>Golang</i> 61 - 80
100	Sangat Baik	Jika nilai <i>Golang</i> 81 - 100

(2) Hasil *Test*

Tabel 5. Bobot Hasil *Test*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai hasil tes 1 - 20
25	Buruk	Jika nilai hasil tes 21 - 40
50	Cukup	Jika nilai hasil tes 41 - 60
75	Baik	Jika nilai hasil tes 61 - 80
100	Sangat Baik	Jika nilai hasil tes 81 - 100

(3) *Code Readability*

Tabel 6. Bobot *Code Readability*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai <i>Code Readability</i> 1 - 20
25	Buruk	Jika nilai <i>Code Readability</i> 21 - 40
50	Cukup	Jika nilai <i>Code Readability</i> 41 - 60
75	Baik	Jika nilai <i>Code Readability</i> 61 - 80
100	Sangat Baik	Jika nilai <i>Code Readability</i> 81 - 100

(4) Dokumentasi

Tabel 7. Bobot Dokumentasi

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai Dokumentasi 1 - 20
25	Buruk	Jika nilai Dokumentasi 21 - 40
50	Cukup	Jika nilai Dokumentasi 41 - 60
75	Baik	Jika nilai Dokumentasi 61 - 80
100	Sangat Baik	Jika nilai Dokumentasi 81 - 100

(5) Unit *Test*

Tabel 8. Bobot Unit *Test*

Bobot Nilai	Detail Nilai	Keterangan
5	Presentase $\leq 10\%$	Jika presentase <i>Unit Test</i> 1 - 10%
25	Presentase $> 10\%$	Jika presentase <i>Unit Test</i> 11 - 29 %
50	Presentase $\geq 30\%$	Jika presentase <i>Unit Test</i> 30 - 39 %
75	Presentase $\geq 40\%$	Jika presentase <i>Unit Test</i> 40 - 59 %
100	Presentase $\geq 60\%$	Jika presentase <i>Unit Test</i> 60 - 100%

(6) *Tech Experience*

Tabel 9. Bobot *Tech Experience*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai <i>Tech Experience</i> 1 - 20
25	Buruk	Jika nilai <i>Tech Experience</i> 21 - 40

50	Cukup	Jika nilai <i>Tech Experience</i> 41 - 60
75	Baik	Jika nilai <i>Tech Experience</i> 61 - 80
100	Sangat Baik	Jika nilai <i>Tech Experience</i> 81 - 100

(7) *Design System*

Tabel 10. Bobot *Design System*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai <i>Design System</i> 1 - 20
25	Buruk	Jika nilai <i>Design System</i> 21 - 40
50	Cukup	Jika nilai <i>Design System</i> 41 - 60
75	Baik	Jika nilai <i>Design System</i> 61 - 80
100	Sangat Baik	Jika nilai <i>Design System</i> 81 - 100

(8) *Code Scalability*

Tabel 11. Bobot *Code Scalability*

Bobot Nilai	Detail Nilai	Keterangan
5	Sangat Buruk	Jika nilai <i>Code Scalability</i> 1 - 20
25	Buruk	Jika nilai <i>Code Scalability</i> 21 - 40
50	Cukup	Jika nilai <i>Code Scalability</i> 41 - 60
75	Baik	Jika nilai <i>Code Scalability</i> 61 - 80
100	Sangat Baik	Jika nilai <i>Code Scalability</i> 81 - 100

(9) Menentukan bobot (W) setiap Kriteria

Tabel 12. Bobot Setiap Kriteria

ID Kriteria	Bobot	Kriteria
1	20	Hasil Tes
2	10	<i>Code Readability</i>
3	15	<i>Golang</i>
4	5	Dokumentasi
5	10	<i>Unit Test</i>
6	20	<i>Tech Experience</i>

2) Data Alternatif

Data alternatif yang akan digunakan pada penelitian ini didapatkan dari hasil wawancara dengan objek penelitian dan data tersebut adalah data asli yang sudah disamarkan dari perusahaan untuk mendukung penelitian ini. Adapun data yang digunakan adalah sebagai berikut:

Tabel 13. Data Alternatif

ID Alternatif	Nama Alternatif
1	Adam
2	Ben
3	Bobby
4	Dedi
5	Felix

6	Dadan
7	Citra
8	Putri
9	Bayu

3) Data Penilaian Kandidat *Backend Engineer*

Data penilaian kandidat *Backend Engineer* yang digunakan pada penelitian ini merupakan data kandidat pada tahun 2021 yang dideskripsikan pada tabel dibawah:

