**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Pertumbuhan industri teknologi informasi, khususnya di Jakarta, semakin pesat dengan munculnya perusahaan-perusahaan teknologi yang terus berkembang. PT. Rackh Lintas Asia sebagai salah satu pemain utama dalam industri ini, terus menghadapi tantangan untuk menjaga dan meningkatkan kualitas layanan teknologinya. Dalam upaya tersebut, rekrutmen s*ystem engineer* menjadi kunci strategis untuk memastikan bahwa perusahaan memiliki tim yang terampil dan berkualitas.

Rekrutmen s*ystem engineer* bukanlah tugas yang mudah, terutama dengan pertumbuhan kompleksitas tuntutan teknologi. Perusahaan tidak hanya perlu memperhatikan keterampilan teknis, tetapi juga aspek kepribadian dan kemampuan beradaptasi terhadap perubahan. Oleh karena itu, diperlukan pendekatan yang sistematis dan terstruktur dalam proses rekrutmen untuk memastikan bahwa kandidat yang terpilih memiliki kecocokan yang optimal dengan kebutuhan perusahaan.

Dalam menghadapi kompleksitas rekrutmen *system engineer*, penggunaan Sistem Pendukung Keputusan (SPK) menjadi semakin relevan. Penelitian ini bertujuan untuk menerapkan metode ARAS dalam konteks rekrutmen s*ystem engineer* di PT Rackh Lintas Asia cabang Jakarta untuk meningkatkan efisiensi dan efektivitas dalam memilih kandidat terbaik.

Selain itu, PT Rackh Lintas Asia juga dihadapkan pada persaingan ketat di pasar kerja teknologi informasi. Dalam rangka menarik dan mempertahankan bakat terbaik, perusahaan perlu menunjukkan komitmen terhadap inovasi dan pengembangan karir. Penggunaan sistem pendukung keputusan dengan pendekatan metode ARAS tidak hanya dapat membantu dalam pemilihan kandidat, tetapi juga menciptakan citra perusahaan sebagai organisasi yang menggunakan teknologi untuk meningkatkan proses bisnisnya.

Penelitian ini juga diilhami oleh kebutuhan untuk meningkatkan transparansi dalam proses rekrutmen. Dengan menerapkan metode ARAS, perusahaan dapat dengan jelas mengidentifikasi faktor-faktor yang paling penting dalam pengambilan keputusan rekrutmen. Hal ini dapat memberikan pemahaman yang lebih baik kepada pihak terkait tentang mengapa seorang kandidat dipilih atau ditolak, mengurangi potensi ketidakpastian dan meningkatkan akseptabilitas keputusan.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi nyata terhadap peningkatan proses rekrutmen *system engineer* di PT Rackh Lintas Asia cabang Jakarta. Melalui implementasi Sistem Pendukung Keputusan dengan pendekatan metode ARAS, perusahaan dapat lebih efektif dan efisien dalam memilih kandidat yang sesuai dengan kebutuhan dan nilai-nilai perusahaan. Hasil penelitian ini diharapkan dapat menjadi landasan untuk pengembangan sistem rekrutmen yang lebih baik dan memberikan dampak positif pada pertumbuhan perusahaan dalam industri teknologi informasi.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang penelitian tersebut, maka berikut ini adalah rumusan masalah penelitiannya yaitu sebagai berikut:

1. Bagaimana peran Sistem Pendukung Keputusan (SPK) dalam proses rekrutmen *system engineer* di PT Rackh Lintas Asia dapat meningkatkan efisiensi dan efektivitas seleksi kandidat?
2. Apakah implementasi metode ARAS dalam Sistem Pendukung Keputusan dapat membantu mengidentifikasi kriteria-kriteria kualifikasi yang paling krusial untuk menyeleksi calon s*ystem engineer* secara objektif?
3. Bagaimana penggunaan metode ARAS dalam Sistem Pendukung Keputusan dapat meningkatkan transparansi dan akuntabilitas dalam pengambilan keputusan rekrutmen di PT Rackh Lintas Asia?
4. Bagaimana dampak penggunaan Sistem Pendukung Keputusan dengan pendekatan metode ARAS terhadap kemampuan PT. Rackh Lintas Asia dalam menarik dan mempertahankan bakat terbaik dalam industri teknologi informasi?
   1. **Batasan Masalah**

Agar penelitian ini dapat lebih dipahami sejauh mana masalah dalam penelitian ini dibahas, maka berikut batasan masalah didalam penelitian ini, yaitu:

1. Penelitian difokuskan pada pengoptimalan prosedur sistem untuk meningkatkan efisiensi pemilihan s*ystem engineer* di PT Rackh Lintas Asia dengan menggunakan metode ARAS.
2. Penelitian mengaplikasikan metode ARAS dan mengintegrasikannya dengan sistem asesmen berbasis web untuk memberikan kerangka kerja yang lebih efisien dalam proses rekrutmen.
3. Fokus penelitian terbatas pada PT Rackh Lintas Asia, khususnya dalam seleksi s*ystem engineer*, memberikan analisis yang berlaku untuk konteks rekrutmen jabatan tersebut.
4. Dengan menggunakan contoh data dari PT Rackh Lintas Asia, penelitian ini berusaha memberikan gambaran yang representatif terhadap proses rekrutmen perusahaan.
   1. **Tujuan Penelitian**

Tujuan dari penelitian ini dilakukan adalah seperti yang tertera berikut ini ;

1. Untuk meningkatkan efisiensi & akurasi seleksi *system engineer* melalui metode ARAS.
2. Untuk menciptakan kerangka kerja efisien dengan metode ARAS dan sistem asesmen berbasis web.
3. Untuk memberikan analisis khusus pada seleksi *system engineer* di PT Rackh Lintas Asia.
4. Untuk memberikan gambaran representatif dengan menggunakan contoh data.
   1. **Manfaat Penelitian**

Berikut ini merupakan manfaat dari penelitian yang dilakukan ialah sebagai berikut:

1. Implementasi metode ARAS dan optimasi prosedur sistem di PT Rackh Lintas Asia diharapkan akan menghasilkan proses rekrutmen s*ystem engineer* yang lebih efisien, memperpendek waktu seleksi, dan meningkatkan akurasi pemilihan kandidat.
2. Penerapan metode ARAS dan integrasi dengan sistem asesmen berbasis web akan membantu menciptakan kerangka kerja yang lebih terstruktur dalam proses rekrutmen. Hal ini dapat memberikan keterbukaan dan transparansi yang lebih besar, memudahkan pemantauan dan evaluasi.
3. Dengan Dengan fokus pada seleksi s*ystem engineer*, penelitian ini diharapkan akan memberikan kontribusi signifikan terhadap peningkatan kualitas pemilihan karyawan di PT Rackh Lintas Asia. metode ARAS dapat membantu mengidentifikasi kandidat yang paling sesuai dengan kebutuhan perusahaan.
4. Penggunaan contoh data yang representatif dari PT Rackh Lintas Asia dapat memberikan kontribusi nyata terhadap pengembangan praktek rekrutmen berbasis data. Temuan dari penelitian ini dapat menjadi landasan untuk peningkatan proses rekrutmen di perusahaan sejenis dalam industri teknologi informasi.

