



**MENENTUKAN PENUGASAN Pengerjaan *PROJECT*
MENGUNAKAN METODE *SIMPLE ADDITIVE*
WEIGHTING (SAW) PADA DIREKTORAT TIK,
DIREKTORAT JENDERAL PAJAK**

Tugas akhir
diajukan untuk melengkapi
persyaratan mencapai
gelar sarjana

NAMA : FAUZAN SAEF
NPM : 202143570038

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS INDRAPRASTA PGRI
2024**

LEMBAR PERSETUJUAN UJIAN TUGAS AKHIR

Nama : Fauzan Saef
NPM : 202143570038
Program Studi : Teknik Informatika
Fakultas : Teknik dan Ilmu Komputer
Judul : Menentukan Penugasan Pengerjaan *Project*
Menggunakan Metode *Simple Additive*
Weighting (SAW) pada Direktorat TIK,
Direktorat Jenderal Pajak

Telah diperiksa dan disetujui untuk diujikan

Pembimbing Materi

Pembimbing Teknik

Irawan Setiadi, M.Kom.
NIDN: 0306119101

Dr. Maria Cleopatra S.Kom., M.Pd.
NIDN: 0313108205

LEMBAR PENGESAHAN

Nama : Fauzan Saef
NPM : 202143570038
Program Studi : Teknik Informatika
Fakultas : Teknik dan Ilmu Komputer
Judul : Menentukan Penugasan Pengerjaan *Project*
Menggunakan Metode *Simple Additive*
Weighting (SAW) pada Direktorat TIK,
Direktorat Jenderal Pajak

Panitia Ujian

Ketua : Prof. Dr. H. Sumaryoto (.....)

Sekretaris : Ir. H. Soepardi Harris, M.T. (.....)

Anggota :

No	Nama	Tanda Tangan
1.	Irawan Setiadi, M.Kom.	
2.	Dr. Maria Cleopatra S.kom., M.Pd.	
3.		

LEMBAR PERNYATAAN

Yang bertandatangan di bawah ini:

Nama : Fauzan Saef
NPM : 202143570038
Program Studi : Teknik Informatika

Dengan ini menyatakan bahwa skripsi/tugas akhir dengan judul “Menentukan Penugasan Pengerjaan *Project* Menggunakan Metode *Simple Additive Weighting (SAW)* pada Direktorat TIK, Direktorat Jenderal Pajak” beserta seluruh isinya adalah benar-benar karya saya sendiri. Saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika ilmu yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung risiko/sanksi apabila dikemudian hari ditemukan adanya pelanggaran etika keilmuan atau ada klaim dari pihak lain terhadap keaslian karya saya ini sesuai dengan Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional Bab VI Pasal 25 Ayat 2 dan Bab XX Pasal 70.

Demikian pernyataan ini saya buat untuk dimanfaatkan sesuai dengan keperluan.

Yang Menyatakan

Fauzan Saef

ABSTRAK

- A. Fauzan Saef, NPM: 202143570038
- B. **Menentukan Penugasan Pengerjaan Project Menggunakan Metode *Simple Additive Weighting* (SAW) pada Direktorat TIK, Direktorat Jenderal Pajak.** Skripsi/ Tugas Akhir: Jakarta: Fakultas Teknik dan Ilmu Komputer: Program Studi Teknik Informatika: Universitas Indraprasta Persatuan Guru Republik Indonesia, Agustus, 2024.
- C. Jumlah halaman romawi xii + 5 Bab + 91 halaman
- D. Kata Kunci: SPK, Penugasan *Project*, SAW, Java
- E. Tujuan dari penelitian ini adalah membangun sistem atau aplikasi pendukung keputusan dengan metode SAW untuk mengatasi masalah menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP. Metodologi penelitian yang digunakan dalam sistem pengolahan datanya adalah metode studi Pustaka yang merupakan metode pengumpulan data dan informasi melalui dokumen-dokumen, baik dokumen tertulis, foto, gambar, maupun dokumen elektronik yang dapat mendukung dalam proses penulisan, serta dilengkapi dengan metode wawancara. Algoritma yang digunakan dalam penyelesaian masalah adalah *Simple Additive Weighting*. Hasil dari penelitian ini adalah adanya suatu perangkat aplikasi yang dibuat dengan bahasa pemrograman Java dan penyimpanan database dari MySQL. Aplikasi ini akan memberikan hasil dari pemrosesan data berupa urutan pegawai yang akan ditugaskan untuk mengerjakan suatu *project* pengembangan aplikasi.
- F. Daftar Pustaka: 1. Buku 10 buah (tahun 2017-2023)
 2. Jurnal 14 buah (tahun 2017-2024)
- G. Pembimbing: 1. Irawan Setiadi, M.Kom (Pembimbing Materi)
 2. Dr. Maria Cleopatra S.Kom., M.Pd. (Pembimbing Teknik)

MOTTO

“DUIT (Doa, Usaha, Ikhtiar, Tawakkal)”

Skripsi ini saya persembahkan untuk orang tua, istri, keluarga, serta teman-teman saya yang senantiasa selalu mendoakan dan mendukung saya pada setiap proses kehidupan yang saya jalani.

PRAKATA

Dengan memanjatkan puji Syukur ke hadirat Allah SWT yang telah melimpahkan Rahmat dan karunia-Nya kepada penulis sehingga akhirnya penulis dapat menyelesaikan tugas akhir ini.

Tugas Akhir yang berjudul **Menentukan Penugasan Pengerjaan Project Menggunakan Metode *Simple Additive Weighting* (SAW) pada Direktorat TIK, Direktorat Jenderal Pajak** ini ditulis untuk memenuhi salah satu syarat guna memperoleh gelar kesarjanaan pada Universitas Indraprasta PGRI. Pada kesempatan yang baik ini, izinkanlah penulis menyampaikan rasa hormat dan ucapan terima kasih kepada semua pihak yang telah memberikan bantuan dan dorongan dalam menyelesaikan skripsi ini, terutama kepada:

1. Bapak Irawan Setiadi, M.Kom. selaku Dosen Pembimbing Materi Universitas Indraprasta PGRI.
2. Ibu Dr. Maria Cleopatra S.Kom., M.Pd. selaku Dosen Pembimbing Teknik Universitas Indraprasta PGRI.
3. Bapak Prof. Dr. H. Sumaryoto selaku Rektor Universitas Indraprasta PGRI.
4. Bapak Ir. H. Soepardi Harris, M.T. selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI.
5. Ibu Atie Ernawati, M.T. selaku Wakil Dekan Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI.
6. Ibu Mei Lestari, M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI.
7. Ibu Ni Wayan Parwati, S., S.T., M.M., M.Kom. selaku Sekretaris Program Studi Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI.
8. Seluruh Dosen dan Staff Program Studi Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Indraprasta PGRI.
9. Orang tua, istri, dan keluarga saya yang senantiasa mencintai, menyayangi, mendukung, dan mendoakan saya.
10. Teman-teman seperjuangan Teknik Informatika Angkatan 2021

11. Pimpinan dan pegawai Direktorat Teknologi Informasi dan Komunikasi,
Direktorat Jenderal Pajak.

12. Serta semua pihak yang tidak dapat saya sebutkan satu persatu. Terima kasih.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan, baik dalam bentuk, isi, dan penyampaian. Oleh karena itu, kritik dan saran yang bersifat membangun dari berbagai pihak akan penulis terima dengan tangan terbuka. Semoga skripsi/tugas akhir ini dapat memenuhi tujuan dan sasaran pembuatan.

Jakarta, Agustus 2024

Fauzan Saef

DAFTAR ISI

LEMBAR PERSETUJUAN UJIAN TUGAS AKHIR.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
ABSTRAK	iv
MOTTO	v
PRAKATA.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Identifikasi Masalah.....	5
C. Batasan Masalah	5
D. Rumusan Masalah.....	6
E. Tujuan Penelitian	6
F. Manfaat Penelitian	7
BAB II TINJAUAN PUSTAKA	9
A. Landasan Teori	9
1. Sistem Pendukung Keputusan.....	9
2. Metode <i>Simple Additive Weighting</i> (SAW)	13
3. Pengertian Penugasan (<i>Task Assignment</i>)	16
4. <i>Project</i> Pengembangan Aplikasi	18
5. <i>Unified Modeling Language</i>	20
6. <i>MySQL</i>	26
7. Bahasa Pemrograman <i>Java</i>	27
B. Penelitian yang Relevan.....	29

BAB III METODE PENELITIAN	31
A. Waktu dan Tempat Penelitian.....	31
1. Waktu Penelitian	31
2. Tempat Penelitian	31
B. Tahapan Penelitian	32
C. Algoritma	38
BAB IV HASIL DAN PEMBAHASAN.....	43
A. Definisi Masalah	43
B. Pembahasan Algoritma	44
C. Pemodelan Perangkat Lunak.....	50
1. Pemodelan Perangkat Lunak dengan UML	50
2. Rancangan Layar	69
3. Tampilan Layar	77
D. Kelebihan dan Kelemahan Penelitian	89
BAB V PENUTUP	90
A. SIMPULAN	90
B. SARAN	91
DAFTAR PUSTAKA	
DAFTAR NARASUMBER	
DAFTAR RIWAYAT HIDUP	
LISTING PROGRAM	

DAFTAR TABEL

Tabel 2. 1 Simbol use case diagram	22
Tabel 2. 2 Simbol Activity Diagram	23
Tabel 2. 3 Simbol Sequence Diagram	24
Tabel 2. 4 Simbol Class Diagram.....	25
Tabel 2. 5 Penelitian yang relevan	29
Tabel 3. 1 Waktu penelitian.....	31
Tabel 4. 1 Penentuan Kriteria.....	44
Tabel 4. 2 Rating Kepentingan.....	44
Tabel 4. 3 Bobot Kriteria Kemampuan Coding	45
Tabel 4. 4 Bobot Kriteria Project yang Sedang Dikerjakan	45
Tabel 4. 5 Bobot Kriteria Pengalaman	45
Tabel 4. 6 Bobot Kriteria Pelatihan.....	46
Tabel 4. 7 Bobot Kriteria Penguasaan Stack	46
Tabel 4. 8 Daftar Kandidat	47
Tabel 4. 9 Pembobotan setiap Alternatif	47
Tabel 4. 10 Bobot Preferensi	48
Tabel 4. 11 Tabel Normalisasi	48
Tabel 4. 12 Perhitungan Bobot Perankingan.....	49
Tabel 4. 13 Perhitungan Alternatif	49
Tabel 4. 14 Skenario Use Case Login	51
Tabel 4. 15 Skenario Use Case Monitoring Project	52
Tabel 4. 16 Skenario Use Case Monitoring	52
Tabel 4. 17 Skenario Use Case Request Project	53
Tabel 4. 18 Skenario Use Case Report Project	54
Tabel 4. 19 Skenario Use Case Manajemen Project	55
Tabel 4. 20 Skenario Use Case Kriteria Developer.....	55

DAFTAR GAMBAR

Gambar 2. 1 Komponen Sistem Pendukung Keputusan	10
Gambar 2. 2 Formula normalisasi berdasarkan atribut	15
Gambar 2. 3 Formula normalisasi	16
Gambar 2. 4 Formula nilai akhir alternatif	16
Gambar 3. 1 Tempat penelitian	32
Gambar 3. 2 Tahapan penelitian.....	32
Gambar 3. 3 Metode Waterfall.....	36
Gambar 3. 4 Kerangka Kerja Metode SAW.....	39
Gambar 4. 1 Use Case Diagram.....	51
Gambar 4. 2 Diagram Activity Login	56
Gambar 4. 3 Diagram Activity sub-menu Monitoring.....	57
Gambar 4. 4 Diagram Activity Lihat data project.....	58
Gambar 4. 5 Diagram Activity Tambah Data Project	58
Gambar 4. 6 Diagram Activity Surat Penugasan Project	59
Gambar 4. 7 Diagram Activity Ajukan Project	60
Gambar 4. 8 Diagram Activity Generate Team.....	60
Gambar 4. 9 Diagram Activity Cetak Hasil Rekomendasi Programmer	61
Gambar 4. 10 Diagram Activity sub-menu Report	62
Gambar 4. 11 Diagram Activity sub-menu Kriteria Developer	63
Gambar 4. 12 Diagram Activity Tambah Kriteria Developer	63
Gambar 4. 13 Diagram Activity Ubah Kriteria Developer	64
Gambar 4. 14 Diagram Activity Hapus Kriteria Developer.....	64
Gambar 4. 15 Diagram Sequence Login	65
Gambar 4. 16 Diagram Sequence Monitoring Project.....	66
Gambar 4. 17 Diagram Sequence Request Project	67
Gambar 4. 18 Diagram Sequence Report.....	67
Gambar 4. 19 Diagram Sequence Kriteria Developer	68