Tabel 14. Data Penilaian Kandidat *Backend Engineer*

Nama	Golang	Hasil Test	Code Readability	Dokumentasi	Unit Test	Tech Experience	Design System	Code Scalability
Adam	86	90	80	75	50	86	79	78
Ben	86	90	80	80	25	80	80	80
Bobby	85	89	83	80	88	88	85	83
Dedi	90	80	78	50	80	88	75	70
Felix	83	70	66	70	60	75	75	50
Dadan	78	80	78	60	90	83	80	78
Citra	85	89	83	80	88	80	78	80
Putri	82	89	79	87	65	85	60	60
Bayu	65	74	70	50	25	75	50	25

b. Analisis Kasus dengan Metode SAW

Analisis kasus dengan menggunakan metode SAW pada penelitian ini diambil langsung menggunakan nilai dari tabel 14 yaitu data penilaian *Backend Engineer* serta tidak menggunakan nilai crips atau bobot nilai yang sudah ditentukan sebagai acuan perhitungan.

1) Melakukan Normalisasi Setiap Alternatif

a) Adam

$$r_{\text{Golang}} = \frac{86}{\max(86,86,85,90,83,78,85,82,65)} = 0.956$$

$$r_{\text{Unit Test}} = \frac{50}{\max(50,25,88,80,60,90,88,65,25)} = 0.556$$

$$r_{\text{Hasil Tes}} = \frac{90}{\max(90,90,89,80,70,80,89,89,74)} = 1$$

$$r_{\text{Tech Exp}} = \frac{86}{\max(86,80,88,88,75,83,80,85,75)} = 0.977$$

$$r_{\text{Code Readability}} = \frac{80}{\max(80,80,83,78,66,78,83,79,70)} = 0.963$$

$$r_{\text{Design System}} = \frac{79}{\max(79,80,85,75,75,80,78,60,50)} = 0.929$$

$$r_{\text{Dokumentasi}} = \frac{75}{\max(75,80,80,50,70,60,80,87,50)} = 0.862$$

$$r_{\text{Code Scalability}} = \frac{78}{\max(78,80,83,70,50,78,80,60,25)} = 0.939$$

b) Ben

$$r_{\text{Golang}} = \frac{86}{\max(86,86,85,90,83,78,85,82,65)} = 0.956$$

$$r_{\text{Unit Test}} = \frac{25}{\max(50,25,88,80,60,90,88,65,25)} = 0.278$$

$$r_{\text{Hasil Tes}} = \frac{90}{\max(90,90,89,80,70,80,89,89,74)} = 1$$

$$r_{\text{Tech Exp}} = \frac{80}{\max(86,80,88,88,75,83,80,85,75)} = 0.909$$

$$r_{\text{Code Readability}} = \frac{80}{\max(80,80,83,78,66,78,83,79,70)} = 0.963$$

$$r_{\text{Design System}} = \frac{80}{\max(79,80,85,75,75,80,78,60,50)} = 0.941$$

$$\begin{aligned} & \max(80,80,83,78,66,78,83,79,70) & \max(79,80,85,75,75,80,78,60,50) \\ & 80 & 80 \\ r_{Dokumentasi} &= \frac{80}{\max(75,80,80,50,70,60,80,87,50)} = \mathbf{0.919} & r_{CodeScalability} &= \frac{80}{\max(78,80,83,70,50,78,80,60,25)} = \mathbf{0.963} \end{aligned}$$

c) Bobby

$$\begin{aligned} r_{Golang} &= \frac{85}{\max(86,86,85,90,83,78,85,82,65)} = \mathbf{0.944} & r_{Unit\ Test} &= \frac{88}{\max(50,25,88,80,60,90,88,65,25)} = \mathbf{0.978} \\ r_{Hasil\ Tes} &= \frac{89}{\max(90,90,89,80,70,80,89,89,74)} = \mathbf{0.989} & r_{Tech\ Exp} &= \frac{88}{\max(86,80,88,88,75,83,80,85,75)} = \mathbf{1} \\ r_{Code\ Readability} &= \frac{83}{\max(80,80,83,78,66,78,83,79,70)} = \mathbf{1} & r_{Design\ System} &= \frac{85}{\max(79,80,85,75,75,80,78,60,50)} = \mathbf{1} \\ r_{Dokumentasi} &= \frac{80}{\max(75,80,80,50,70,60,80,87,50)} = \mathbf{0.919} & r_{Code\ Scalability} &= \frac{83}{\max(78,80,83,70,50,78,80,60,25)} = \mathbf{1} \end{aligned}$$