**BAB II**

**TINJAUAN PUSTAKA**

**2.1 PT. Rackh Lintas Asia**

PT. Rackh Lintas Asia adalah salah satu penyedia layanan *Internet Service Provider* (ISP) yang telah membangun reputasi yang kuat dalam industri telekomunikasi. Dengan kantor pusatnya yang berlokasi di Medan, perusahaan ini telah meluaskan jangkauannya dengan membuka cabang di Jakarta, memperkuat kehadirannya di pusat bisnis dan teknologi Indonesia. Sebagai penyedia layanan internet terkemuka, PT Rackh Lintas Asia tidak hanya menawarkan konektivitas internet berkualitas tinggi tetapi juga berkomitmen untuk memberikan solusi teknologi informasi yang inovatif kepada pelanggan di seluruh wilayah.

Cabang PT. Rackh Lintas Asia di Jakarta menjadi pusat strategis untuk mendukung kebutuhan pelanggan di ibu kota. Dengan infrastruktur yang canggih dan tim teknis yang terampil, cabang ini berperan penting dalam memberikan layanan yang handal dan terjangkau kepada pelanggan korporat dan individu di Jakarta. Selain itu, keberadaan PT Rackh Lintas Asia di Jakarta juga mencerminkan komitmennya untuk mendukung pertumbuhan sektor teknologi informasi dan komunikasi di wilayah ini.

Melalui rekrutmen cabang Jakarta, PT. Rackh Lintas Asia menunjukkan komitmen terhadap pengembangan sumber daya manusia lokal, meningkatkan lapangan kerja, dan mendukung pertumbuhan ekonomi di Jakarta. Dengan membangun tim yang beragam dan terampil di tingkat lokal, perusahaan dapat

meraih peluang yang lebih luas dan memberikan kontribusi positif terhadap perkembangan sektor teknologi informasi dan komunikasi di ibu kota. Oleh karena itu, rekrutmen untuk cabang Jakarta menjadi langkah strategis untuk memperkuat keberlanjutan dan dominasi pasar PT Rackh Lintas Asia dalam industri ISP di Indonesia.

**2.2 Sistem Pendukung Keputusan (SPK)**

Sistem Pendukung Keputusan (SPK) adalah platform berbasis komputer yang membantu menyelesaikan masalah dengan memanajemen mengatasi berbagai masalah terstruktur dan tidak terstruktur dengan menggunakan data dan model yang disiapkan[1]. Menjadi alat yang penting dalam konteks pengambilan keputusan di era informasi digital, membantu organisasi dalam mengidentifikasi masalah, mengumpulkan data yang diperlukan, dan menganalisis situasi dengan cermat. Dari analisis ini, SPK dapat memberikan berbagai pilihan keputusan yang relevan yang memungkinan para pengambil keputusan untuk membuat keputusan yang lebih terinformasi dan efektif[2].

**2.2.1 Konsep Sistem Pendukung Keputusan**

Konsep Sistem Pendukung Keputusan (SPK) pertama kali diungkapkan pada tahun 1971 oleh *Michael Scoot Morton* dengan istilah *Management Decision System.* Setelah itu sejumlah perusahaan, lembaga penelitian dan perguruan tinggi mulai melakukan penelitian dan membangun sistem pendukung keputusan, sehingga dari produksi yang dihasilkan dapat disimpulkan bahwa sistem ini merupakan suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur[3].

*Decision Support System* (DSS) merupakan sistem yang memberikan fasilitas yang menyediakan informasi, permodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi struktur dan situasi yang tidak terstruktur, dimana tidak ada seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkorelasi satu dengan yang lainnya untuk mencapai tujuan tertentu. Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan *(input)* sehingga menghasilkan keluaran *(output)*[4].

Secara Sederhana Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, terpadu[5]. Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tertentu. Tindakan memilih strategi atau aksi yang diyakini *supervisor* akan memberikan solusi terbaik atas sesuatu disebut pengambilan keputusan.

Suatu keputusan yang diambil untuk menyelesaikan suatu masalah dilihat dari rekonstruksi pendukung yang bisa dibagi menjadi bermacam macam klasifikasi dalam sistem pendukung keputusan guna untuk mempermudah penerapan ilmu sistem pendukung keputusan dalam berbagai aspek permasalahan. Jenis-jenis keputusan juga bisa membantu dalam menganalisis sebuah permasalahan yang akan di selesaikan dengan sistem, berikut adalah jenis-jenis keputusan:

1. Keputusan terstruktur (*structure decision*)

Keputusan terstruktur adalah keputusan yang dilakukan secara berulang-ulang dan bersikap rutin. Misalnya, keputusan pemesanan barang dan keputusan penagihan piutang.

1. Keputusan semi-terstruktur (*semistructured decision*)

Keputusan semi-terstruktur adalah keputusan yang memiliki dua sifat. Sebagian keputusan bisa atasi oleh komputer namun tetap harus dilakukan oleh pengambil keputusan. Contoh keputusan jenis ini adalah pengevaluasian kredit, penjadwalan produksi, dan pengendalian sediaan.

1. Keputusan tidak terstruktur (*unstructured decision*)

Keputusan tidak terstruktur adalah keputusan yang penanganannya rumit karena tidak terjadi berulang-ulang atau tidak selalu terjadi[1].

**2.2.2 Komponen-Komponen Sistem Pendukung Keputusan**

Komponen-komponen Sistem Pendukung Keputusan terdiri dari:

1. *Data* *Management*

Termasuk *database,* yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS)[6].

1. *Model* *Management*

Melibatkan model finansial, statistikal, *management science,* atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

1. *Communication* (dialog subsistem)

Melalui subsistem ini, pengguna dapat berkomunikasi dan memberikan perintah pada DSS. Ini berarti menyediakan antarmuka.

1. *Knowledge Management*

Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri[6].

**2.2.3 Tujuan Sistem Pendukung keputusan**

Tujuan Sistem Pendukung Keputusan (SPK) yaitu:

1. Membantu *supervisor* membuat keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung penilaian *supervisor* bukan mencoba untuk menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan *supervisor* daripada efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Konstruksi suatu tim pengambil keputusan, terutama yang terdiri dari para ahli, dapat menimbulkan biaya yang signifikan. Solusi terkomputerisasi dapat meminimalkan ukuran tim dan memfasilitasi partisipasi anggota tim dari lokasi yang berbeda (mengurangi biaya perjalanan). Lebih lanjut, produktivitas staf pendukung, seperti analis keuangan dan hukum, dapat ditingkatkan melalui penggunaan peralatan optimisasi yang merancang strategi terbaik untuk mengelola bisnis.
6. Dukungan kualitas. Komputer Dapat meningkatkan mutu keputusan yang dihasilkan. Sebagai contoh, semakin besar akses terhadap data, semakin banyak alternatif yang dapat dievaluasi.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.[7]

**2.2.4 ProsesPengambilan Keputusan Dalam Sistem Pendukung Keputusan**

Ada tiga fase dalam proses pengambilan keputusan.

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian ruang lingkup problematika secara proses pengenalan masalah masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasikan masalah.

1. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis *alternative* tindakan yang bisa dilakukan. Tahap ini meliputi kelayakan solusi.

1. *Choice*

Pada fase ini, dilakukan seleksi antara berbagai alternatif tindakan yang potensial untuk dijalankan. Hasil dari proses seleksi tersebut selanjutnya diaplikasikan dalam tahap pengambilan keputusan.