Gambar 4. 20 Class Diagram SPK Menentukan Penugasan Project	69
Gambar 4. 21 Rancangan Form Login.....	70
Gambar 4. 22 Rancangan Layar Menu Home.....	70
Gambar 4. 23 Rancangan Layar sub-menu Monitoring.....	71
Gambar 4. 24 Rancangan Layar sub-menu Request Project Aktor Analis	71
Gambar 4. 25 Rancangan Layar sub-menu Request Project Aktor Kasi	72
Gambar 4. 26 Rancangan Layar sub-menu Request Project Aktor Programmer..	72
Gambar 4. 27 Rancangan Layar Tambah Project.....	73
Gambar 4. 28 Rancangan Layar Generate Team.....	73
Gambar 4. 29 Rancangan Layar sub-menu Report	74
Gambar 4. 30 Rancangan Layar sub-menu Kriteria Developer	74
Gambar 4. 31 Rancangan Layar Tambah Kriteria Developer	75
Gambar 4. 32 Rancangan Layar Cetak Surat Penugasan	75
Gambar 4. 33 Rancangan Layar Cetak Rekapitulasi Pengembangan Project...	76
Gambar 4. 34 Rancangan Layar Cetak Hasil Rekomendasi Penugasan Project	76
Gambar 4. 35 Tampilan Layar Menu Login.....	77
Gambar 4. 36 Tampilan Layar Menu Home	77
Gambar 4. 37 Tampilan Layar sub-menu Monitoring.....	78
Gambar 4. 38 Tampilan Layar sub-menu Request Project Aktor Analis	79
Gambar 4. 39 Tampilan Layar sub-menu Request Project Aktor Kasi	79
Gambar 4. 40 Tampilan Layar sub-menu Request Project Aktor Programmer	80
Gambar 4. 41 Tampilan Layar Tambah Project	81
Gambar 4. 42 Tampilan Layar Generate Team	81
Gambar 4. 43 Tampilan Layar Cetak Hasil Rekomendasi Penugasan Project..	82
Gambar 4. 44 Tampilan Layar Cetak Surat Penugasan	83
Gambar 4. 45 Tampilan Layar sub-menu Report	84
Gambar 4. 46 Tampilan Layar Cetak Rekapitulasi Pengembangan Project.....	85
Gambar 4. 47 Tampilan Layar sub-menu Kriteria Developer.....	86
Gambar 4. 48 Tampilan Layar Tambah Kriteria Developer.....	87

BAB I

PENDAHULUAN

A. Latar Belakang

Di era digital saat ini, Teknologi Informasi (TI) telah berkembang menjadi salah satu pilar utama dalam transformasi berbagai sektor kehidupan dan industri. TI mencakup penggunaan alat dan sistem teknologi untuk mengelola, menyimpan, dan mendistribusikan informasi secara efisien. Perkembangan pesat dalam bidang ini telah membawa perubahan signifikan dalam cara organisasi beroperasi, individu berinteraksi, dan layanan publik yang disediakan. Perubahan teknologi ini mendorong reformasi mendalam pada berbagai aspek kehidupan sehari-hari. Di dunia bisnis, misalnya, TI telah mengubah proses operasional dan strategi perusahaan dengan memfasilitasi otomatisasi, integrasi sistem, dan pengambilan keputusan berbasis data. Dalam pendidikan, TI telah memperkenalkan metode pembelajaran baru yang lebih fleksibel dan terjangkau, memungkinkan akses yang lebih luas ke sumber daya pendidikan. Dalam pemerintahan merujuk pada penerapan teknologi informasi untuk meningkatkan cara pemerintah beroperasi, memberikan layanan, dan berinteraksi dengan warganya. Tujuannya adalah untuk menciptakan sistem pemerintahan yang lebih efisien, transparan, dan responsif terhadap kebutuhan masyarakat.

Pemerintahan terus menghasilkan inovasi-inovasi baru dalam kehidupan bermasyarakat dan bernegara. Inovasi teknologi tidak henti-hentinya

terus terjadi salah satunya adalah inovasi teknologi di bidang perpajakan. Direktorat Jenderal Pajak (DJP), yang berada di bawah Kementerian Keuangan Indonesia, memiliki wewenang untuk merumuskan dan melaksanakan kebijakan serta standarisasi teknis di bidang perpajakan. Selama dua dekade terakhir, Direktorat Jenderal Pajak (DJP) telah melakukan transformasi digital untuk meningkatkan kualitas layanan dan efektivitas pengawasan terhadap kepatuhan wajib pajak. Untuk mendukung transformasi digital oleh DJP, Direktorat TIK memainkan peran penting dalam pengembangan aplikasi dan sistem teknologi informasi yang mendukung berbagai fungsi perpajakan. Transformasi digital membutuhkan banyaknya pengembangan aplikasi demi mempertahankan eksistensi layanan atau produk yang dimiliki oleh DJP. Oleh karena itu sangat penting untuk melakukan manajemen *project* pengembangan aplikasi tersebut. Manajemen proyek pengembangan aplikasi di Direktorat TIK DJP bertujuan untuk memastikan aplikasi yang dikembangkan sesuai dengan kebutuhan pengguna, mengelola sumber daya dan anggaran dengan efisien, serta menjamin kualitas dan keandalan aplikasi melalui pengujian yang menyeluruh. Selain itu, manajemen proyek berfokus pada identifikasi dan pengelolaan risiko serta masalah yang mungkin muncul, memastikan proyek diselesaikan tepat waktu dan sesuai anggaran, serta memberikan dukungan dan pemeliharaan setelah peluncuran untuk menjaga performa aplikasi.

Saat ini pada Direktorat TIK proses manajemen *project* sudah baik, dimana permintaan pengembangan aplikasi dari *Business Owner* masuk ke Direktorat TIK melewati serangkaian prosedur hingga sampai kepada proses

pengembangan aplikasi yang dikerjakan oleh tim *developer*. Permintaan pengerjaan *project* pengembangan aplikasi yang diajukan oleh *business owner* ke Direktorat TIK diserahkan kepada subdirektorat terkait. Kepala subdirektorat kemudian menentukan seksi yang tepat untuk melaksanakan *project* tersebut. Selanjutnya, analis berdiskusi bersama dengan kepala seksi memilih *developer* dari seksi yang telah ditunjuk tersebut. *Developer* dipilih berdasarkan pemilihan subjektif antara analis dan kepala seksi atau kepala subdirektorat. Atasan mengevaluasi dan melakukan perbandingan kinerja para *developer* secara manual karena belum adanya dokumentasi yang memuat *skill* dan pengalaman *developer* untuk menjadi bahan pertimbangan menentukan penugasan pengerjaan *project*. Banyaknya permintaan pengembangan aplikasi yang diterima, membuat tim *developer* memiliki beban kerja yang cukup banyak dan kurang merata, yang mengakibatkan kurang optimalnya penyelesaian *project* tersebut. Ketika banyaknya *project* yang sedang dikerjakan secara bersamaan, harus mempertimbangkan beberapa strategi untuk mengelola tim *developer* tersebut. Salah satunya adalah menentukan penugasan pengerjaan *project*. Hal tersebut merupakan salah satu kendala yang ada di Direktorat TIK DJP. Tidak adanya sistem yang dapat menentukan penugasan pengerjaan *project* menimbulkan beberapa masalah seperti mundurnya tenggat waktu pengerjaan *project*, kurang meratanya penyebaran beban kerja perorangan.

Untuk menentukan penugasan pengerjaan *project* yang lebih efektif dan efisien, dapat menggunakan sistem pendukung keputusan sebagai alat yang dapat mempermudah pekerjaan manajemen *project* pengembangan aplikasi di

Direktorat TIK DJP. Hasil dari sistem pendukung keputusan ini akan menjadi alat bantu untuk menentukan penugasan pengerjaan *project* pengembangan aplikasi di Direktorat TIK, DJP, sehingga *project* yang akan berlangsung, dapat dikerjakan oleh orang yang tepat dan tidak mengganggu beberapa *project* yang sedang berlangsung secara bersamaan. Salah satu metode Sistem Pendukung Keputusan yang dapat digunakan adalah metode SAW (*Simple Additive Weighting*). Inti dari metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Menggunakan metode SAW dapat meningkatkan kualitas pengambilan keputusan dalam penugasan proyek dengan memberikan dasar kuantitatif yang jelas untuk memilih programmer yang paling sesuai berdasarkan berbagai kriteria. Ini membantu memastikan bahwa keputusan tidak hanya berdasarkan intuisi atau preferensi subjektif, tetapi didukung oleh analisis yang sistematis. Sehingga metode SAW bisa menjadi metode yang tepat untuk mengembangkan sistem pendukung keputusan untuk permasalahan yang terjadi di Direktorat TIK DJP.

Berdasarkan permasalahan di atas, dapat disimpulkan bahwa diperlukannya sistem atau aplikasi yang dapat memudahkan Perusahaan untuk menentukan penugasan pengerjaan *project* pengembangan aplikasi yang diminta oleh *Business owner* kepada Direktorat TIK DJP. Maka Penulis memberikan judul tugas akhir ini dengan judul “Menentukan Penugasan Pengerjaan *Project* Menggunakan Metode *Simple Additive Weighting* (SAW) pada Direktorat TIK, Direktorat Jenderal Pajak.”

B. Identifikasi Masalah

Berdasarkan permasalahan diatas, maka penulis menyimpulkan beberapa hal sebagai berikut:

1. Banyaknya *project* pengembangan aplikasi yang diminta kepada Direktorat TIK DJP dari internal maupun eksternal Perusahaan.
2. Para *developer* di Direktorat TIK dipilih berdasarkan penilaian subjektif atasan yang tidak jarang terjadi bias saat penentuan.
3. Belum adanya pendataan terkait *skill* dan pengalaman dari para *developer*
4. Para *developer* di Direktorat TIK, Direktorat Jenderal Pajak memiliki beban kerja yang cukup banyak dan kurang merata.
5. Evaluasi dan perbandingan antara *developer* dilakukan secara manual sehingga berlangsung lebih lama.
6. Belum adanya sistem yang dapat menentukan penugasan pengerjaan *project* pada Direktorat TIK DJP sehingga pengerjaan *project* pengembangan aplikasi kurang efektif dan efisien.

C. Batasan Masalah

Batasan masalah dalam melaksanakan pengembangan sistem informasi ini adalah sebagai berikut:

1. Metode yang digunakan untuk penelitian ini adalah SAW.
2. Sistem ini mencakup daftar *project* pengembangan aplikasi dan daftar pengembang aplikasi (*developer/programmer*) mencakup kriteria kemampuan yang sesuai untuk digunakan sebagai atribut dalam menentukan penugasan pengerjaan *project*.

3. *Form input* pada sistem ini mencakup pengisian data *project* yang akan dikerjakan, data kriteria kemampuan pengembang aplikasi (*developer/programmer*), dan data referensi profil pegawai.
4. *Output* dari sistem ini adalah surat penugasan pengerjaan *project* aplikasi, laporan preferensi pegawai pengerjaan *project*, dan laporan permintaan pengembangan *project*.
5. Ruang lingkup penelitian ini dilakukan di Direktorat TIK, Direktorat Jenderal Pajak.

D. Rumusan Masalah

Berdasarkan latar belakang masalah dan batasan masalah yang dikemukakan maka dapat dirumuskan masalah sebagai berikut:

1. Bagaimana membuat sistem atau aplikasi untuk mengatasi masalah menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP?
2. Bagaimana penerapan metode SAW dalam aplikasi untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP?
3. Bagaimana penyajian data hasil dari penerapan metode SAW dalam aplikasi untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP?

E. Tujuan Penelitian

1. Membangun sistem atau aplikasi pendukung keputusan dengan metode SAW untuk mengatasi masalah menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP.

2. Penerapan metode SAW dalam pengembangan aplikasi untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP dengan kriteria pembobotan.
3. Menyajikan data hasil dari penerapan metode SAW dalam aplikasi untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP berupa daftar pegawai yang sudah diurutkan berdasarkan pembobotan menggunakan metode SAW.

F. Manfaat Penelitian

1. Manfaat Teoritis

Penelitian ini diharapkan dapat memberikan kontribusi teoritis dalam pengembangan model sistem pendukung keputusan yang berbasis SAW serta dapat digunakan sebagai referensi teoritis bagi peneliti dan praktisi lain dalam merancang dan menerapkan SPK di berbagai konteks, khususnya dalam manajemen proyek TI.

2. Manfaat Praktis

Penelitian ini diharapkan dapat memberikan manfaat

- a. Bagi Direktorat TIK Direktorat Jenderal Pajak

- 1) Tersedianya sistem pendukung Keputusan untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP yang dapat meningkatkan efektivitas dan efisiensi karyawan dalam bekerja.

- 2) Sebagai alat dokumentasi yang jelas mengenai proses penugasan dan memudahkan pemantauan serta akuntabilitas keputusan yang diambil.

b. Bagi Universitas

- 1) Penelitian ini diharapkan dapat digunakan untuk menambah referensi sebagai bahan penelitian lanjutan yang lebih mendalam pada masa yang akan datang

c. Bagi Penulis

- 1) Dapat menambah wawasan tentang Sistem Pendukung Keputusan dengan menggunakan metode SAW.
- 2) Untuk menyelesaikan kewajiban Tugas Akhir dalam rangka memperoleh gelar sarjana.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Sistem Pendukung Keputusan

a. Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem yang dapat membantu (*support*) para pengambil keputusan (*decision maker*) membuat keputusan yang logis, rasional, terstruktur, dan benar tentang masalah yang kompleks dengan banyaknya parameter yang harus dipertimbangkan (Utama, 2017:24).