2) Membuat Matrik Normalisasi

$$R = \begin{bmatrix} 0.956 & 1 & 0.963 & 0.862 & 0.556 & 0.977 & 0.929 & 0.939 \\ 0.956 & 1 & 0.963 & 0.919 & 0.278 & 0.909 & 0.941 & 0.963 \\ 0.944 & 0.988 & 1 & 0.919 & 0.978 & 1 & 1 & 1 \\ 1 & 0.889 & 0.939 & 0.574 & 0.889 & 1 & 0.882 & 0.843 \\ 0.922 & 0.778 & 0.795 & 0.804 & 0.667 & 0.852 & 0.882 & 0.602 \\ 0.867 & 0.889 & 0.939 & 0.689 & 1 & 0.943 & 0.941 & 10.939 \\ 0.944 & 0.989 & 1 & 0.919 & 0.978 & 0.909 & 0.917 & 0.963 \\ 0.911 & 0.989 & 0.951 & 1 & 0.722 & 0.965 & 0.705 & 0.722 \\ 0.722 & 0.822 & 0.843 & 0.574 & 0.278 & 0.852 & 0.588 & 0.301 \end{bmatrix}$$

3) Menghitung Nilai Preferensi (V)

$$V_1 = (15 \times 0.956) + (20 \times 1) + (10 \times 0.963) + (5 \times 0.862) + (10 \times 0.556) + (20 \times 0.977) + (10 \times 0.929) + (10 \times 0.939) = \mathbf{92.0749}$$

$$V_2 = (15 \times 0.956) + (20 \times 1) + (10 \times 0.963) + (5 \times 0.919) + (10 \times 0.278) + (20 \times 0.909) + (10 \times 0.941) + (10 \times 0.963) = \mathbf{88.579}$$

$$V_3 = (15 \times 0.944) + (20 \times 0.989) + (10 \times 1) + (5 \times 0.919) + (10 \times 0.978) + (20 \times 1) + (10 \times 1) + (10 \times 1) = \mathbf{98.319}$$

$$V_4 = (15 \times 1) + (20 \times 0.889) + (10 \times 0.939) + (5 \times 0.574) + (10 \times 0.889) + (20 \times 1) + (10 \times 0.882) + (10 \times 0.843) = \mathbf{91.195}$$

$$V_5 = (15 \times 0.922) + (20 \times 0.778) + (10 \times 0.795) + (5 \times 0.804) + (10 \times 0.667) + (20 \times 0.852) + (10 \times 0.882) + (10 \times 0.602) = \mathbf{79.923}$$

$$V_6 = (15 \times 0.867) + (20 \times 0.889) + (10 \times 0.939) + (5 \times 0.689) + (10 \times 1) + (20 \times 0.943) + (10 \times 0.941) + (10 \times 0.939) = \mathbf{91.296}$$

$$V_7 = (15 \times 0.944) + (20 \times 0.989) + (10 \times 1) + (5 \times 0.919) + (10 \times 0.978) \\ + (20 \times 0.909) + (10 \times 0.917) + (10 \times 0.963) = \mathbf{95.316}$$

$$V_8 = (15 \times 0.911) + (20 \times 0.989) + (10 \times 0.951) + (5 \times 1) + (10 \times 0.722) \\ + (20 \times 0.965) + (10 \times 0.705) + (10 \times 0.722) = \mathbf{88.790}$$

$$V_9 = (15 \times 0.722) + (20 \times 0.822) + (10 \times 0.843) + (5 \times 0.574) + (10 \times 0.278) \\ + (20 \times 0.852) + (10 \times 0.588) + (10 \times 0.301) = \mathbf{67.302}$$

4) Menentukan Nilai Preferensi (V)

Tabel 15. Nilai Preferensi (V)

Alternatif	Nama Alternatif	Hasil Preferensi
V_1	Adam	92.074
V_2	Ben	88.579
V_3	Bobby	98.319
V_4	Dedi	91.195
V_5	Felix	79.923
V_6	Dadan	91.296
V_7	Citra	95.316
V_8	Putri	88.790
V_9	Bayu	67.302

Dari hasil perhitungan preferensi diatas, didapat bahwa nilai alternatif tertinggi di dapatkan oleh V_3 (**Bobby**). Jadi yang dapat di pertimbangkan untuk di terima menjadi *Backend Engineer* adalah kandidat bernama **Bobby**.

2. Pembahasan

Uji hasil dari penelitian ini yaitu dengan menggunakan *Rank Spearman* dengan membandingkan hasil ranking dari perhitungan setiap nilai alternatif yang sudah menggunakan 8 (delapan) kriteria dengan cara menggunakan rata – rata dan membandingkan nya dengan hasil perhitungan menggunakan aplikasi yang sudah menerapkan metode SAW. Berikut adalah hasil perhitungan nilai alternatif menggunakan rata – rata sebagaimana dapat di lihat pada tabel 16:

Tabel 16 Perhitungan Nilai Alternatif Menggunakan Rata - Rata

Nama	Golang	Hasil Test	Code Readability	Dokumentasi	Unit Test	Tech Experience	Design System	Code Scalability	Rata - Rata	Rank
Adam	86	90	80	75	50	86	79	78	69.333	4
Ben	86	90	80	80	25	80	80	80	66.778	7
Bobby	85	89	83	80	88	88	85	83	75.667	1
Dedi	90	80	78	50	80	88	75	70	67.889	5
Felix	83	70	66	70	60	75	75	50	61	8
Dadan	78	80	78	60	90	83	80	78	69.667	3
Citra	85	89	83	80	88	80	78	80	73.667	2
Putri	82	89	79	87	65	85	60	60	67.444	6
Bayu	65	74	70	50	25	75	50	25	48.222	9

Tabel 17 berikut adalah hasil dari perhitungan Korelasi *Rank Spearman*:

Tabel 17 Perhitungan Korelasi Spearman Rank

Kode Alternatif	Ranking		d (X-Y)	d ² (X-Y) ²
	X	Y		
Adam	4	3	1	1
Ben	7	7	0	0
Bobby	1	1	0	0
Dedi	5	5	0	0
Felix	8	8	0	0
Dadan	3	4	-1	1
Citra	2	2	0	0
Putri	6	6	0	0
Bayu	9	9	0	0
Jumlah ($\sum d^2$)				2

Keterangan:

X = Ranking sebelum penerapan metode

Y = Ranking sesudah penerapan metode

Pada tabel 17 diketahui bahwa jumlah dari perhitungan ranking sebelum dan sesudah menggunakan aplikasi, selanjutnya dihitung menggunakan rumus korelasi *Rank Spearman* adalah sebagai berikut:

$$rs = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

$$rs = 1 - \frac{6 \times 2}{9(9^2 - 1)}$$

$$rs = 1 - \frac{12}{720}$$

$$rs = 1 - 0.0166$$

$$rs = 0.98$$

Keterangan:

Nilai Interval Koefisien : 0.98

Tingkat Hubungan : Sangat Kuat

Hasil Interval Koefisien *Rank Spearman* memperoleh nilai sebesar 0.98 dan berdasarkan korelasi rank spearman, nilai tersebut memiliki tingkat **Hubungan** yang **Sangat Kuat**.

D. KESIMPULAN

Berdasarkan pada hasil penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan yaitu sebagai berikut:

1. Penentuan ranking rekomendasi penerimaan *backend engineer* dengan menggunakan metode SAW dapat membantu perusahaan dalam memberikan ketepatan serta keefisienan dalam proses penentuan penerimaan *backend engineer* pada periode yang akan dilakukan.
2. Pengukuran ketepatan dan keefektifan metode SAW dengan menggunakan Spearman Rank dalam memberikan rekomendasi penentuan penerimaan *backend engineer* mendapatkan nilai 0.98, maka dapat disimpulkan bahwa ada perubahan yang "Sangat Kuat" terhadap penelitian

yang dilakukan dengan membandingkan hasil ranking sebelum menggunakan metode dan sesudah menggunakan metode.

E. DAFTAR PUSTAKA

- [1] Arikunto, S. (2009). *Prosedur Penelitian Suatu Pendekatan Praktik* (Revisi 6). Rineka Cipta
- [2] Diana. (2018). *Metode dan Aplikasi Sistem Pendukung Keputusan*. Deepublish. <https://books.google.co.id/books?id=nJSEDwAAQBAJ&printsec>
- [3] Fishburn, P. C. (1967). A Problem-based selection of multi-attribute decision making methods, Blackwell Publishing. *Management Science*, 38(5).
- [4] Pibriana, D. (2020). Penggunaan Metode Simple Additive Weighting (SAW) dalam Pengambilan Keputusan Rekrutmen Karyawan Pada PT. ABC. *Techno.Com*, 19(1). <https://doi.org/10.33633/tc.v19i1.2771>
- [5] Prayogo, T., Cleopatra, M., & Irawan, A. (2020). Aplikasi Sistem Pendukung Keputusan Penerimaan Karyawan Baru dengan Metode Simple Additive Weighting (SAW). *Jurnal Informatika Universitas Pamulang*, 5(1). <https://doi.org/10.32493/informatika.v5i1.4119>
- [6] Raymond Mcleod, J., & George P. Schell. (2007). *Management Informasi Sytem*. PERSONE Education
- [7] Sugiyono. (2019). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif dan R&D*. Alfabeta.
- [8] Sukamto, R. A., & Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak*. Informatika Bandung
- [9] Triharseno, W., Pradnya Duhita, W. M., & Priadana, A. (2020). Sistem Pendukung Keputusan Penerimaan Programmer Software House Menggunakan Metode Simple Additive Weighting (SAW). *JURNAL PILAR TEKNOLOGI : Jurnal Ilmiah Ilmu Ilmu Teknik*, 5(1). <https://doi.org/10.33319/piltek.v5i1.52>
- [10] Warmansyah, J. (2020). *Metode Penelitian dan Pengolahan Data untuk Pengambilan Keputusan pada Perusahaan*. Depublish Publisher.