**2.2.5 Elemen Sistem Pendukung Keputusan**

Elemen sistem pendukung keputusan adalah suatu pembagian ataupun entitas yang ada pada sistem pendukung keputusan itu sendiri. Secara konsep ada 3 (tiga) elemen yang terkait dengan sistem pendukung keputusan, yaitu:

1. Masalah.

Dalam sebuah sistem pendukung keputusan terdapat beberapa jenis masalah yaitu: Masalah terstruktur, masalah semi-terstruktur dan masalah tidak terstruktur.

1. Solusi.

Dalam sebuah sistem pendukung keputusan terdapat beberapa jenis solusi pemecahan masalah diantaranya yaitu: *Multi* *Attribute Decision Making* (MADM) seperti: metode *Simple Additive Weighting* (SAW), metode *Weight Product* (WP), metode *Analythical Hierarchy Process* (AHP), metode *Topsis* dan lain-lain. Kemudian metode *Multi Criteria Decision Making* (MCDM) seperti: metode *Promethee*, metode *Electre*, metode *Oreste*, metode *Entropi* dan lain-lain. Selain terdapat juga metode *Multi Factor Evaluation Process* (MFEP), metode *Multi Attribute Utility Theory* (MAUT), serta metode FMADM (*Fuzzy Multi Attribute Decision Making*) yang terdiri dari F-AHP, F-SAW, dan lain-lain.

1. Hasil.

Keluaran dari suatu sistem pendukung keputusan adalah keputusan yang menjadi pedoman untuk merumuskan kebijakan terkait dengan masalah yang sedang diselidiki atau dibahas. Keputusan merupakan hasil dari pemilihan strategi atau tindakan untuk menanggapi suatu masalah. Proses pemilihan strategi atau tindakan yang dianggap supervisor sebagai solusi terbaik disebut sebagai aktivitas pengambilan keputusan..[8]

**2.3 Metode Yang Digunakan Dalam Pembuatan Sistem**

Pada pengembangan sistem rekrutmen di PT Rackh Lintas Asia, diperlukan pendalaman algoritma sistem guna mendukung pengambilan keputusan terkait input yang dimasukkan ke dalam sistem. Metode yang diterapkan

**2.3.1 Metode ARAS**

Additive Ratio Assessment (ARAS) adalah sebuah metode yang digunakan untuk perangkingan kriteria, secara konsep metode ARAS ini digunakan dengan metode lain yang menggunakan konsep perangkingan seperti SAW atau TOPSIS, dimana proses penentuan ranking harus di olah kembali dengan menggunakan Metode ARAS sehingga hasil peringkat dengan metode SAW dan metode SAW+ARAS bisa berberda hasilnya.

Perhitungan Metode Additive Ratio Assessment (ARAS) dapat diartikan sebagai satu-satunya metode sistem pendukung keputusan yang dalam perangkinganya menggunakan konsep perangkingan Utility Degree, yaitu dengan membandingkan nilai indeks keseluruhan alternatif optimal terhadap nilai indeks keseluruhan setiap alternatif [9].

1. Menentukan kriteria, bobot, alternatif dan menentukan nilai alternatif dari setiap kriteria serta menentukan nilai optimal *benefit* dan *cost*.
2. Merubah nilai kriteria menjadi matriks keputusan.

( i=0, m; … j = 1,n ) …………………… [2.1]

Dimana:

m = Jumlah alternatif

n = Jumlah kriteria

= Nilai kriteria dari alternatif i

= Nilai optimal dari kriteria j

1. Menentukan nilai optimal kriteria j (𝑋0𝑗) yaitu dengan cara:

= . If Lebih Baik  
 …...……………………………….[2.2]

= = If Lebih Baik

1. Menentukan normalisasi matriks keputusan dari semua kriteria mempunyai dua cara yaitu:

Perhitungan dengan kategori Benefit Dimana adalah nilai normalisasi

= ……………………………………………………………..[2.3]

Perhitungan dengan kategori Cost mempunyai dua cara yaitu:  
Tahap 1 : Xij = ………………………………………………………..[2.4]

Tahap 2 : R = ……………………………………………………..[2.5]

1. Menentukan pembobotan pada matriks yang sudah dinormalisasi:

D = [] m x n = …...………………..………………………………..[2.5]  
Dimana :  
 = bobot kriteria *j*

1. Menentukan Nilai dari fungsi optimalisasi (*Si*)

Si = = 1 (*i* = 1,2,.,*m:j* = 1,2,…,*n*) ………………………………...[2.6]

Dimana *Si* merupakan fungsi optimalisasi alternatif i. Nilai terbesar adalah nilai yang terbaik dan nilai terkecil merupakan nilai yang terburuk. Nilai dan bobot kriteria yang berhubungan akan berpengaruh pada hasil akhir.

1. Langkah terakhir adalah menentukan nilai Derajat Utilitas (Peringkat) dengan menggunakan rumus:  
    = ; …………………………………………………………………...[2.7]  
   Dimana hasil dari *Si* dan *S0* merupakan nilai kriteria optimalitas (perangkingan)

Keterangan:

= nilai tingkat peringkat alternatif

= nilai optimum untuk alternatif i

= nilai optimum untuk alternatif optimal

**2.4 UML *(Unified Modeling Languange)***

Dalam perkembangan teknologi perangkat lunak, diperlukan bahasa pemodelan yang dapat digunakan untuk merancang perangkat lunak. Standarisasi bahasa tersebut juga penting agar pemodelan dapat dipahami oleh berbagai pihak di seluruh dunia. Mengingat kompleksitas kolaborasi di antara individu dengan latar belakang yang beragam, diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat diakses dan dimengerti oleh banyak orang.

*Unified Modelling Language* (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Abstraksi konsep dasar UML terdiri dari *structural classification, dynamic behavior*, dan model *management* dapat kita pahami *main concepts* sebagai *term* yang akan muncul pada saat membuat *diagram* dan *view* adalah kategori dari diagram tersebut.[10]

**2.4.1 *Use Case Diagram***

*Use Case* atau *diagram* *use case* merupakan pemodelan yang digunakan untuk menggambarkan sebuah kasus interaksi antara aktor dan sistem meliputi apa yang dapat dilakukan seorang pengguna terhadap sistem yang dijalankan.

Suatu *use case* diagram akan ditujukan untuk menyatakan visualisasi interaksi yang terjadi diantara pengguna (aktor) dengan sistem. Ada 2 elemen penting yang harus digambarkan, yaitu aktor dan *use case*. Berikut ini adalah simbol simbol *use case* diagram:

Tabel 2.1 Simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **Simbol** | **Keterangan** |
| 1 | *Use case* |  | Fungsional yang disediakan dari sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*. |
| 2 | Aktor / *actor* |  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang. |
| 3 | *System Boundary* |  | Digambarkan dengan kotak di sekitar *use case* dan digunakan saat memberikan pilihan sistem alternatif. |
| 4 | Asosiasi/ *association* |  | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* memiliki interaksi dengan aktor. |

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **Simbol** | **Keterangan** |
| 5 | Extensi / *extend* |  | Relasi *use case* tambahan ke semua *use case* yang ada dan berdiri sendiri |
| 6 | Generalisasi /*generalization* |  | Generalisasi dan spesialisasi adalah hubungan antara dua use case di mana satu lebih umum daripada yang lain |
| 7 | Menggunakan /*include / uses* |  | Ralasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya. |