Sistem Pendukung Keputusan adalah sistem berbasis komputer yang memfasilitasi proses pengambilan keputusan dengan menyediakan alat untuk menganalisis data dan alternatif, serta memberikan rekomendasi berbasis model (Sharda et al., 2018:13)

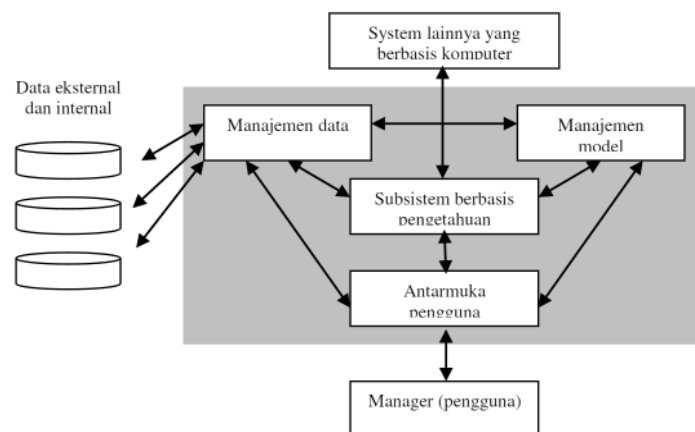
Sistem pendukung keputusan diartikan sebagai sistem informasi yang memberikan dukungan pada aktivitas dalam proses pengambilan keputusan di dalam organisasi. Sistem ini mencakup pekerjaan dan subsistem berbasis komputer interaktif yang membantu dalam menerapkan teknologi utama untuk menghasilkan sistem komputasi (Fernando & Baldeovar, 2022:13).

Berdasarkan beberapa pengertian di atas dapat diartikan bahwa Sistem Pendukung Keputusan (SPK) adalah sistem berbasis komputer

yang membantu para pengambil keputusan dalam proses pengambilan keputusan yang kompleks. SPK sebagai alat untuk menganalisis data dan alternatif, memberikan rekomendasi berbasis model, serta mendukung aktivitas keputusan melalui teknologi dan subsistem interaktif. Secara keseluruhan, SPK bertujuan untuk membantu menghasilkan keputusan yang logis, rasional, dan terstruktur dalam organisasi.

b. Komponen Sistem Pendukung Keputusan

Sistem Pendukung Keputusan memiliki komponen yang terdiri dari empat subsistem menurut Turban dalam Rahmansyah et al., 2021:8-9 yaitu:



Gambar 2. 1
Komponen Sistem Pendukung Keputusan
Sumber: Rahmansyah, 2021

- 1) Subsistem Data (*Data Subsystem*) adalah komponen Sistem Pendukung Keputusan yang menyediakan data ke sistem. Termasuk *database* yang mengandung data relevan dan diatur oleh *Database Management System* (DBMS).
- 2) Subsistem Model (*Model Subsystem*) adalah suatu peniruan dari alam nyata. Hal yang perlu diperhatikan adalah dalam menyimpan berbagai model harus tetap menjaga fleksibilitasnya. Melibatkan model finansial, statistik, *management science*, atau berbagai model kualitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis. Keunikan dari Sistem Pendukung adalah kemampuannya dalam mengintegrasikan data dengan model-model keputusan.
- 3) Subsistem Dialog (*User System Interface*) adalah komponen Sistem Pendukung Keputusan yang merupakan alat komunikasi antara *user* dan sistem. Melalui subsistem ini, sistem diartikulasikan dan diimplementasikan sehingga *user* dapat berkomunikasi dengan sistem yang dirancang. *User* dapat memberikan perintah pada Sistem Pendukung Keputusan melalui subsistem ini. Fasilitas yang dimiliki subsistem ini terdiri dari: bahasa aktif (perangkat keras yang digunakan untuk berkomunikasi dengan sistem), bahasa tampilan (perangkat yang digunakan sebagai sarana untuk menampilkan sesuatu), basis pengetahuan (yang harus diketahui *user* agar pemakaian sistem efektif).

- 4) Subsistem Pengetahuan adalah subsistem lainnya yang dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

c. Manfaat dan Keterbatasan Sistem Pendukung Keputusan

Sistem Pendukung Keputusan dapat memberikan berbagai manfaat bagi penggunanya (Rahmansyah et al., 2021:6-8), yaitu:

- 1) Memperluas kemampuan pengambilan keputusan dalam pemrosesan data atau informasi bagi *user*.
- 2) Membantu menghemat waktu dalam mengambil keputusan untuk menyelesaikan masalah, terutama berbagai masalah yang kompleks dan tidak terstruktur.
- 3) Menghasilkan solusi yang lebih cepat dan lebih handal.
- 4) Menjadi stimulus bagi pengambil keputusan dalam memahami masalah. Mengingat sistem pendukung keputusan mampu memberikan alternatif.
- 5) Memberikan informasi tambahan untuk membenarkan posisi pembuat keputusan.

Selain berbagai manfaat yang sudah dikemukakan di atas, Sistem Pendukung Keputusan memiliki berbagai keterbatasan juga, yaitu:

- 1) Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan yang sebenarnya.

- 2) Kemampuan suatu sistem pendukung keputusan terbatas pada pengetahuan dasar serta model dasar yang dimilikinya.
- 3) Proses-proses yang dapat dilakukan oleh sistem pendukung keputusan biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya
- 4) Sistem pendukung keputusan tidak memiliki intuisi seperti yang dimiliki oleh manusia. Karena sistem pendukung keputusan hanya suatu Kumpulan perngkat keras, perangkat lunak, dan sistem operasi yang tidak dilengkapi oleh kemampuan berpikir. Secara tidak langsung, sistem pendukung keputusan berlandaskan pada kemampuan dari sebuah sistem berbasis komputer dan dapat melayani penyelesaian masalah.

2. Metode *Simple Additive Weighting* (SAW)

Metode SAW merupakan salah satu metode pengambilan keputusan dalam menghadapi situasi *Multi-Attribute Decision Making* (MADM) yang digunakan untuk mengevaluasi masalah pengambilan keputusan sebagai model penyeleksi untuk memilih alternatif terbaik dari sejumlah alternatif dengan kriteria tertentu (Taherdoost, 2023:21). Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Menurut Kusumadewi dalam Rahmansyah *et al.*, 2021:39 konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Metode SAW biasanya digunakan untuk menyelesaikan masalah

penyeleksian dalam sebuah sistem pendukung keputusan yang memiliki banyak atribut (Noval *et al.*, 2020:117).

Metode SAW memiliki beberapa kelebihan yaitu memiliki kemampuan untuk menyeimbangkan antar kriteria, intuitif bagi pengambil keputusan, perhitungan yang sederhana, tidak diperlukan pemrograman yang rumit, menggunakan nilai yang dinormalisasi untuk membantu menentukan perbedaan secara visual antara objek yang dibandingkan. Selain beberapa kelebihan tersebut, metode SAW juga memiliki beberapa kekurangan yaitu menggeser kriteria minimalisasi ke maksimalisasi pada konsep hitungan utama, mengubah nilai normalisasi matriks negatif menjadi positif, hasil yang diperoleh belum tentu logis, bobot atribut dan matriks keputusan harus disediakan (Taherdoost, 2023:22)

Metode SAW membutuhkan proses normalisasi matriks keputusan ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada. Dalam proses perancangan database relasional, normalisasi dikenal sebagai teknik perancangan dan sering digunakan sebagai panduan saat mendesain database. Normalisasi pada dasarnya adalah proses dua langkah: menghapus grup berulang untuk menyediakan data dalam bentuk tabel, dan kemudian menghapus data duplikat dari tabel relasional. Teori normalisasi didasarkan pada konsep bentuk normal. Metode SAW terdiri dari dua atribut yaitu kriteria keuntungan (*benefit*) dan kriteria biaya (*cost*). Perbedaan antara dua kriteria ini adalah dalam pemilihan kriteria ketika mengambil keputusan.

Langkah-langkah dalam menggunakan metode SAW adalah:

- 1) Menentukan Alternatif (A) (kandidat)
- 2) Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan (C)
- 3) Menentukan nilai rating kecocokan setiap alternatif pada setiap kriteria
- 4) Menentukan bobot preferensi atau tingkat kepentingan (W) setiap kriteria.
- 5) Membuat tabel rating kecocokan dari setiap alternatif pada setiap kriteria.
- 6) Membuat matriks keputusan berdasarkan kriteria.
- 7) Melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut keuntungan (max) dan atribut biaya (min) dengan formula:

$$r_{ij} = \frac{\text{Min}_i x_{ij}}{x_{ij}}$$

$$r_{ij} = \frac{x_{ij}}{\text{Max}_i x_{ij}}$$

Gambar 2. 2
Formula normalisasi berdasarkan atribut
Sumber: Rahmansyah, 2021

- 8) Hasil dari nilai rating kinerja ternormalisasi (r_{ij}) membentuk matriks ternormalisasi.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} \\ \frac{\min_i x_{ij}}{x_{ij}} \end{cases}$$

Gambar 2. 3
Formula normalisasi
Sumber: Rahmansyah, 2021

- 9) Hasil akhir nilai alternatif (V_i) diperoleh dari penjumlahan dari setiap perkalian baris matriks ternormalisasi dengan bobot preferensi (W) yang bersesuaian elemen kolom matrik (W). Hasil perhitungan nilai V_i yang lebih besar mengindikasikan bahwa alternatif terbaik.

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

Gambar 2. 4
Formula nilai akhir alternatif
Sumber: Rahmansyah, 2021

3. Pengertian Penugasan (*Task Assignment*)

Menentukan penugasan merupakan proses manajemen sebuah *project* dengan mengorganisasikan dan mengelola atau menempatkan orang-orang yang terlibat, sehingga orang-orang tersebut dapat dimanfaatkan potensinya secara efektif dan efisien (Belferik et al., 2023:67).

Penugasan adalah masalah keputusan dalam proses kerja, di mana beban kerja dan tanggung jawab suatu tugas dialokasikan ke berbagai unit

organisasi, yang akan menangani bagian masing-masing dalam proyek bersama (Su et al., 2020).

Penugasan merupakan proses menugaskan suatu tugas atau tanggung jawab tertentu kepada individu atau tim dalam suatu organisasi. Hal ini termasuk menentukan yang akan bertanggung jawab untuk menyelesaikan tugas, menyediakan informasi dan sumber daya yang diperlukan, serta menetapkan tujuan yang jelas untuk hasil yang diinginkan. Dengan membagi tugas antar individu dalam tim yang mengerjakan suatu *project*, dapat membantu mencapai tujuan (jadwal penyelesaian *project*) dan menghindari risiko peningkatan waktu dan biaya tenaga kerja (Görsch et al., 2024:433-434).

Faktor-faktor pertimbangan dalam penugasan adalah:

- 1) Keterampilan, pengetahuan, dan keahlian individu
- 2) Ketersediaan dan beban kerja anggota tim
- 3) Batas waktu dan prioritas tugas
- 4) Persyaratan komunikasi dan kolaborasi
- 5) Ketergantungan dan hubungan antar tugas
- 6) Pertimbangan pengembangan individu atau peluang pertumbuhan skillnya
- 7) Menyeimbangkan beban kerja dan menghindari beban berlebihan
- 8) Pertimbangan penggunaan *task assignment tools* untuk kemudahan pembagian tugas

Proses penugasan ini menjadi hal yang penting dalam proses pengerjaan *project* karena berfungsi untuk memperjelas ekspektasi, memungkinkan anggota tim untuk fokus pada tugas-tugas tertentu pada masing-masing individu, dan mendukung komunikasi yang lebih baik dalam tim *project*, sehingga meningkatkan koordinasi dan kolaborasi dalam mencapai tujuan bersama. Proses penugasan ini juga menjadi salah satu factor keberhasilan pengerjaan *project*.

4. *Project* Pengembangan Aplikasi

Proyek didefinisikan sebagai kumpulan kegiatan lintas fungsi dan multidisiplin yang dilaksanakan di berbagai tingkat dalam struktur organisasi, dan dapat melibatkan mulai dari satu orang hingga ribuan pemangku kepentingan dengan keterampilan, disiplin, dan peran profesional yang berbeda-beda (Spezie & Bragantini, 2023).

Project adalah usaha atau bisnis yang bersifat temporer dengan jangka waktu tertentu sejak dimulainya pekerjaan. Walaupun masih dalam tahap pengembangan, dampak jangka panjang dari *project* tersebut dapat dirasakan oleh perusahaan (Belferik et al., 2023:6). *Project* memiliki ciri-ciri yaitu: memiliki tujuan dan kebutuhan yang spesifik, memiliki batasan waktu dan ruang lingkup pengerjaan, dilaksanakan secara terencana, terkoordinasi, dan terkendali, serta *project* mengelola tiga aspek terdiri dari biaya, waktu, dan sumber daya.

Terdapat lima tahapan perencanaan *project* (Belferik et al., 2023:14-23) yaitu:

- 1) Tahap inisiasi *project* adalah tahap pertama yang mengubah ide abstrak menjadi tujuan yang bermakna. Pada tahap ini harus merincikan informasi seperti ruang lingkup *project*, tujuan, *project manager*, anggaran, jadwal kerja, mengidentifikasi *stakeholder*, serta menentukan orang-orang yang terlibat dalam *project*.
- 2) Tahap perencanaan *project* adalah tahap menguraikan tugas dan garis waktu yang diperlukan untuk melaksanakan *project*. Selain itu berisi ukuran kinerja yang jelas terakini peran dan tugas setiap orang dalam *project* tersebut.
- 3) Tahap pelaksanaan *project* adalah tahap awal *project* mulai dikerjakan oleh tim. Pada tahap ini sangat penting memberikan tanggung jawab dari *project* tersebut kepada orang yang tepat dan mengevaluasi kinerjanya sehingga tujuan dari *project* tersebut dapat tercapai.
- 4) Tahap pemantauan dan pengendalian *project* adalah tahap yang berjalan dengan memantau pekerjaan yang berjalan dengan biaya yang dikeluarkan. Tahap pemantauan dan pengendalian ini berjalan secara bersamaan dengan pelaksanaan *project*.
- 5) Tahap penutupan *project* adalah tahap akhir, yang artinya *project* yang ditangi telah selesai dan disetujui dengan standar yang diinginkan. *Project* yang sudah selesai ini tetap akan dievaluasi secara keseluruhan.