**2.4.2 *Activity Diagram***

Diagram ini menggambarkan berbagai aktifitas dalam sistem yang sedang dirancang, mulai dari titik awal, melalui kondisi *(decision)* yang mungkin terjadi, kemudian sampai pada titik akhir. *Diagram* ini tidak menggambarkan perilaku/proses *internal* sebuah sistem maupun interaksi antar sub-sistem, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas secara umum atau global.[11]

Tabel 2.2 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **Gambar** | **Keterangan** |
| 1 | Status Awal |  | Status awal aktifitas sistem, sebuah diagram memiliki sebuah status awal. |

Tabel 2.2 Simbol *Activity Diagram* (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama** | **Gambar** | **Keterangan** |
| 2 | Aktifitas |  | Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja. |
| 3 | Percabangan/ *Decision* |  | Asosiasi percabangan jika ada pilihan aktifitas lebih dari satu. |
| 4 | Percabangan/ *fork* |  | Asosiasi percabangan lebih dari satu aktifitas dipisahkan. |
| 5 | Penggabungan /*join* |  | Asosiasi penggabungan lebih dari satu aktifitas digabungkan. |
| 6 | Status akhir |  | Status akhir yang dilakukan sistem sebuah diagram aktifitas memiliki sebuah status akhir. |
| 7 | *Swimlane* | Nama *Swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terdapat aktifitas yang terjadi. |

**2.4.3 *Class Diagram***

Diagram kelas atau *class diagram* adalah diagram yang menggambarkan struktur yang berjalan pada sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi[12].

Tabel 2.3 Simbol *Class Diagram*

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Deskripsi** |
| 1 | |  | | --- | | Nama\_kelas | | + Atribut | | +Operasi () | | Kelas pada struktur sistem |
| 2 | Antarmuka/*interface* | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| 3 | Asosiasi/*association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| 4 | Asosiasi berarah/*directed assosiation* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain. |
| 5 | Agregasi/*aggregation* | Relasi antar kelas dengan makna semua bagian (whole-part) |
| 6 | Generalisasi | Relasi antar kelas dengan maknna generalisasi-spesialisasi (umum khusus) |
| 7 | Kebergantungan/*dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas |

**2.5 Aplikasi Pendukung**

Aplikasi Pendukung merujuk pada perangkat lunak atau sistem yang diciptakan untuk memberikan dukungan atau bantuan dalam berbagai kegiatan atau tugas. Aplikasi ini dirancang dengan tujuan spesifik untuk meningkatkan efisiensi, produktivitas atau kinerja dalam suatu konteks tertentu. Dalam SPK, aplikasi pendukung ini secara khusus dirancang untuk membantu proses pengambilan keputusan atau manajemen informasi. Adapun aplikasi *tool* pendukung dalam menjalankan suatu sistem antara lain:

**2.5.1 Visual Studio Code**

*Visual Studio Code* merupakan aplikasi penyuntingan kode-sumber buatan *Microsoft* untuk *Linux*, *macOS*, dan *Windows*. *VSCode* menyediakan fitur seperti penyorotan sintaksis, penyelesaian kode, kutipan kode, merefaktor kode, dan *Git*



Gambar 2.1 *Visual Studio Code*

*Visual Studio Code* pertama kali pada tanggal 29 April 2015 oleh *Microsoft* di konferensi Build 2015. *VSCode* dibuat dengan menggunakan Electron, sebuah *framework* yang memungkinkan aplikasi web berjalan di luar browser. *VSCode* adalah perangkat lunak gratis dan sumber terbuka yang dirilis dengan lisensi MIT. *VSCode* tersedia di *GitHub* dan dapat diunduh secara gratis. *Visual Studio Code* mendukung berbagai bahasa pemrograman, termasuk *JavaScript, TypeScript, Python, Java, C/C++*, dan lainnya. *Visual Studio Code* juga mendukung berbagai ekstensi yang dapat ditambahkan untuk menambah fitur baru atau menyesuaikan tampilan dan nuansa *Visual Studio Code*.

* + 1. ***XAMPP (X-Apache, Mysql, PHP, dan Perl)***

sebuah paket perangkat lunak yang terdiri dari beberapa program yang digunakan untuk membangun dan menjalankan situs web dan aplikasi web. *XAMPP* adalah singkatan dari *Apache*, *MySQL*, *PHP*, dan *Perl*. *XAMPP* adalah *software* *open source* berbasis *web* *server* yang berisi berbagai program. Aplikasi ini mendukung berbagai sistem operasi seperti *Linux, Windows, MacOS,* dan *Solaris*. Fungsi *XAMPP* adalah sebagai *server* lokal/ *localhost*, di dalamnya sudah mencakup *program* *Apache*, *MySQL* dan *PHP*[13].



Gambar 2.2 *XAMPP*

*XAMPP* adalah pilihan yang bagus untuk pengembang web yang ingin membangun situs web dan aplikasi web lokal. *XAMPP* memungkinkan pengembang untuk menguji situs web dan aplikasi web mereka tanpa harus menghubungkannya ke internet. Dalam *XAMPP*, terdapat beberapa komponen penting yang perlu kamu ketahui. Berikut beberapa komponen utama *XAMPP* :

* + 1. *XAMPP control panel* adalah komponen yang digunakan untuk mengelola komponen lainnya dalam *XAMPP*. Dengan menggunakan *control panel*, kamu bisa mengaktifkan fungsi *apache*, *mySQL*, *filezilla*, *config*, *netstat* dan konfigurasi *XAMPP* lainnya
    2. *Htdocs* adalah Komponen *XAMPP* tersedia dalam bentuk folder, dengan folder *'htdocs'* berperan sebagai lokasi penyimpanan untuk folder dan *file* yang dapat diakses melalui peramban (*browser*). Pada penggunaan *hosting*, *'htdocs'* berfungsi sebagai folder publik. Kapasitas *'htdocs'* sejalan dengan kapasitas partisi yang digunakan. Secara umum, *'htdocs'* biasanya terletak di alamat *path* *C:\xampp\htdocs*.
    3. *Config* merupakan komponen pada *XAMPP* yang berfungsi untuk mengatur pengaturan dasar. Seperti mengatur aplikasi editor teks dan browser yang akan digunakan secara default oleh aplikasi *XAMPP*.
    4. *Netstat* adalah Komponen dalam *XAMPP* bertugas memeriksa ketersediaan port yang digunakan oleh *XAMPP*, memastikan tidak ada konflik dengan aplikasi lain. Jika port standar *XAMPP* sudah terpakai oleh aplikasi lain, hal ini dapat menghambat fungsi optimal aplikasi *XAMPP*. Dalam situasi ini, disarankan untuk mengganti port yang terkait dengan port lain yang masih tersedia.
    5. ***Draw.io***

Aplikasi selanjutnya, untuk membangun rancangan flow aplikasi kedalam bentuk diagram, menggunakan *Draw.io*

Gambar 2.3 *Dashwork Draw.io*

*Draw.io* adalah aplikasi *Draw.io* yang memungkinkan pengguna untuk membuat berbagai jenis *Draw.io*, seperti diagram alir, diagram organisasi, diagram *UML*, dan banyak lagi. Aplikasi ini dapat diakses secara gratis melalui browser web dan tidak memerlukan instalasi atau pendaftaran. Pengguna dapat menyimpan dan membasgikan diagram yang dibuat dalam berbagai format file, seperti *PNG*, *PDF*, *SVG*, atau *XML*.

### **2.5.4 *MySQL***

Adi Nugroho (2011;5), sebagaimana dikutip dalam jurnal Fery Wongso (ISSN: 1829-9822), menyatakan bahwa basis data merupakan koleksi data yang teroganisir dengan cara yang memungkinkan penyimpanan, manipulasi, dan pengambilan data oleh pengguna menjadi lebih mudah. Dalam perkembangannya, *MySQL* sering disebut sebagai *SQL* yang merupakan singkatan dari *Structured Query Language*. *SQL* adalah jenis Bahasa terstruktur yang dirancang khusus untuk mengolah basis data[17]. *MySQL* adalah sistem manajemen basis data yang memiliki sifat relasional. Ini berarti bahwa data yang dikelola dalam basis data ditempatkan di beberapa tabel terpisah, yang menghasilkan proses manipulasi data yang lebih efisien. Berikut merupakan tampilan *MySQL*:



Gambar 2.4 *Database MySQL*

*SQL* memungkinkan pengguna untuk mengetahui lokasi atau susunan informasi dengan lebih mudah. Meskipun lebih sederhana daripada bahasa pemrograman, *SQL* memiliki tingkat kompleksitas yang lebih tingi jika dibandingkan dengan perangkat lunak lembar kerja dan pengolah data. *SQL* adalah bahasa pemrograman yang diciptakan khusus untuk mengirimkan perintah *query* ke *database*, yaitu untuk mengakses data berdasarkan alamat tertentu.

* + 1. ***Github***

Dalam proses pengembangan sistem maupun pembuatan sistem, sangat diperlukan *tracking version,* pada aplikasi yang sedang dibangun, dengan tujuan untuk *monitoring updating system* apa saja yang sudah dilakukan pada sistem, serta memudahkan *developer* untuk mengembangkan atau membangun sebuah sistem. Adapun aplikasi yang dibutuhkan adalah *Github.*

****

Gambar 2.5 *Dashboard* *GitHub*

*Github* sendiri, diambil dari 2 kata yang berbeda yaitu *Git* dan *Hub*, *Git* dikembangkan oleh Linus Torvalds pada tahun 2005, dan ini merupakan inti atau jantung GitHub. berfungsi untuk membantu developer melakukan *version control development* terhadap suatu aplikasi/*software*, yang berarti code base dan riwayat kode akan tersedia di komputer setiap *developer* bertujuan untuk memudahkan pembuatan *branch* dan penggabungan. Sedangkan *Hub* adalah jiwa *GitHub*. Sistem *Hub* yang ada pada *GitHub* berfungsi untuk mengubah baris perintah (*command line*), seperti Git, menjadi jaringan media sosial terbesar bagi para *developer*, dan memungkinkan usernya untuk berkomunikasi dengan orang-orang yang memiliki kesamaan visi dan misi.

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Metode Penelitian**

Metode penelitian merupakan langkah-langkah yang dilakukan untuk mengumpulkan data atau informasi agar tercapainya sebuah tujuan penelitian, khususnya terkait dalam menentukan kandidat seorang *system engineer* Didalam melakukan penelitian ini, terdapat beberapa langkah yang harus dilakukan dalam mengolah data menjadi sebuah keputusan.

**3.1.1 Pengumpulan Data**

Pengumpulan data atau yang dikenal dengan istilah *Data Collecting* merupakan teknik atau prosedur yang digunakan dalam pengumpulan data dari penelitian, adapun langkah prosedur pengumpulan data pada penelitian ini yaitu:

1. Observasi

Kegiatan observasi didalam penelitian ini dilakukan dengan tinjauan langsung ke kantor PT Rackh Lintas Asia di Jl. Senam No.2, Pasar Merah Barat, Kec. Medan Kota, Kota Medan, Sumatera Utara.

1. Wawancara

Kegiatan wawancara dilakukan kepada pihak yang terlibat pada kantor PT Rackh Lintas Asia. Dalam hal ini, kegiatan wawancara dilakukan dengan Pak Dedy selaku HRGA Manager PT Rackh Lintas Asia. Dalam kegiatan ini wawancara dilakukan untuk menggali informasi terkait kriteria data alternatif penilaian terkait rekrutmen *system engineer*, Berikut adalah data alternatif penilaian dan kriteria yang telah didapatkan dari perusahaan:

Tabel 3.1 Data Alternatif

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama** | **Network Management** | **Server Management** | **Cloud Computing** | **Kerjasama**  **Tim** | **Trouble**  **shoot** |
| 1 | Hafiz Sitepu | 78 | 90 | 65 | 87 | 73 |
| 2 | Ahmad Rahmad | 85 | 78 | 92 | 80 | 95 |
| 3 | Heru Pranata | 70 | 88 | 75 | 85 | 72 |
| 4 | Agung  Alponi | 82 | 75 | 80 | 78 | 85 |
| 5 | Dika  Radit | 95 | 88 | 92 | 90 | 94 |
| 6 | Dwi Susanto | 77 | 83 | 78 | 80 | 79 |
| 7 | Dandi Ilyas | 58 | 75 | 50 | 52 | 55 |
| 8 | Andreas | 90 | 85 | 92 | 88 | 95 |
| 9 | Derry Akbar | 72 | 88 | 58 | 75 | 50 |
| 10 | Rico Zahiri | 88 | 92 | 85 | 89 | 90 |

**3.1.2 Studi Pustaka**

Di dalam studi literatur, penelitian ini banyak menggunakan jurnal-jurnal baik jurnal nasional sebagai sumber referensi. Diharapkan literatur tersebut dapat membantu peneliti dalam menyelesaikan permasalahan yang terjadi di PT Rackh Lintas Asia terkait menentukan seorang s*ystem engineer*. Semua literatur yang telah dikumpulkan kemudian akan dihubungkan menggunakan aplikasi *Mendeley* *Dekstop*.

* 1. **Penerapan Metode Additive Ratio Assesment (ARAS)**

Penerapan metode ARAS merupakan penjelasan langkah-langkah penyelesaian masalah dalam menyelesaikan kebutuhan yang terjadi terkait menentukan *system engineer* di PT Rackh Lintas Asia bentuk kerangka kerja. Berikut ini adalah kerangka kerja dari algoritma metode ARAS:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Menentukan Data Alternatif, Data Kriteria Serta Bobot Penilaian |  |  | | --- | | Membentuk Decision Making Matriks |  |  | | --- | | Melakukan Normalisasi Matriks Pada Setiap Kriteria  Benefit  *Xij* =  Cost  *Xij* = ; *Xij* = *Xij* |  |  | | --- | | Menentukan Bobot Matriks Normalisasi |  |  | | --- | | Menentukan Nilai Fungsi Optimalisasi (Si)  Si = |  |  | | --- | | Mencari Nilai (KD Dengan Cara Membagi Setiap Nilai Alternatif Dengan Alternatif  = , |  |  | | --- | | Melakukan Perangkingan Dari Hasil Perhitungan | |

Gambar 3.1 Kerangka Kerja metode ARAS

1. Menentukan Data Alternatif, Data Kriteria Serta Bobot Penilaian

Menentukan Data Alternatif, Data Kriteria Serta Bobot Penilaian Pengambilan keputusan ini berdasarkan pada kriteria yang sudah menjadi penentu dalam menyelesaikan kebutuhan yang terjadi di PT Rackh Lintas Asia terkait menentukan *system engineer*. Adapun deskripsi data kriteria yang diambil dari tahap wawancara pada PT Rackh Lintas Asia yaitu sebagai berikut:

Tabel 3.2 Data Kriteria Penelitian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Kode** | **Nama Kriteria** | **Bobot** | **Jenis Kriteria** |
| 1 | C1 | Kemampuan Network Management | 25% | *Benefit* |
| 2 | C2 | Kemampuan Server Management | 25% | *Benefit* |
| 3 | C3 | Kemampuan Cloud Computing | 25% | *Benefit* |
| 4 | C4 | Kerjasama Tim | 10% | *Benefit* |
| 5 | C5 | Kemampuan Troubleshoot | 15% | *Benefit* |

Data yang telah didapat perlu dilakukan konversi pada setiap kriteria untuk dilakukan pengolahan kedalam metode ARAS. Berikut adalah tabel konversi dari kriteria yang digunakan:

1. Kemampuan *Network Management*

Kriteria kemampuan *network management* berkaitan dengan kemampuan seorang *system engineer* yang perlu memiliki kemampuan mendalam dalam manajemen jaringan, mencakup pemahaman teknis yang kuat, keahlian dalam menanggapi perubahan teknologi, serta kemampuan analitis untuk mengidentifikasi dan menyelesaikan masalah jaringan.