Project pada bidang teknologi informasi adalah *project* pengembangan aplikasi. Perkembangan teknologi informasi telah meningkatkan kebutuhan terhadap informasi yang digital. Proses pengembangan aplikasi melibatkan programmer atau sekelompok *developer/programmer* dalam pembuatan serangkaian proses mulai dari perancangan awal, pembuatan *prototype*, implementasi, pengujian, hingga aplikasi siap digunakan.

Proses pengembangan aplikasi terdapat serangkaian tahap yang telah diatur untuk memastikan bahwa aplikasi dapat dibuat dengan struktur yang baik dan terencana. Proses tersebut melibatkan pengumpulan data, perencanaan, pengembangan, dan pemeliharaan. Tahap ini merupakan tahap yang penting untuk *developer* dalam membangun aplikasi. Dalam praktiknya, terdapat berbagai metode pengembangan aplikasi yang sering digunakan oleh para *developer* seperti metode *waterfall*, metode *spiral*, metode XP (*Extreme Programming*), metode *kanban*, pengembangan *agile*, metode *prototype*, dan metode *scrum* (Pressman & Maxim, 2020:34-35).

5. *Unified Modeling Language*

UML adalah bahasa visual yang serba guna untuk menjelaskan, memvisualisasikan. UML berguna dalam memahami, merancang, mengonfigurasi, merawat, dan mengelola informasi mengenai suatu sistem. Bahasa ini dibuat dengan tujuan dapat dimodifikasi, diperluas, dan komprehensif, tetapi juga sederhana sehingga dapat digunakan sebagai

dasar untuk semua kebutuhan pemodelan sistem. Dengan karakteristiknya, berbagai elemen yang ditandai oleh tipe diagram yang ditangani dan atribut yang disediakan (Sumirat et al., 2023: 73).

Unified Modeling Language (UML) merupakan alat yang digunakan dalam pengembangan sistem di bidang rekayasa perangkat lunak. UML berfungsi sebagai bahasa visual untuk mendefinisikan dan mendokumentasikan sistem. Skenario yang menggambarkan cara pengguna berinteraksi dengan sistem dijelaskan melalui UML (Koç et al., 2021).

UML adalah kumpulan diagram, struktur, dan teknik yang digunakan dalam pemodelan dan perancangan program serta aplikasi yang berorientasi objek (Kroenke et al., 2018: 220).





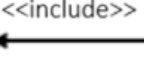
Berdasarkan beberapa pengertian di atas, dapat diartikan bahwa, *Unified Modeling Language* (UML) adalah bahasa visual yang serbaguna dan komprehensif, dirancang untuk mendukung berbagai aspek pengembangan sistem, mulai dari pemahaman hingga pengelolaan informasi sistem. UML dapat disesuaikan dan diperluas, serta berfungsi sebagai alat dasar dalam semua kebutuhan pemodelan sistem. Sebagai bahasa visual, UML memfasilitasi pendefinisian dan pendokumentasian sistem, termasuk mendetailkan interaksi pengguna. UML juga mencakup berbagai diagram, struktur, dan teknik yang digunakan dalam pemodelan serta perancangan program dan aplikasi berorientasi objek.


a. *Use Case Diagram*

Merupakan salah satu jenis bentuk dari UML diagram yang menggambarkan hubungan interaktif antara sistem dan aktor. *Use case* adalah metode untuk menggambarkan bagaimana *user* berinteraksi dengan sistem programnya sendiri. Skenarionya adalah ketika sebuah sistem digunakan melalui sebuah narasi. Penggunaan *use case* juga digunakan untuk membentuk perilaku sistem yang akan dikembangkan. Sebuah skenario penggunaan menunjukkan interaksi antara pengguna (aktor) dan sistem yang ada. Diagram ini menggambarkan aktivitas dan operasi bisnis yang dilakukan oleh pengguna (aktor). Dalam *Use Case diagram* ini, setiap aktivitas direpresentasikan dalam sebuah *oval Use Case* dengan hubungan-hubungan internalnya (Ahmad et al., 2022: 70).

Komponen atau simbol diagram *use case* (Sumirat et al., 2023:82):

Tabel 2. 1
Simbol *use case diagram*

Simbol	Nama	Keterangan
	Aktor	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use Case</i>	Abstraksi dan interaksi antar sistem dan aktor
	Association	Abstraksi dari penghubung antara actor dengan <i>use case</i>
	Generalisasi	Memperlihatkan spesialisasi aktor untuk bisa berperan dengan <i>use case</i>
	Include	Memperlihatkan bahwa suatu <i>use case</i> seluruhnya adalah fungsionalitas dari <i>use case</i> lainnya



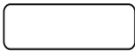
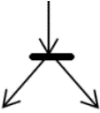

	Extend	Memperlihatkan bahwa suatu <i>use case</i> adalah tambahan fungsional dari <i>use case</i> lainnya jika satu keadaan terpenuhi
---	--------	--


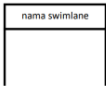
Sumber: Dokumen Pribadi

b. Activity Diagram

Activity Diagram atau diagram aktivitas adalah alat yang digunakan untuk merepresentasikan perilaku (aktivitas) dalam proses bisnis (sistem). Diagram aktivitas dapat dipakai untuk memodelkan berbagai macam hal alur kerja bisnis dengan banyak *use case* yang berbeda. Singkatnya, diagram aktivitas berguna untuk memodelkan berbagai jenis proses-proses yang terjadi pada sebuah sistem. Simbol-simbol yang digunakan dalam *Activity Diagram* terlihat pada tabel berikut:

Tabel 2. 2
Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i>	Akhir aktivitas
	<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
	<i>Fork</i> (Percabangan)	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (Penggabungan)	Digunakan untuk menunjukkan adanya dekomposisi




	<i>Decision Points</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>True</i> dan <i>False</i>
	<i>Swimlane</i>	Pembagian <i>Activity Diagram</i> untuk menunjukkan siapa melakukan apa

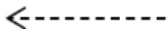
Sumber: Dokumen Pribadi

c. *Sequence Diagram*

Sequence Diagram adalah diagram yang menggambarkan dan mendeskripsikan urutan aktivitas *object* yang ada pada *use case* dan urutan pesan eksplisit yang disampaikan di antara *object* dalam interaksi yang ditentukan berdasarkan waktu pelaksanaannya. Simbol yang digunakan dalam penggambaran *Sequence Diagram* seperti yang terlihat pada tabel:

Tabel 2. 3
Simbol *Sequence Diagram*

Simbol	Nama Simbol	Keterangan
	Aktor	Sebuah objek yang berasal dari kelas, atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus - putus menunjukkan garis hidup suatu objek
	Aktivasi	Menunjukkan masa hidup dari subjek
Message 1 	Pesan	Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukan pada bagian operasi pada diagram kelas.

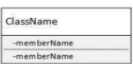




<p>Message 2</p> 	Return	Pesan kembalian dari komunikasi antar objek
--	--------	---

Sumber: Dokumen Pribadi

d. *Class Diagram*

Class diagram merupakan salah satu jenis diagram yang paling banyak digunakan dalam rekayasa perangkat lunak. Diagram ini menggambarkan entitas-entitas kunci dalam konteks bisnis maupun domain teknis. Class diagram memiliki sifat yang sangat struktural dan statis. Selain itu, class diagram mampu merepresentasikan kelas-kelas yang ada pada tingkat bisnis, serta kelas-kelas pada tingkat teknis yang dihasilkan dari bahasa pemrograman, seperti Java atau C++. Simbol-simbol yang digunakan dalam *class diagram* terlihat pada tabel berikut:

Tabel 2. 4
Simbol Class Diagram

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem
	Association	Relasi antar kelas dengan makna umum, biasanya juga disertai dengan <i>multiplicity</i>
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
	Agregasi	Relasi antar kelas dengan makna semua bagian (<i>whole part</i>)
	Multiplicity	Relasi antara dua kelas yang membawa informasi tentang jumlah objek yang diperhitungkan pada setiap akhir asosiasi

Sumber: Dokumen Pribadi

6. *MySQL*

Basis data adalah kumpulan catatan terintegrasi yang mendeskripsikan dirinya sendiri. Basis data terdiri dari tabel-tabel yang terintegrasi, di mana tabel terintegrasi menyimpan baik data maupun hubungan antara data. Sistem manajemen basis data (DBMS) adalah perangkat lunak yang mengatur dan memproses basis data serta aplikasinya. Sistem informasi melibatkan pengguna, aplikasi basis data, DBMS, dan basis data (Kroenke et al., 2018).

Basis data (*database*) merupakan kumpulan informasi penting untuk semua proses bisnis yang berlaku untuk jangka waktu yang lama. Perusahaan menyimpan setiap data-data penting ke dalam basis data. Basis data diwujudkan dalam bentuk perangkat lunak khusus yang disebut DBMS (*Database Management System*) atau disebut juga Sistem Basis Data. Beberapa contoh DBMS populer adalah *MySQL*, *PostgreSQL*, *MongoDB*, *Oracle Database*, dan *Microsoft SQL Server* (Putri, 2022:3-7).

MySQL adalah *software opensource* yang tersedia secara gratis berdasarkan ketentuan GNU *General Public License*. Selain itu, *MySQL* juga dapat diperoleh melalui berbagai jenis lisensi *proprietary*. *MySQL* mengelola data dalam satu atau lebih tabel, di mana setiap jenis data memiliki koneksi langsung satu sama lain. Hubungan ini memfasilitasi pengaturan data secara efisien. *SQL (Structured Query Language)* adalah bahasa pemrograman yang digunakan untuk membuat, memodifikasi, dan mengambil data dari database relasional, serta mengatur akses pengguna ke

database tersebut. Selain *database* relasional dan *SQL*, RDBMS seperti *MySQL* bekerja sama dengan sistem operasi untuk menjalankan *database* relasional di komputer (Silalahi, 2022:1-2).

Basis data adalah kumpulan catatan yang saling terintegrasi dan mendeskripsikan dirinya sendiri, terdiri dari tabel-tabel yang menyimpan data serta hubungan antar data. Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang mengatur dan memproses basis data serta aplikasinya, dan termasuk dalam sistem informasi bersama pengguna dan aplikasi basis data. Basis data menyimpan informasi penting untuk jangka panjang dalam bisnis, dan DBMS populer seperti *MySQL*, *PostgreSQL*, dan *Oracle Database* digunakan untuk mengelola data. *MySQL* adalah perangkat lunak open-source yang mengelola data dalam tabel yang saling terhubung dan menggunakan *SQL* untuk manipulasi dan pengambilan data, serta berintegrasi dengan sistem operasi untuk menjalankan *database*.

7. Bahasa Pemrograman Java

Bahasa pemrograman, sering disebut juga sebagai bahasa komputer atau bahasa pemrograman komputer, adalah sekumpulan instruksi standar yang digunakan untuk mengendalikan komputer. Bahasa ini terdiri dari aturan sintaks dan semantik yang mendefinisikan bagaimana program komputer dibuat. Dengan bahasa pemrograman, seorang programmer dapat menentukan dengan jelas data yang akan diproses oleh komputer, cara penyimpanan dan transfer data tersebut, serta langkah-langkah yang harus diambil dalam berbagai situasi (Musfikar et al., 2023: 2).

Java adalah bahasa pemrograman yang diciptakan oleh James Gosling dan pertama kali diluncurkan versi 1.0 nya pada 23 Mei 1995 oleh *Sun Microsystems*. *Java* menyempurnakan konsep dari bahasa *C++*. Filosofi *Java* “*Write Once, Run Anywhere*” menjadikan bahasa ini ideal untuk mendistribusikan kode melalui Internet. *Java* adalah bahasa pemrograman berorientasi object yang tersusun dari *Class* dan *Object*. *Java* merupakan bahasa pemrograman *multi-platform*, yang dapat berjalan pada semua jenis sistem operasi yang sudah terpasang JVM (*Java Virtual Machine*). *Java* dapat digunakan untuk mengembangkan berbagai jenis aplikasi seperti *mobile apps*, *web apps*, *desktop*, dan *game* (Martinez et al., 2023).

Bahasa pemrograman *JAVA* adalah bahasa yang khas. Selama periode pesatnya perkembangan World Wide Web dan Internet, bahasa *Java* telah mengalami perkembangan yang signifikan. Dari versi awalnya hingga bentuknya yang matang saat ini, bahasa ini telah mengalami perubahan besar. Kesederhanaan dan fitur-fitur unik *Java* yang tidak ada di bahasa lain telah menjadikannya populer. Pengembang perangkat lunak cenderung memilih bahasa ini karena banyak keuntungan yang ditawarkannya. Dengan demikian, penggunaan dan penerapan bahasa pemrograman *Java* akan semakin meluas (Jiang, 2020: 5).