Tabel 3.3 Kriteria Kemampuan *Networking*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kemampuan Networking** | **Keterangan** | **Nilai** |
| 1 | 90 - 100 | Sangat Baik | 4 |
| 2 | 80 - 89 | Baik | 3 |
| 3 | 60 - 79 | Cukup | 2 |
| 4 | <60 | Kurang Baik | 1 |

1. Kemampuan *Server Management*

Kriteria kemampuan *server management* perlu memiliki keahlian mendalam dalam mengelola *server*, termasuk pemahaman yang kuat terhadap arsitektur *server*, administrasi sistem, dan keamanan informasi. Kemampuan untuk merancang, mengimplementasikan, dan memelihara infrastruktur *server* dengan efisien merupakan aspek penting

Tabel 3.4 Kriteria Kemampuan *Server Management*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kemampuan Server Management** | **Keterangan** | **Nilai** |
| 1 | 90 - 100 | Sangat Baik | 4 |
| 2 | 80 - 89 | Baik | 3 |
| 3 | 60 - 79 | Cukup | 2 |
| 4 | <60 | Kurang Baik | 1 |

1. Kemampuan *Cloud Computing*

Kriteria kemampuan c*loud computing* perlu memiliki pemahaman yang mendalam tentang teknologi *cloud*, termasuk keahlian dalam merancang, mengimplementasikan, dan mengelola solusi *cloud*. Kreativitas dalam mengadaptasi teknologi *cloud* untuk meningkatkan efisiensi dan skalabilitas, pemahaman yang kuat terhadap keamanan *cloud*, serta kemampuan analitis untuk mengevaluasi dan memilih layanan *cloud* yang tepat menjadi kriteria penting. Selain itu, responsif terhadap perkembangan tren *cloud computing*.

Tabel 3.5 Kriteria *Cloud Computing*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kemampuan Cloud Computing** | **Keterangan** | **Nilai** |
| 1 | 90 - 100 | Sangat Baik | 4 |
| 2 | 80 - 89 | Baik | 3 |
| 3 | 60 - 79 | Cukup | 2 |
| 4 | <60 | Kurang Baik | 1 |

1. Kerjasama Tim

Kriteria kerjasama tim mencakup kemampuan berkolaborasi, berbagi ide, dan mendukung tujuan bersama. Kriteria penting termasuk responsif terhadap masukan rekan tim, kemampuan membangun hubungan positif, dan keterampilan komunikasi interpersonal yang baik.

Tabel 3.6 Kriteria kerja Kerjasama Tim

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Kemampuan Kerjasama Tim** | **Keterangan** | **Nilai** |
| 1 | 90 - 100 | Sangat Baik | 4 |
| 2 | 80 - 89 | Baik | 3 |
| 3 | 60 - 79 | Cukup | 2 |
| 4 | <60 | Kurang Baik | 1 |

1. Kemampuan *Troubleshoot*

Kriteria kemampuan *troubleshoot* perlu memiliki kemampuan yang handal dalam mengidentifikasi, menganalisis, dan menyelesaikan masalah teknis. Kemampuan untuk melakukan troubleshoot dengan cepat dan efektif pada perangkat keras, perangkat lunak, atau jaringan merupakan aspek utama.

Tabel 3.7 Kriteria *Troubleshoot*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Troubleshoot** | **Keterangan** | **Nilai** |
| 1 | 90 - 100 | Sangat Baik | 4 |
| 2 | 80 - 89 | Baik | 3 |
| 3 | 60 - 79 | Cukup | 2 |
| 4 | <60 | Kurang Baik | 1 |

Berikut ini merupakan data alternatif pada setiap kriteria yang nantinya akan dilakukan proses penilaian menggunakan metode ARAS.

Tabel 3.8 Data Alternatif

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Nama Alternatif** | **C1** | **C2** | **C3** | **C4** | **C5** |
| A01 | Hafiz Sitepu | Cukup | Sangat Baik | Cukup | Baik | Cukup |
| A02 | Ahmad Rahmad | Baik | Cukup | Sangat Baik | Baik | Sangat Baik |
| A03 | Heru Pranata | Cukup | Baik | Cukup | Baik | Cukup |
| A04 | Agung Alponi | Baik | Cukup | Baik | Cukup | Baik |
| A05 | Dika Radit | Sangat Baik | Baik | Sangat Baik | Sangat Baik | Sangat Baik |

Tabel 3.8 Data Alternatif (Lanjutan)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Nama Alternatif** | **C1** | **C2** | **C3** | **C4** | **C5** |
| A06 | Dwi Susanto | Cukup | Baik | Cukup | Baik | Cukup |
| A07 | Dandi Ilyas | Kurang Baik | Cukup | Kurang Baik | Kurang Baik | Kurang Baik |
| A08 | Andreas | Sangat Baik | Baik | Sangat Baik | Baik | Sangat Baik |
| A09 | Derry Akbar | Cukup | Baik | Kurang Baik | Baik | Kurang Baik |
| A10 | Rico Zahiri | Baik | Sangat Baik | Baik | Baik | Sangat Baik |

Langkah selanjutnya adalah melakukan konversi dari data alternatif sesuai dengan nilai bobot penilaian dari masing-masing kriteria yang telah ditentukan, berikut ini adalah hasil dari konversi nilai alternatif:

Tabel 3.9 Hasil Konversi Data Alternatif

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kode** | **Nama Alternatif** | **C1** | **C2** | **C3** | **C4** | **C5** |
| A00 | **-** | **Max** | **Max** | **Max** | **Max** | **Max** |
| 4 | 4 | 4 | 4 | 4 |
| A01 | Hafiz Sitepu | 2 | 4 | 2 | 3 | 2 |
| A02 | Ahmad Rahmad | 3 | 2 | 4 | 3 | 4 |
| A03 | Heru Pranata | 2 | 3 | 2 | 3 | 2 |
| A04 | Agung Alponi | 3 | 2 | 3 | 2 | 3 |
| A05 | Dika Radit | 4 | 3 | 4 | 4 | 4 |
| A06 | Dwi Susanto | 2 | 3 | 2 | 3 | 2 |
| A07 | Dandi Ilyas | 1 | 2 | 1 | 1 | 1 |
| A08 | Andreas | 4 | 3 | 4 | 3 | 4 |
| A09 | Derry Akbar | 2 | 3 | 1 | 3 | 1 |
| A10 | Rico Zahiri | 3 | 4 | 3 | 3 | 4 |

1. Membentuk Matriks Keputusan (*Decision Making Matriks*)

Berdasarkan data pada tabel diatas, berikut ini adalah perhitungan metode ARAS untuk mendapatkan hasil perangkingan terkait menentukan seorang s*ystem engineer*. Lakukan pembentukan Matriks Keputusan (*Decision Making Matriks*) seperti di bawah ini:

1. Melakukan Normalisasi Matriks

Benefit: *Rij* =

*Cost:*

Tahap 1 : *Rij* =

Tahap 2 : *Rij* =

Normalisasi matriks keputusan pada kriteria C1

= = 0.1333

= = 0.0667

= = 0.1000

= = 0.0667

= = 0.1000

= = 0.1333

= = 0.0667

= = 0.0333

= = 0.1333

= = 0.0667

= = 0.1000

Normalisasi matriks keputusan pada kriteria C2

= = 0.1212

= = 0.1212

= = 0.0606

= = 0.0909

= = 0.0606

= = 0.0909

= = 0.0909

= = 0.0606

= = 0.0909

= = 0.0909

= = 0.1212

Normalisasi matriks keputusan pada kriteria C3

= = 0.1333

= = 0.0667

= = 0.1333

= = 0.0667

= = 0.1000

= = 0.1333

= = 0.0667

= = 0.0333

= = 0.1333

= = 0.3333

= = 0.1000

Normalisasi matriks keputusan pada kriteria C4

= = 0.1250

= = 0.0937

= = 0.0937

= = 0.0937

= = 0.0625

= = 0.1250

= = 0.0937

= = 0.0312

= = 0.0937

= = 0.0937

= = 0.0937

Normalisasi matriks keputusan pada kriteria C5

= = 0.1290

= = 0.0645

= = 0.1290

= = 0.0645

= = 0.0968

= = 0.1290

= = 0.0645

= = 0.0322

= = 0.1290

= = 0.0323

= = 0.1290

Berdasarkan dari perhitungan diatas maka dapat diperoleh sebuah matrik keputusan yang telah dinormalisasikan sebagai berikut:

=

1. Menentukan Bobot Matriks Ternormalisasi

Selanjutnya adalah langkah melakukan perkalian nilai matriks dengan nilai bobot kriteria dengan rumus sebagai berikut:

Rumus:

D = [dij] m x n rij . wj

Kriteria C1 dengan nilai bobot 0.25

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.10000,25 = 0.0250

= 0.06670,25 = 0.0166

= 0.10000,25 = 0.0250

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.03330,25 = 0.0083

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.10000,25 = 0.0250

Kriteria C2 dengan nilai bobot 0.25

= 0.12120,25 = 0.0303

= 0.12120,25 = 0.0303

= 0.06060,25 = 0.0151

= 0.09090,25 = 0.0227

= 0.06060,25 = 0.0151

= 0.09090,25 = 0.0227

= 0.09090,25 = 0.0227

= 0.06060,25 = 0.0151

= 0.09090,25 = 0.0227

= 0.09090,25 = 0.0227

= 0.12120,25 = 0.0303

Kriteria C3 dengan nilai bobot 0.25

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.10000,25 = 0.0250

= 0.13330,25 = 0.3333

= 0.06670,25 = 0.0166

= 0.03330,25 = 0.0083

= 0.13330,25 = 0.3333

= 0.03330,25 = 0.0083

= 0.10000,25 = 0.0250

Kriteria C4 dengan nilai bobot 0.10

= 0.12500,10 = 0.0125

= 0.09370,10 = 0.0093

= 0.09370,10 = 0.0093

= 0.09370,10 = 0.0093

= 0.06250,10 = 0.0062

= 0.12500,10 = 0.0125

= 0.09370,10 = 0.0093

= 0.03120,10 = 0.0031

= 0.09370,10 = 0.0093

= 0.09370,10 = 0.0093

= 0.09370,10 = 0.0093

Kriteria C5 dengan nilai bobot 0.15

= 0.12900,15 = 0.0193

= 0.06450,15 = 0.0097

= 0.12900,15 = 0.0193

= 0.06450,15 = 0.0097

= 0.09680,15 = 0.0145

= 0.12900,15 = 0.0193

= 0.06450,15 = 0.0097

= 0.03220,15 = 0.0048

= 0.12900,15 = 0.0193

= 0.03230,15 = 0.0049

= 0.12900,15 = 0.0193

Dari perhitungan hasil normalisasi matriks terbobot di atas, kemudian dapat diperoleh hasil ke dalam bentuk matriks sebagai berikut:

=

1. Menentukan Nilai Dari Fungsi Optimalisasi (Si)

Menentukan nilai dari fungsi optimalisasi dengan menjumlahkan nilai kriteria pada setiap alternatif dari hasil perkalian matriks dengan bobot yang telah dilakukan sebelumnya dengan rumus sebagai berikut.

Rumus:

Si = 1 *dij* : (*i* = 1,2, …, *m : j* = 1,2, …, *n*)

S0 = 0.0333 + 0.0303 + 0.0333 + 0.0125 + 0.0193 = 0.1287

S1 = 0.0166 + 0.0303 + 0.0166 + 0.0093 + 0.0097 = 0.0825

S2 = 0.0250 + 0.0151 + 0.0333 + 0.0093 + 0.0193 = 0.1020

S3 = 0.0166 + 0.0227 + 0.0166 + 0.0093 + 0.0097 = 0.0749

S4 = 0.0250 + 0.0151 + 0.0250 + 0.0062 + 0.0145 = 0.0858

S5 = 0.0333 + 0.0227 + 0.0333 + 0.0125 + 0.0193 = 0.1211

S6 = 0.0166 + 0.0227 + 0.0166 + 0.0093 + 0.0097 = 0.0749

S7 = 0.0083 + 0.0151 + 0.0083 + 0.0031 + 0.0048 = 0.0396

S8 = 0.0333 + 0.0227 + 0.0333 + 0.0093 + 0.0193 = 0.1179

S9 = 0.0166 + 0.0227 + 0.0083 + 0.0093 + 0.0049 = 0.0618

S10 = 0.0250 + 0.0303 + 0.0250 + 0.0093 + 0.0193 = 0.1089

1. Mencari Nilai (Ki) Dengan Cara Membagi Setiap Nilai Alternatif Dengan Alternatif 0 (A0).

Rumus:

Ki =

K0 = 1

K1 = 0.6410

K2 = 0.7925

K3 = 0.5820

K4 = 0.6667

K5 = 0.9409

K6 = 0.5820

K7 = 0.3077

K8 = 0.9161

K9 = 0.4802

K10 = 0.8462

Tabel 3.10 Hasil Penilaian

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Nama** | **S** | **K** |
| A00 | - | 0.1287 | K = 1 |
| A01 | Hafiz Sitepu | 0.0825 | K1 = 0.6410 |

Tabel 3.10 Hasil Penilaian (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Nama** | **S** | **K** |
| A02 | Ahmad Rahmad | 0.1020 | K2 = 0.7925 |
| A03 | Heru Pranata | 0.0750 | K3 = 0.5820 |
| A04 | Agung Alponi | 0.0858 | K4 = 0.6667 |
| A05 | Dika Radit | 0.1211 | K5 = 0.9409 |
| A06 | Dwi Susanto | 0.0750 | K6 = 0.5820 |
| A07 | Dandi Ilyas | 0.1896 | K7 = 0.3077 |
| A08 | Andreas | 0.1180 | K8 = 0.9161 |
| A09 | Derry Akbar | 0.1378 | K9 = 0.4802 |
| A10 | Rico Zahiri | 0.1089 | K10 = 0.8462 |