B. Penelitian yang Relevan

Berikut ini beberapa hasil penelitian yang relevan dan dapat dijadikan bahan telaah:

Tabel 2. 5
Penelitian yang relevan

No.	Peneliti (Tahun)	Judul Penelitian	Metode Penelitian	Hasil Penelitian
1.	Nuriana Santiara (2017)	Penerapan Metode SAW pada Manajemen <i>Project</i> Untuk Menentukan Pegawai dalam Pengerjaan suatu <i>Project</i> (Studi Kasus PT. Deliman Integra Teknologi)	Simple Additive Weighting (SAW)	Telah berhasil dirancang sistem yang dapat membantu pimpinan perusahaan untuk memilih pegawai dalam penentuan pengerjaan <i>project</i> serta pimpinan dapat memulai, merencanakan, melaksanakan, memantau dan mengendalikan suatu <i>project</i> . Lalu klien juga dapat memantau <i>project</i> nya yang sedang dikerjakan. (Santiara, 2017)
2.	M. Henky Saputra (2019)	Sistem Pendukung Keputusan Penentuan Prioritas Penyelesaian <i>Project</i> dengan Metode Simple Additive Weighting (SAW) di PT. Nindya Karya (Persero) Kanwil 1 Palembang	Simple Additive Weighting (SAW)	Telah berhasil dirancang sistem pendukung keputusan dengan metode SAW yang dapat membantu Manajer Estimasi dalam mengambil keputusan untuk memilih <i>project</i> mana yang layak dan dapat dipercepat penyelesaiannya sebelum batas waktu yang ada di dalam kontrak kerja. (M. H. Saputra, 2019)
3.	Muhammad Andryan Wahyu Saputra, Muh Nur Aslam, Muhammad Ridha, Muhammad Ainul Yaqin (2021)	Sistem Pendukung Keputusan Untuk Menentukan Manajer Proyek Menggunakan Metode <i>Simple Additive Weighting</i> (SAW)	Simple Additive Weighting (SAW)	Telah berhasil dirancang sistem pendukung keputusan menggunakan metode SAW untuk menentukan manajer proyek. Metode SAW dapat diimplementasikan terhadap sistem pendukung keputusan dalam menentukan manajer proyek dengan baik. Penilaian penentuan manajer proyek dapat ditentukan menggunakan metode SAW dengan kriteria utama yang digunakan dalam sistem ini adalah kebutuhan dasar, keterampilan manajemen, keterampilan manajemen proyek, kemampuan interpersonal yang

				didalamnya terdapat beberapa sub kriteria. (M. A. W. Saputra et al., 2021)
4.	Kevin Gautama, dan Suwarno (2021)	Analisis dan Perancangan Sistem Pendukung Keputusan Untuk Menentukan Prioritas Pengerjaan Aplikasi Menggunakan Metode <i>Simple Additive Weighting</i>	Simple Additive Weighting (SAW)	Telah berhasil dirancang sistem pendukung keputusan untuk menentukan prioritas pengerjaan aplikasi menggunakan metode SAW. Penilaian yang digunakan pada sistem adalah risiko, pengeluaran, penggunaan sumber daya, lama waktu penyelesaian dan keuntungan dari penyelesaian tugas. (Gautama & Suwarno, 2021)

Sumber: Dokumen Pribadi

BAB III METODE PENELITIAN

A. Waktu dan Tempat Penelitian

1. Waktu Penelitian

Waktu yang dibutuhkan dalam melakukan penelitian yaitu selama 4 (empat) bulan, mulai dari bulan Mei 2024 sampai dengan bulan Agustus 2024, dengan perincian jadwal sebagai berikut:

Tabel 3. 1
Waktu penelitian

No.	Kegiatan	Mei				Juni				Juli				Agustus		
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1	Mengumpulkan data															
2	Studi Kepustakaan															
3	Analisis Kebutuhan Sistem															
4	Perancangan Sistem															
5	Implementasi Pengembangan Sistem															
6	Pengujian Sistem															

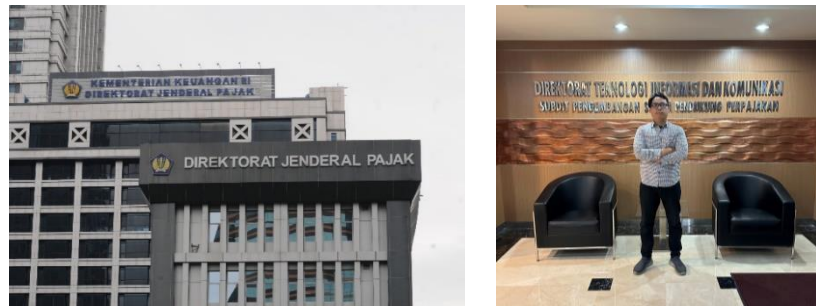
Sumber: Dokumen Pribadi

2. Tempat Penelitian

Tempat penelitian dalam proses pengumpulan data yang menunjang proses penyusunan skripsi ini dilakukan pada:

Lokasi : Direktorat TIK, Direktorat Jenderal Pajak

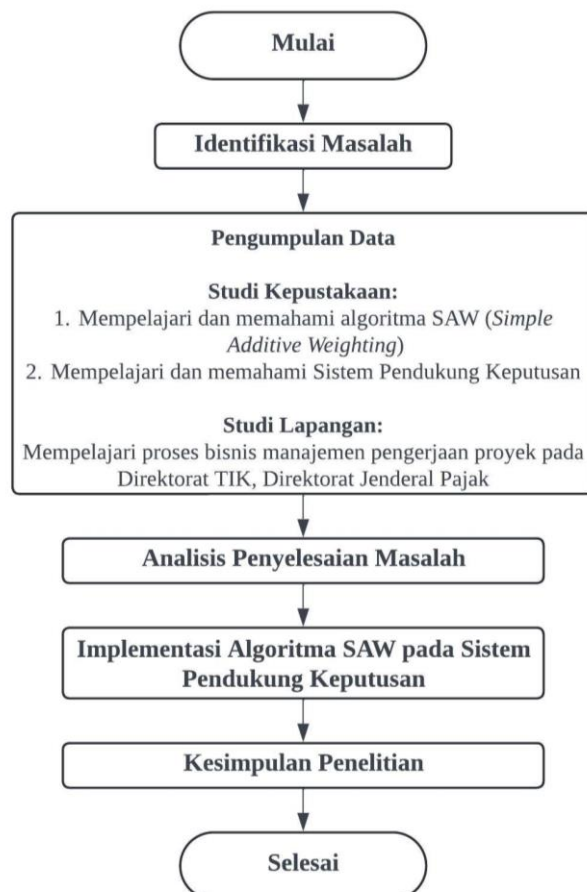
Alamat : Kav. 40-42, Jl. Gatot Subroto No.7, RT.7/RW.1, Senayan,
Kec. Kby. Baru, Kota Jakarta Selatan, DKI Jakarta, 12190



Gambar 3. 1
Tempat penelitian
Sumber: Dokumen Pribadi

B. Tahapan Penelitian

Penelitian ini dilakukan dalam beberapa tahapan, berikut *workflow* dari tahapan penelitian ini:



Gambar 3. 2
Tahapan penelitian
Sumber: Dokumen Pribadi

Berdasarkan *workflow* tahapan penelitian di atas, berikut penjelasan dari masing-masing proses:

1. Identifikasi Masalah

Pada tahap ini, penulis melakukan identifikasi masalah pada manajemen *project* di Direktorat TIK, Direktorat Jenderal Pajak. Direktorat TIK sudah memiliki aplikasi untuk melakukan manajemen *project*, mulai dari permintaan aplikasi dari Buisness Owner ke Direktorat TIK, proses analisis bisnis dan teknis, sampai pengembangan dan *deployment* aplikasi yang dilakukan *Developer*. Pada alur manajemen project tersebut penulis menemukan banyaknya permintaan aplikasi dari Buisness Owner internal maupun eksternal perusahaan, sehingga berdampak pada beban kerja dari para Pengembang (*Developer*) Aplikasi yang semakin banyak dan kurang merata yang mengakibatkan kurang optimalnya penyelesaian dari *project-project* tersebut. Penulis menemukan adanya kebutuhan *tools* berupa sistem yang dapat melakukan pengambilan keputusan dalam menentukan penugasan pengerjaan project aplikasi pada Direktorat TIK, Direktorat Jenderal Pajak.

2. Pengumpulan Data

Metode yang dilakukan penulis dalam pengumpulan data adalah:

a. Studi Kepustakaan

Penulis melakukan studi kepustakaan sebagai berikut:

- 1) Mempelajari buku-buku dan jurnal terkait Algoritma SAW (*Simple Additive Weighting*)

- 2) Mempelajari buku-buku dan jurnal terkait Sistem Pendukung Keputusan
- 3) Mempelajari aplikasi sistem pendukung keputusan dengan algoritma yang sama dengan aplikasi yang akan dibuat

b. Studi Lapangan

Penulis mengumpulkan data dan informasi dengan melakukan studi lapangan mengenai bagaimana proses bisnis pada manajemen pengerjaan project di Direktorat TIK. Adapun teknik yang digunakan penulis dalam mengumpulkan data dan informasi pada studi lapangan, yaitu:

1) Observasi

Penulis melakukan observasi dengan mengamati dan mempelajari proses bisnis manajemen pengerjaan *project* pada Direktorat TIK.

2) Wawancara

Penulis melakukan wawancara kepada Kepala Seksi, Sistem Analis dan Programmer pada Direktorat TIK mengenai alur pengerjaan project, menentukan *Developer* yang akan mengerjakan project pengembangan aplikasi, hingga kriteria yang digunakan dalam menentukan *Developer* tersebut.

3. Analisis Penyelesaian Masalah

Dengan mengidentifikasi masalah dan mengumpulkan data serta informasi dari berbagai sumber, penulis melakukan analisis dalam penyelesaian masalah tersebut. Adapun hasil dari analisis untuk

menyelesaikan masalah tersebut adalah dengan membangun sebuah aplikasi sistem pendukung keputusan menggunakan Algoritma SAW (*Simple Additive Weighting*) yang mampu merekomendasikan *Developer* yang tepat dalam pengerjaan project, yang diharapkan dapat membantu dalam mengambil keputusan untuk menentukan penugasan kepada pengembang (*Developer*) pada manajemen project di Direktorat TIK, sehingga beban kerja dari para Pengembang (*Developer*) Aplikasi menjadi merata dan mengoptimalkan penyelesaian dari *project-project* tersebut.

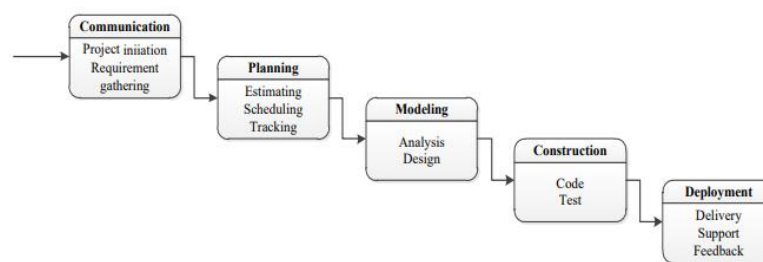
Alasan penulis memutuskan untuk menggunakan algoritma SAW karena, dari hasil wawancara penulis menemukan adanya kriteria-kriteria tertentu yang digunakan manajemen *project* pada Direktorat TIK dalam menugaskan para *developer*. Metode SAW merupakan salah satu metode pengambilan keputusan dalam menghadapi situasi *Multi-Attribute Decision Making* (MADM) yang digunakan untuk mengevaluasi masalah pengambilan keputusan sebagai model penyeleksi untuk memilih alternatif terbaik dari sejumlah alternatif dengan kriteria tertentu (Taherdoost, 2023:21).

Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Menurut Kusumadewi dalam Rahmansyah et al., 2021:39 konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua kriteria. Metode SAW biasanya digunakan untuk menyelesaikan masalah penyeleksian dalam sebuah sistem pendukung keputusan yang memiliki banyak atribut (Noval et al.,

2020:117). Metode SAW memiliki beberapa kelebihan yaitu memiliki kemampuan untuk menyeimbangkan antar kriteria, intuitif bagi pengambil keputusan, perhitungan yang sederhana, tidak diperlukan pemrograman yang rumit, menggunakan nilai yang dinormalisasi untuk membantu menentukan perbedaaan secara visual antara objek yang dibandingkan.

4. Implementasi Algoritma SAW pada Sistem Pendukung Keputusan

Dalam implementasi Algoritma SAW pada sistem pendukung keputusan, penulis menerapkan metodologi pengembangan sistem dengan *Waterfall*, dengan tahapan pada gambar berikut:



Gambar 3. 3
Metode *Waterfall*
Sumber: Pressman

Berikut ini adalah penjelasan dari tahapan-tahapan yang terdapat dalam metode *Waterfall*:

- a. *Communication* merupakan tahap awal yang dilakukan dengan melakukan komunikasi kepada *Business Owner* dalam penelitian ini adalah Direktorat TIK, Direktorat Jenderal Pajak. Tahap ini merupakan yang paling penting karena merupakan tahap pengumpulan informasi tentang kebutuhan fitur pada aplikasi yang akan dibangun.