1. Melakukan Perangkingan Dari Hasil Perhitungan

Kemudian langkah selanjutnya yaitu menentukan tingkatan dari peringkat dari hasil perhitungan metode ARAS seperti dijelaskan pada tabel di bawah ini:

Tabel 3.11 Perangkingan

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Nama Alternatif** | **Nilai Akhir** | **Hasil** |
| A05 | Dika Radit | 0.9409 | Peringkat 1 |
| A08 | Andreas | 0.9161 | Peringkat 2 |
| A10 | Rico Zahiri | 0.8462 | Peringkat 3 |
| A02 | Ahmad Rahmad | 0.7925 | Peringkat 4 |
| A04 | Agung Alponi | 0.6667 | Peringkat 5 |
| A01 | Hafiz Sitepu | 0.6410 | Peringkat 6 |
| A06 | Dwi Susanto | 0.5820 | Peringkat 7 |

Tabel 3.11 Perangkingan (Lanjutan)

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Nama Alternatif** | **Nilai Akhir** | **Hasil** |
| A03 | Heru Pranata | 0.5820 | Peringkat 8 |
| A09 | Derry Akbar | 0.4802 | Peringkat 9 |
| A07 | Dandi Ilyas | 0.3077 | Peringkat 10 |

Berdasarkan hasil perhitungan dengan menggunakan metode ARAS diatas, dapat disimpulkan bahwa yang mendapatkan nilai akhir tertinggi dan berada pada peringkat pertama adalah alternatif dengan kode A05 atas nama Dika Radit dengan nilai akhir 0.9409.

**BAB IV**

**PEMODELAN DAN PERANCANGAN SISTEM**

* 1. **Pemodelan Sistem**

Pemodelan sistem bertujuan untuk membuat suatu pemodelan kerangka dasar sistem pendukung keputusan metode ARASyang akan digunakan, sistem masukan yang dibutuhkan, keluaran yang diharapkan, serta prosedur penggunaan sistem. Tahapan yang akan dilakukan dalam pemodelan sistem *Unified Modelling Language* diantaranya adalah *Use Case Diagram, Activity Diagram,* dan *Class Diagram.*

* + 1. ***Use Case Diagram***

Dalam pemodelan *Use Case Diagram* dilakukan pembuatan skanario *Use Case* *Diagram* *login*, kriteria, proses dan data alternatif. *Form login* berfungsi untuk menginput *username* dan *password* untuk masuk ke dalam menu utama. Berikut ini adalah skenario dari prosedur *Login*.

* + - 1. Skenario *Login*

Aktor : Admin

*use case* ini menggambarkan aktifitas admin memasukan *username* dan *password* untuk *login* ke sistem

Deskripsi :

**DAFTAR PUSTAKA**

[1] D. Pratama and A. Basry, “Rancang Bangun Sistem Perekomendasian Lokasi Usaha Menggunakan Metode Composite Performance Index Berbasis Laravel (Studi Kasus : Lokasi Usaha Di Jakarta),” *Tekinfo J. Bid. Tek. Ind. dan Tek. Inform.*, vol. 23, no. 2, pp. 24–38, 2022, doi: 10.37817/tekinfo.v23i2.2594.

[2] M. Arifin, “Sistem Informasi Geografis (SIG) Penentuan Tempat Pembuangan Akhir Sampah (TPA) Kabupaten Pamekasan Menggunakan Metode Composite Performance Index …,” *Semin. Nas. Hum. dan Apl. …*, vol. 8, no. 1, pp. 22–26, 2022, [Online]. Available: http://www.prosiding.uim.ac.id/index.php/sehati/article/view/216%0Ahttp://www.prosiding.uim.ac.id/index.php/sehati/article/download/216/114.

[3] P. Adytia, M. Fahmi, and R. Andrea, “Analisis Dalam Pendukung Keputusan Seleksi Reporter dengan Menerapkan Metode EDAS dan Pembobotan ROC,” *J. Media Inform. Budidarma*, vol. 7, no. 2, pp. 809–818, 2023, doi: 10.30865/mib.v7i2.6064.

[4] A. Karim, S. Esabella, M. Hidayatullah, and T. Andriani, “Sistem Pendukung Keputusan Aplikasi Bantu Pembelajaran Matematika Menggunakan Metode EDAS,” *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, 2022, doi: 10.47065/bits.v4i3.2494.

[5] I. Purnama, M. Bobbi, K. Nasution, A. Karim, and S. Trianovie, “Sistem Pendukung Keputusan Pemilihan Sales Supervisor Menerapkan Metode EDAS berdasarkan Pembobotan ROC,” vol. 5, no. 1, pp. 181–190, 2023, doi: 10.47065/bits.v5i1.3558.

[6] N. Nezafati, S. Razaghi, H. Moradi, S. Shokouhyar, and S. Jafari, “Promoting knowledge sharing performance in a knowledge management system: do knowledge workers’ behavior patterns matter?,” *VINE J. Inf. Knowl. Manag. Syst.*, vol. 53, no. 4, pp. 637–662, 2023, doi: 10.1108/VJIKMS-11-2020-0202.

[7] R. Safitri and I. Firdaus, “SPK Rekomendasi Pekerjaan Dengan Metode EDAS ( Studi Kasus : Lembaga Kursus dan Pelatihan Komputer Widya Informatika Selat Panjang ),” *J. Inf. Komput. Log.*, vol. 1, no. 4, 2020.

[8] L. Sinambela, L. Nababan, and J. Elnovreny, “Penerapan Metode CPI Dalam Penentuan Cara Terbaik Meningkatkan Kualitas Belajar Mengajar (Studi Kasus UPH Medan),” *J. Tek. Inform. Kaputama*, vol. 6, no. 2, pp. 718–724, 2022.

[9] R. Fachrizal, “IMPLEMENTASI ARAS ( Additive Ratio Assessment ) DALAM PEMILIHAN KASIR TERBAIK STUDI KASUS OUTLET CARDINAL STORE PLAZA MEDAN FAIR,” Sainteks, no. Januari, pp. 501–510, 2019.

[10] S. Widjaja and R. E. Hermanto, “Pengembangan Sistem Informasi Akademik Menggunakan Metode Unified Modeling Language Berbasis Website,” *Dinamik*, vol. 28, no. 1, pp. 25–34, 2023, doi: 10.35315/dinamik.v28i1.8980.

[11] N. Devi, D. Erwanto, and Y. Utomo, “Multitek Indonesia : Jurnal Ilmiah Multitek Indonesia : Jurnal Ilmiah,” *Multitek Indones. J. Ilm.*, vol. 12, no. 2, pp. 104–113, 2022.

[12] E. R. Subhiyakto and Y. P. Astuti, “Aplikasi Pembelajaran Class Diagram Berbasis Web Untuk Pendidikan Rekayasa Perangkat Lunak,” *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 11, no. 1, pp. 143–150, 2020.

[13] M Ali Maksum, “Pengertian XAMPP, Fungsi, dan Cara Menggunakannya,” 2022. https://www.dewaweb.com/blog/apa-itu-xampp/ (accessed Sep. 07, 2023).