- b. *Planning*, pada tahap ini setelah mengumpulkan informasi dilakukan perencanaan pada proses pengerjaan pengembangan aplikasi yang meliputi tugas-tugas teknis, sumber daya yang dibutuhkan, timeline pengerjaan, dan target pengerjaan.
- c. *Modelling*, sebelum dilakukan pengerjaan pengembangan aplikasi (coding) perlu adanya tahap menerjemahkan informasi dari alur proses yang sudah dikomunikasikan pada tahap sebelumnya ke dalam bentuk perancangan sistem yang bertujuan untuk mempermudah dalam proses *coding*. Pada tahap ini berfokus pada perancangan struktur database, arsitektur aplikasi, *User Interface*, dan detail algoritma prosedural.
- d. *Construction*, adalah tahap penerjemahan modelling yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan (*coding*). Tahap ini merupakan tahapan yang nyata dari pengembangan aplikasi. Setelah proses *coding* selesai dilakukan, perlu adanya proses testing terhadap aplikasi yang sudah dibangun. Tujuan dari proses testing adalah menemukan error dan bugs dari aplikasi untuk diperbaiki, sehingga kualitas aplikasi menjadi lebih baik.
- e. *Deployment*, merupakan tahap pemasangan aplikasi pada suatu server agar aplikasi yang sudah dibangun dapat digunakan oleh *user*. Setelah aplikasi ter-*deploy* dan digunakan oleh *user*, perlu adanya pemeliharaan secara berkala, yaitu mengoreksi kesalahan pada aplikasi yang baru diketahui pada saat aplikasi dipergunakan yang bertujuan untuk

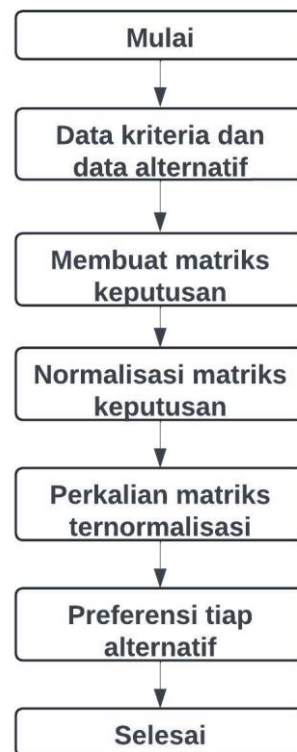
meningkatkan kualitas layanan dan performa dari aplikasi yang sudah dibangun.

5. Kesimpulan Penelitian

Penulis dapat menarik kesimpulan dari hasil analisa pada identifikasi masalah, bahwa solusi dari masalah yang ditemukan penulis adalah dengan membangun suatu aplikasi Sistem Pendukung Keputusan yang dapat membantu manajemen *project* untuk menentukan penugasan dalam pengerjaan *project* aplikasi di Direktorat TIK, Direktorat Jenderal Pajak.

C. Algoritma

Metode SAW merupakan salah satu metode pengambilan keputusan dalam menghadapi situasi *Multi-Attribute Decision Making* (MADM) yang digunakan untuk mengevaluasi masalah pengambilan keputusan sebagai model penyeleksi untuk memilih alternatif terbaik dari sejumlah alternatif dengan kriteria tertentu (Taherdoost, 2023:21). Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Metode SAW terdiri dari dua atribut yaitu kriteria keuntungan (*benefit*) dan kriteria biaya (*cost*). Perbedaan antara dua kriteria ini adalah dalam pemilihan kriteria ketika mengambil keputusan. Adapun kerangka kerja algoritma metode SAW seperti Gambar 3.4:



Gambar 3. 4
Kerangka Kerja Metode SAW
Sumber: Dokumen Pribadi

Penjelasan penyelesaian dalam menggunakannya pada penelitian ini adalah:

1. Menentukan alternatif (A_i): Nama pegawai berjumlah 6 orang sebagai sampel penelitian.
2. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan (C_i). Kriteria dalam penelitian ini adalah:
 - a. Kemampuan *coding* (C_1): kemampuan dari pegawai/ *developer* dalam menggunakan bahasa pemrograman untuk membangun suatu aplikasi.

- b. *Project* yang sedang dikerjakan (C2): jumlah *project* aplikasi yang sedang dikembangkan oleh pegawai/ *developer*.
 - c. Pengalaman (C3): jumlah *project* aplikasi yang sudah dikembangkan sampai tahap *deployment* oleh pegawai/ *developer*.
 - d. Pelatihan (C4): jumlah pelatihan IT yang sudah diikuti oleh pegawai/ *developer*.
 - e. Penguasaan *stack* (C5): jumlah teknologi yang dikuasai oleh pegawai/ *developer* seperti *framework* (*Springboot*, *Laravel*, dll), jenis *database* (SQL dan noSQL), *others* (*DevOps*, *Microservice Architecture*, dll) disesuaikan dengan teknologi yang dibutuhkan pada pengerjaan *project* aplikasi.
3. Menentukan nilai rating kecocokan setiap alternatif pada setiap kriteria.
Nilai rating pada penelitian ini adalah skala 1 (rendah) sampai dengan 5 (sangat tinggi)
 4. Menentukan bobot preferensi atau tingkat kepentingan (W) setiap kriteria.
Bobot preferensi setiap kriteria dalam penelitian ini adalah:
 - a. $C1 \rightarrow 0,35$ (35%)
 - b. $C2 \rightarrow 0,25$ (25%)
 - c. $C3 \rightarrow 0,15$ (15%)
 - d. $C4 \rightarrow 0,05$ (5%)
 - e. $C5 \rightarrow 0,2$ (20%)
 5. Membuat tabel rating kecocokan dari setiap alternatif pada setiap kriteria
 6. Membuat matrik keputusan berdasarkan kriteria.

7. Melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut keuntungan (max) dan atribut biaya (min) dengan formula:

$$\left\{ \frac{X_{ij}}{\text{Max}_i X_{ij}} \right\} = r_{ij} \text{ (Jika adalah atribut keuntungan (benefit))}$$

$$\left\{ \frac{\text{Min}_i X_{ij}}{X_{ij}} \right\} = r_{ij} \text{ (Jika adalah atribut biaya (cost))}$$

Keterangan :

r_{ij}	=	Nilai Rating Kinerja Ternormalisasi
X_{ij}	=	Nilai atribut yang dimiliki dari setiap kriteria
$\text{Max } X_{ij}$	=	Nilai terbesar dari kriteria
$\text{Min } X_{ij}$	=	Nilai terkecil dari kriteria
<i>Benefit</i>	=	Jika nilai terbesar adalah terbaik
<i>Cost</i>	=	Jika nilai terkecil adalah terbaik

8. Hasil dari nilai rating kinerja ternormalisasi (r_{ij}) membentuk matriks ternormalisasi

$$r_{ij} \left\{ \begin{array}{l} \frac{X_{ij}}{\text{Max}_i X_{ij}} \\ \frac{\text{Min}_i X_{ij}}{X_{ij}} \end{array} \right.$$

9. Hasil akhir nilai preferensi (V_i) diperoleh dari penjumlahan dari perkalian elemen ternormalisasi (R) dengan bobot (W) yang bersesuaian elemen kolom matrik (W).

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

Keterangan :

V_i = Ranking untuk setiap alternatif

W_j = Nilai bobot dari setiap kriteria

r_{ij} = Nilai Rating kinerja ternormalisasi

Hasil perhitungan nilai V_i yang lebih besar mengindikasikan bahwa alternatif tertentu merupakan alternatif terbaik.

BAB IV

HASIL DAN PEMBAHASAN

A. Definisi Masalah

Direktorat TIK Direktorat Jenderal Pajak memiliki manajemen *project* pengembangan aplikasi yang sudah baik. Semua permintaan pengembangan aplikasi dari *business owner* yang berasal dari internal maupun eksternal perusahaan, melewati serangkaian prosedur sebelum diserahkan kepada tim *developer*. *Project* tersebut kemudian diserahkan kepada subdirektorat terkait yang ada di Direktorat TIK. Namun, *developer* yang ditugaskan untuk mengerjakan *project* tersebut dipilih secara subjektif dan langsung oleh atasan. Hal tersebut mengakibatkan para *developer* memiliki beban kerja yang banyak dan tidak merata. Tidak adanya dokumentasi yang jelas terkait jumlah *project* yang sedang dikerjakan juga mengakibatkan beberapa *developer* memiliki tugas yang banyak, dan beberapa *developer* hanya mengerjakan tugas yang sedikit, sehingga penyelesaian *project* lebih lama. Oleh karena itu dibutuhkan Sistem Pendukung Keputusan untuk menentukan penugasan pengerjaan *project* pengembangan aplikasi di Direktorat TIK untuk membantu memudahkan Direktorat TIK dalam menentukan *developer* yang akan mengerjakan *project* yang masuk. Adanya sistem yang dapat membantu menentukan penugasan *project* ini juga diharapkan dapat meningkatkan kualitas pengambilan keputusan dan tepat sasaran karena dianalisis secara sistematis.

B. Pembahasan Algoritma

Pembangunan Sistem Pendukung Keputusan untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, Direktorat Jenderal Pajak menggunakan metode SAW (*Simple Additive Weighting*). Berikut adalah perhitungan manual dari penerapan algoritma Sistem Pendukung Keputusan dengan metode SAW:

1. Penentuan kriteria dan rating kepentingan yang digunakan dalam pengolahan data

Tabel 4. 1
Penentuan Kriteria

Kriteria	Keterangan
C1	Kemampuan <i>coding</i> (<i>benefit</i>)
C2	<i>Project</i> yang sedang dikerjakan (<i>cost</i>)
C3	Pengalaman (<i>benefit</i>)
C4	Pelatihan (<i>benefit</i>)
C5	Penguasaan <i>stack</i> (<i>benefit</i>)

Sumber: Dokumen Pribadi

Tabel 4. 2
Rating Kepentingan

Rating Kepentingan	Bobot
Sangat rendah	1
Rendah	2
Sedang	3
Tinggi	4
Sangat Tinggi	5

Sumber: Dokumen Pribadi

2. Penentuan Bobot Preferensi berdasarkan kriteria

Tabel 4. 3
Bobot Kriteria Kemampuan Coding

Kemampuan Coding (C1)	Bobot
<i>Novice</i>	1
<i>Advance</i>	2
<i>Competent</i>	3
<i>Proficient</i>	4
<i>Expert</i>	5

Sumber: Dokumen Pribadi

Tabel 4. 4
Bobot Kriteria *Project* yang Sedang Dikerjakan

<i>Project</i> yang sedang dikerjakan	Bobot
Tidak ada <i>project</i>	1
Hanya satu <i>project</i>	2
Dua <i>project</i>	3
Tiga <i>project</i>	4
Empat atau lebih <i>project</i>	5

Sumber: Dokumen Pribadi

Tabel 4. 5
Bobot Kriteria Pengalaman

Pengalaman	Bobot
Tidak pernah mengerjakan <i>project</i>	1
Pernah mengerjakan satu <i>project</i>	2
Pernah mengerjakan dua <i>project</i>	3
Pernah mengerjakan tiga <i>project</i>	4
Pernah mengerjakan empat atau lebih <i>project</i>	5

Sumber: Dokumen Pribadi

Tabel 4. 6
Bobot Kriteria Pelatihan

Pelatihan	Bobot
Tidak pernah atau hanya sekali mengikuti pelatihan	1
Pernah mengikuti 2 sampai dengan 3 pelatihan	2
Pernah mengikuti 4 sampai dengan 5 pelatihan	3
Pernah mengikuti 6 sampai dengan 7 pelatihan	4
Pernah mengikuti 8 atau lebih pelatihan	5

Sumber: Dokumen Pribadi

Tabel 4. 7
Bobot Kriteria Penguasaan Stack

Penguasaan <i>Stack</i>	Bobot
0 - 1 <i>stack</i> yang sesuai dengan <i>requirement project</i>	1
2 <i>stack</i> yang sesuai dengan <i>requirement project</i>	2
3 <i>stack</i> yang sesuai dengan <i>requirement project</i>	3
4 <i>stack</i> yang sesuai dengan <i>requirement project</i>	4
> 4 <i>stack</i> yang sesuai dengan <i>requirement project</i>	5

Sumber: Dokumen Pribadi

3. Pengisian Nilai Bobot pada masing-masing kriteria. Sebagai contoh pada bulan Juli 2024, enam orang pegawai akan dipilih untuk ditugaskan pengerjaan *project* yang masuk ke salah satu subdirektorat pada Direktorat TIK. Data pegawai tersebut dapat dilihat pada daftar kandidat (Tabel 4.8).

Tabel 4. 8
Daftar Kandidat

Nama Pegawai	Kemampuan coding	Project yang sedang dikerjakan	Pengalaman	Pelatihan	Penguasaan stack
Reza	<i>proficient</i>	0 <i>project</i>	3 <i>project</i>	2 pelatihan	3 <i>stack</i>
Putri	<i>proficient</i>	2 <i>project</i>	3 <i>project</i>	4 pelatihan	2 <i>stack</i>
Ahmad	<i>competent</i>	1 <i>project</i>	2 <i>project</i>	5 pelatihan	0 <i>stack</i>
Ayu	<i>advance</i>	0 <i>project</i>	0 <i>project</i>	4 pelatihan	2 <i>stack</i>
Bayu	<i>competent</i>	1 <i>project</i>	2 <i>project</i>	7 pelatihan	3 <i>stack</i>
Ade	<i>advance</i>	0 <i>project</i>	1 <i>project</i>	8 pelatihan	1 <i>stack</i>

Sumber: Dokumen Pribadi

4. Berdasarkan langkah penyelesaian dengan metode SAW, selanjutnya memberikan nilai setiap alternatif pada setiap kriteria yang sudah ditentukan.

Tabel 4. 9
Pembobotan setiap Alternatif

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
Reza	4	1	3	2	3
Putri	4	3	3	3	2
Ahmad	3	2	2	3	1
Ayu	2	1	1	3	2
Bayu	3	2	2	4	3
Ade	2	1	1	5	1

Sumber: Dokumen Pribadi

5. Selanjutnya proses penentuan bobot preferensi

Tabel 4. 10
Bobot Preferensi

Kriteria	Keterangan	Bobot Preferensi
C1	Kemampuan <i>Coding</i>	35% (0,35)
C2	<i>Project</i> yang sedang dikerjakan	25% (0,25)
C3	Pengalaman	15% (0,15)
C4	Pelatihan	5% (0,05)
C5	Penguasaan Stack	20% (0,2)
Total		100% (1)

Sumber: Dokumen Pribadi

6. Setelah itu maka dibuatkan matriks keputusan

$$M = \begin{Bmatrix} 4 & 1 & 3 & 2 & 3 \\ 4 & 3 & 3 & 3 & 2 \\ 3 & 2 & 2 & 3 & 1 \\ 2 & 1 & 1 & 3 & 2 \\ 3 & 2 & 2 & 4 & 3 \\ 2 & 1 & 1 & 5 & 1 \end{Bmatrix}$$

7. Tabel Normalisasi

Tabel 4. 11
Tabel Normalisasi

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
Reza	$\frac{4}{4} = 1$	$\frac{1}{1} = 1$	$\frac{3}{3} = 1$	$\frac{2}{5} = 0,4$	$\frac{3}{3} = 1$
Putri	$\frac{4}{4} = 1$	$\frac{1}{3} = 0,33$	$\frac{3}{3} = 1$	$\frac{3}{5} = 0,6$	$\frac{2}{3} = 0,67$
Ahmad	$\frac{3}{4} = 0,75$	$\frac{1}{2} = 0,5$	$\frac{2}{3} = 0,67$	$\frac{3}{5} = 0,6$	$\frac{1}{3} = 0,33$
Ayu	$\frac{2}{4} = 0,5$	$\frac{1}{1} = 1$	$\frac{1}{3} = 0,33$	$\frac{3}{5} = 0,6$	$\frac{2}{3} = 0,67$
Bayu	$\frac{3}{4} = 0,75$	$\frac{1}{2} = 0,5$	$\frac{2}{3} = 0,67$	$\frac{4}{5} = 0,8$	$\frac{3}{3} = 1$
Ade	$\frac{2}{4} = 0,5$	$\frac{1}{1} = 1$	$\frac{1}{3} = 0,33$	$\frac{5}{5} = 1$	$\frac{1}{3} = 0,3$

Sumber: Dokumen Pribadi

8. Tabel Perhitungan Bobot Perankingan

Tabel 4. 12
Perhitungan Bobot Perankingan

Alternatif	Kriteria				
	C1	C2	C3	C4	C5
Reza	$1 \times 0,35$ = 0,35	$1 \times 0,25$ = 0,25	$1 \times 0,15$ = 0,15	$0,4 \times 0,05$ = 0,02	$1 \times 0,2$ = 0,2
Putri	$1 \times 0,35$ = 0,35	$0,33 \times 0,25$ = 0,08	$1 \times 0,15$ = 0,15	$0,6 \times 0,05$ = 0,03	$0,67 \times 0,2$ = 0,13
Ahmad	$0,75 \times 0,35$ = 0,26	$0,5 \times 0,25$ = 0,13	$0,67 \times 0,15$ = 0,1	$0,6 \times 0,05$ = 0,03	$0,33 \times 0,2$ = 0,06
Ayu	$0,5 \times 0,35$ = 0,18	$1 \times 0,25$ = 0,25	$0,33 \times 0,15$ = 0,05	$0,6 \times 0,05$ = 0,03	$0,67 \times 0,2$ = 0,13
Bayu	$0,75 \times 0,35$ = 0,26	$0,5 \times 0,25$ = 0,13	$0,67 \times 0,15$ = 0,1	$0,8 \times 0,05$ = 0,04	$1 \times 0,2$ = 0,2
Ade	$0,5 \times 0,35$ = 0,18	$1 \times 0,25$ = 0,25	$0,33 \times 0,15$ = 0,05	$1 \times 0,05$ = 0,05	$0,33 \times 0,2$ = 0,06

Sumber: Dokumen Pribadi

Setelah mendapatkan hasil perkalian antara bobot ternormalisasi untuk masing-masing kriteria, selanjutnya menjumlahkan hasil perkalian tersebut sesuai dengan alternatifnya

Tabel 4. 13
Perhitungan Alternatif

Alternatif		Total
Reza	$0,35 + 0,25 + 0,15 + 0,02 + 0,2$	0,97
Putri	$0,35 + 0,08 + 0,15 + 0,03 + 0,13$	0,74
Ahmad	$0,26 + 0,13 + 0,1 + 0,03 + 0,06$	0,58
Ayu	$0,18 + 0,25 + 0,05 + 0,03 + 0,13$	0,64
Bayu	$0,26 + 0,13 + 0,1 + 0,04 + 0,2$	0,73
Ade	$0,18 + 0,25 + 0,05 + 0,05 + 0,06$	0,59

Sumber: Dokumen Pribadi

Hasil dari urutan perankingan

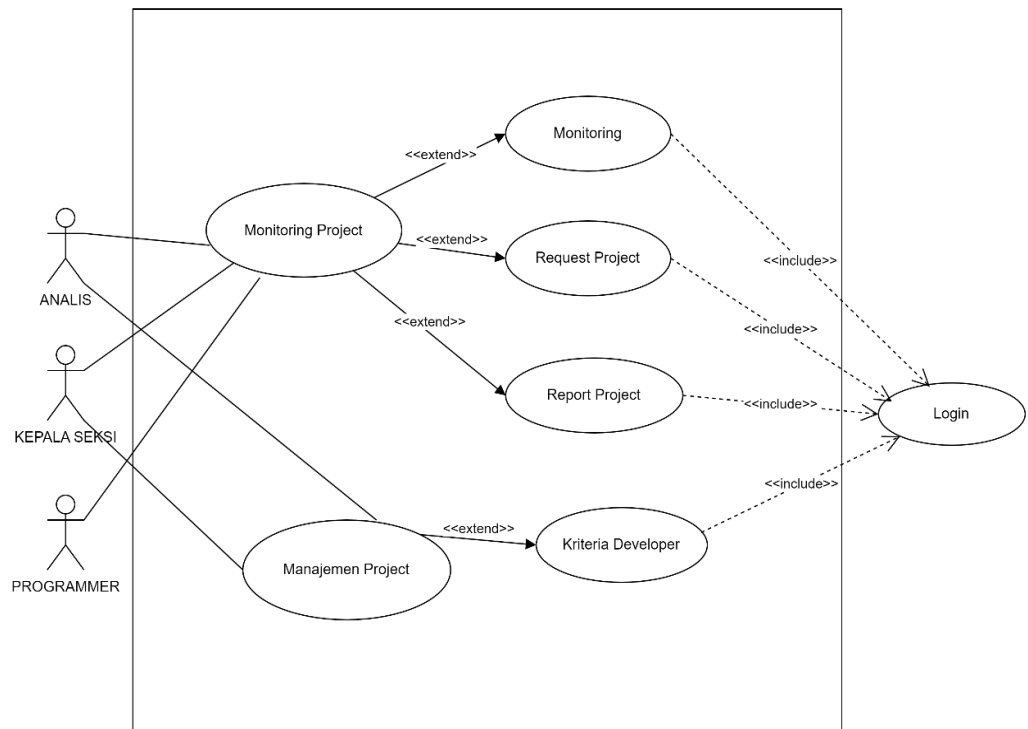
1	Reza
2	Putri
3	Bayu
4	Ayu
5	Ade
6	Ahmad

C. Pemodelan Perangkat Lunak

1. Pemodelan Perangkat Lunak dengan UML

a. *Use Case Diagram*

Berikut merupakan *use case* diagram (Gambar 4.1) dari Sistem Pendukung Keputusan untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, DJP.



Gambar 4. 1
Use Case Diagram
 Sumber: Dokumen Pribadi

Berikut ini adalah penjelasan mengenai *use case diagram* di atas:

Tabel 4. 14
 Skenario *Use Case Login*

Skenario <i>Use case Login</i>	
Tujuan	Melakukan proses <i>login</i> untuk dapat masuk ke dalam aplikasi
Aktor	Analisis, Kepala Seksi, Programmer
Kondisi Awal	Tampilan <i>form Login</i>
Skenario Utama	1) Sistem Menampilkan <i>form Login</i> 2) User mengisi <i>form Login</i> dengan memasukkan <i>username</i> dan <i>password</i>
Skenario Alternatif	1) Jika validasi berhasil, akan masuk ke halaman <i>Home</i> aplikasi 2) Jika validasi gagal, akan diminta untuk memasukkan ulang <i>username</i> dan <i>password</i> yang benar
Kondisi Akhir	Sistem akan masuk ke halaman <i>Home</i> aplikasi

Sumber: Dokumen Pribadi

Tabel 4. 15
Skenario *Use Case Monitoring Project*

Skenario <i>Use case Monitoring Project</i>	
Tujuan	Melihat <i>sub-menu</i> yang ada di dalam menu monitoring project, yaitu sub-menu monitoring dan request project.
Aktor	Analisis, Kepala Seksi, Programmer
Kondisi Awal	Tampilan halaman <i>Home</i> setelah berhasil <i>login</i> .
Skenario Utama	1) Sistem mengarahkan <i>user</i> ke halaman <i>Home</i> setelah berhasil masuk. 2) <i>User</i> memilih menu <i>Monitoring Project</i> 3) <i>User</i> memilih <i>sub-menu</i> yang tersedia.
Skenario Alternatif	1) Jika <i>user</i> memilih sub-menu monitoring, maka akan diarahkan ke halaman monitoring 2) Jika <i>user</i> memilih sub-menu request project, maka akan diarahkan ke halaman request project 3) Jika <i>user</i> memilih sub-menu report project maka akan diarahkan ke halaman report project.
Kondisi Akhir	Sistem akan mengarahkan ke halaman sub-menu yang dipilih oleh <i>user</i>

Sumber: Dokumen Pribadi

Tabel 4. 16
Skenario *Use Case Monitoring*

Skenario <i>Use case Monitoring</i>	
Tujuan	Untuk menampilkan seluruh data project yang ada dan memonitoring status proses pengerjaannya sudah sampai mana.
Aktor	Analisis, Kepala Seksi, Programmer
Kondisi Awal	<i>Datatable</i> daftar semua project
Skenario Utama	-
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan <i>datatable</i> daftar semua project.

Sumber: Dokumen Pribadi

Tabel 4. 17
Skenario *Use Case Request Project*

Skenario <i>Use case Request Project</i>	
Tujuan	<ul style="list-style-type: none"> – Analis: Untuk menambah, mengedit, menghapus, dan mengajukan pengembangan project yang baru – Kepala Seksi: Untuk melihat detail, membentuk tim berdasarkan hasil rekomendasi perhitungan metode SAW, mencetak daftar <i>programmer</i> hasil rekomendasi, dan sekaligus membuat surat penugasan tim atas pengembangan project yang diajukan sebelumnya oleh analis. – <i>Programmer</i>: Untuk melihat <i>project</i> yang ditugaskan, mencetak surat penugasan pengembangan project yang ditugaskan dan mengajukan ke tahap selanjutnya apabila sudah selesai proses pengerjaan (<i>coding</i>)
Aktor	Analis, Kepala Seksi, Programmer
Kondisi Awal	Datatable daftar project
Skenario Utama	<p>Analis:</p> <ol style="list-style-type: none"> 1) Analis dapat menambahkan, mengedit, dan menghapus data permintaan pengembangan project yang baru 2) Analis dapat mengajukan <i>project</i> yang baru tersebut kepada kepala seksi untuk dibentuk tim pengerjaannya. <p>Kepala Seksi:</p> <ol style="list-style-type: none"> 1) Kepala seksi dapat melihat detail project yang diajukan. 2) Kepala seksi dapat membentuk tim pengerjaan project tersebut dengan memilih daftar programmer yang direkomendasikan oleh sistem setelah diproses menggunakan metode SAW 3) Kepala seksi dapat mencetak hasil rekomendasi penugasan project. 4) Setelah memilih tim, kepala seksi dapat menetapkan dan sekaligus membuat surat penugasan tim atas project tersebut. <p>Programmer:</p> <ol style="list-style-type: none"> 1) Programmer dapat melihat detail project yang ditugaskan. 2) Programmer dapat melihat surat penugasan atas project yang sudah ditugaskan 3) Programmer dapat mengajukan ke tahap deployment apabila project sudah selesai dikerjakan.

Skenario Alternatif	-
Kondisi Akhir	<p>Analisis: Datatable seluruh project yang sudah diajukan beserta status pengerjaannya sudah sampai mana, dan melihat data detail dari setiap project.</p> <p>Kepala Seksi: Datatable yang sudah kosong karena daftar project sebelumnya sudah dibentuk tim pengerjaannya.</p> <p>Programmer: Datatable yang sudah kosong karena daftar project sebelumnya sudah selesai dikerjakan.</p>

Sumber: Dokumen Pribadi

Tabel 4. 18
Skenario *Use Case Report Project*

Skenario <i>Use case Report Project</i>	
Tujuan	<p>Analisis: Mencetak rekapitulasi pengerjaan <i>project</i> aplikasi.</p> <p>Kepala Seksi: Mencetak rekapitulasi pengerjaan <i>project</i> aplikasi.</p> <p>Programmer: Mencetak rekapitulasi <i>project</i> yang ditugaskan</p>
Aktor	Analisis, Kepala Seksi, Programmer
Kondisi Awal	Datatable daftar project
Skenario Utama	<p>Analisis: Analisis dapat mencetak rekapitulasi pengerjaan <i>project</i> aplikasi</p> <p>Kepala Seksi: Kepala Seksi dapat mencetak rekapitulasi pengerjaan <i>project</i> aplikasi</p> <p>Programmer: Programmer dapat mencetak rekapitulasi <i>project</i> yang ditugaskan</p>
Skenario Alternatif	-

Kondisi Akhir	Analis: Hasil cetakan data dalam bentuk pdf Kepala Seksi: Hasil cetakan data dalam bentuk pdf <i>Programmer</i> : Hasil cetakan data dalam bentuk pdf
---------------	---

Sumber: Dokumen Pribadi

Tabel 4. 19
Skenario *Use Case* Manajemen Project

Skenario <i>Use case</i> Manajemen Project	
Tujuan	Melihat <i>sub-menu</i> yang ada di dalam menu manajemen project, yaitu sub-menu kriteria developer dan report developer.
Aktor	Analis, Kepala Seksi
Kondisi Awal	Tampilan halaman <i>Home</i> setelah berhasil <i>login</i>
Skenario Utama	1) Sistem mengarahkan <i>user</i> ke halaman <i>Home</i> setelah berhasil masuk. 2) <i>User</i> memilih menu manajemen project 3) <i>User</i> memilih <i>sub-menu</i> kriteria developer
Skenario Alternatif	-
Kondisi Akhir	Sistem akan mengarahkan ke halaman sub-menu kriteria developer

Sumber: Dokumen Pribadi

Tabel 4. 20
Skenario *Use Case* Kriteria Developer

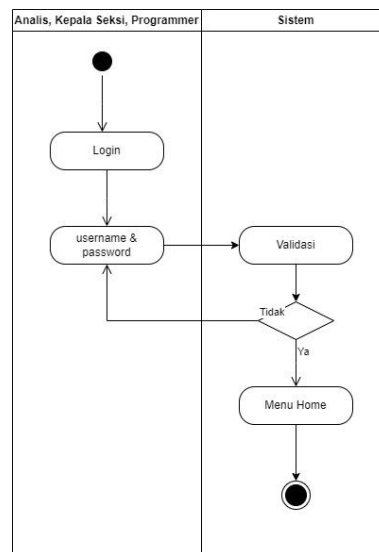
Skenario <i>Use case</i> Kriteria Developer	
Tujuan	Analis: Untuk menambah, melihat, mengedit, dan menghapus data kriteria penilaian <i>programmer</i> untuk dihitung menggunakan metode SAW Kepala Seksi: Untuk melihat detail data kriteria penilaian <i>programmer</i> .
Aktor	Analis, Kepala Seksi
Kondisi Awal	Datatable daftar <i>programmer</i>

Skenario Utama	Analis: 1) Analis dapat menambahkan , mengedit, melihat dan menghapus data kriteria penilaian <i>programmer</i> . Kepala Seksi: 1) Kepala seksi dapat melihat detail data kriteria penilaian <i>programmer</i> .
Skenario Alternatif	-
Kondisi Akhir	Datatable seluruh kriteria penilaian <i>programmer</i> .

Sumber: Dokumen Pribadi

b. *Activity Diagram*

1) Diagram *Activity* Login

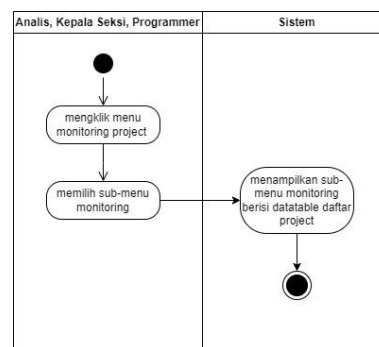


Gambar 4. 2
Diagram *Activity* Login
Sumber: Dokumen Pribadi

Berdasarkan gambar di atas diuraikan hak akses analis, kepala seksi, dan programmer terhadap sistem. Analis, kepala seksi dan programmer melakukan login terlebih dahulu yang kemudian

dilakukan validasi oleh sistem sesuai dengan data yang terdapat pada database. Jika data valid, maka pengguna akan menuju halaman Home. Pada halaman Home terdapat menu-menu yang bisa diakses oleh pengguna.

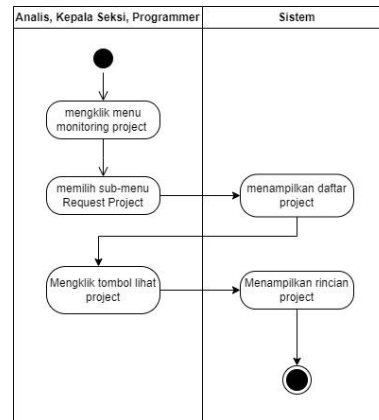
2) Diagram *Activity* sub-menu Monitoring



Gambar 4. 3
Diagram *Activity* sub-menu Monitoring
Sumber: Dokumen Pribadi

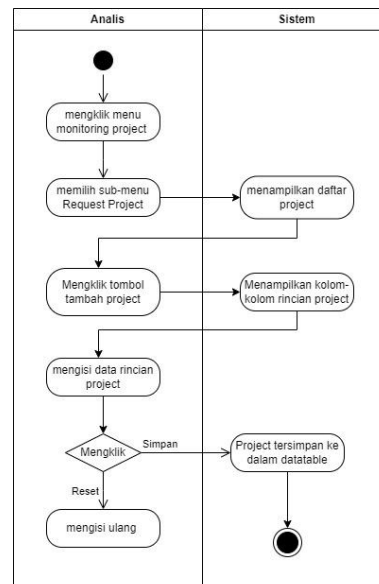
Berdasarkan gambar di atas diuraikan hak akses analis, kepala seksi, dan programmer terhadap sistem. Analis, kepala seksi, dan programmer memilih sub-menu monitoring yang ada pada menu utama Monitoring Project. Analis, kepala seksi, dan programmer dapat melihat daftar project berupa datatable yang berisi no, nama project, nama-nama tim pengembang, bisnis owner, versi pengembangan aplikasi, tanggal nota dinas, dan status pengerjaan project tersebut.

3) Diagram Activity sub-menu Request Project

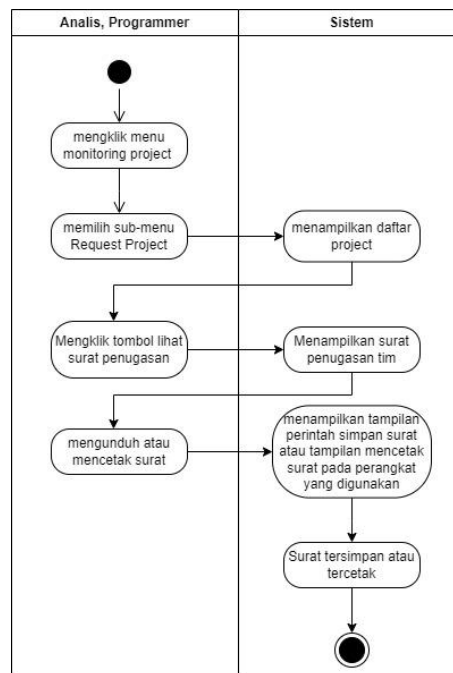


Gambar 4. 4
Diagram *Activity* Lihat data project
Sumber: Dokumen Pribadi

Berdasarkan gambar di atas diuraikan hak akses analis, kepala seksi, dan programmer terhadap sistem. Analis, kepala seksi dan programmer dapat melihat rincian project yang tersimpan dalam database.

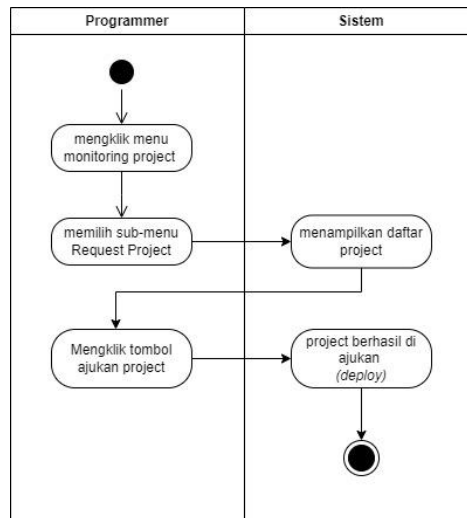


Gambar 4. 5
Diagram *Activity* Tambah Data Project
Sumber: Dokumen Pribadi



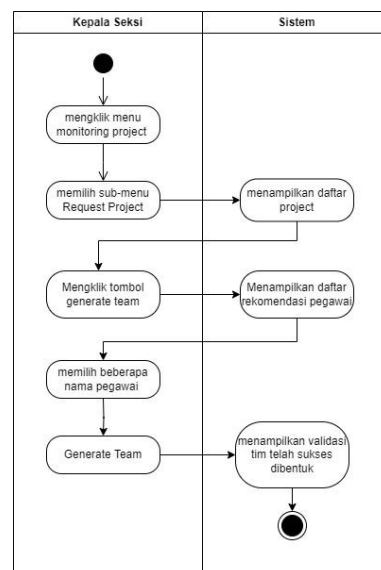
Gambar 4. 6
Diagram *Activity* Surat Penugasan Project
Sumber: Dokumen Pribadi

Berdasarkan gambar di atas (Gambar 4.5 dan Gambar 4.6), diuraikan hak akses analis terhadap sistem. Analis dapat menambah data project yang baru, dan dapat melihat surat penugasan pengerjaan project. Selain analis, Programmer juga bisa melihat surat penugasan project yang ditugaskan pada programmer tersebut (Gambar 4.6).

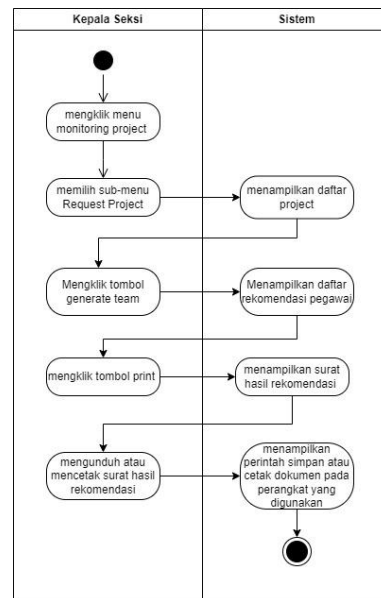


Gambar 4. 7
Diagram *Activity* Ajukan Project
Sumber: Dokumen Pribadi

Berdasarkan gambar di atas, diuraikan hak akses programmer terhadap sistem. Programmer dapat mengajukan project yang sedang dikerjakannya untuk dilanjutkan ke tahap selanjutnya.



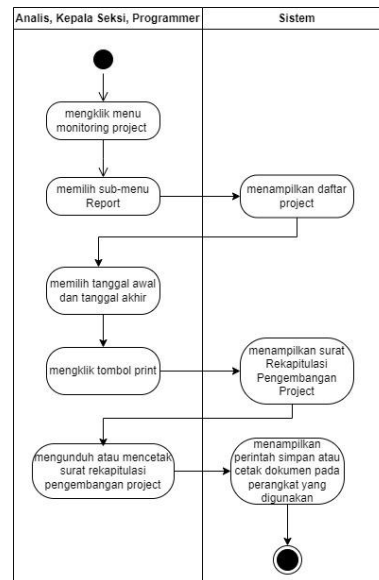
Gambar 4. 8
Diagram *Activity* Generate Team
Sumber: Dokumen Pribadi



Gambar 4. 9
Diagram *Activity* Cetak Hasil Rekomendasi Programmer
Sumber: Dokumen Pribadi

Berdasarkan Gambar 4.8 dan Gambar 4.9, diuraikan hak akses Kepala Seksi terhadap sistem. Pada sub-menu request project, Kepala Seksi dapat memilih programmer yang direkomendasikan berdasarkan hasil perhitungan SAW. Kepala Seksi dapat menentukan beberapa orang programmer yang akan mengerjakan project tersebut. Kepala Seksi juga dapat mencetak hasil rekomendasi pengerjaan project tersebut.

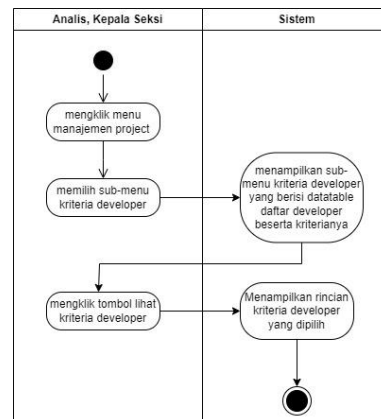
4) Diagram Activity sub-menu Report



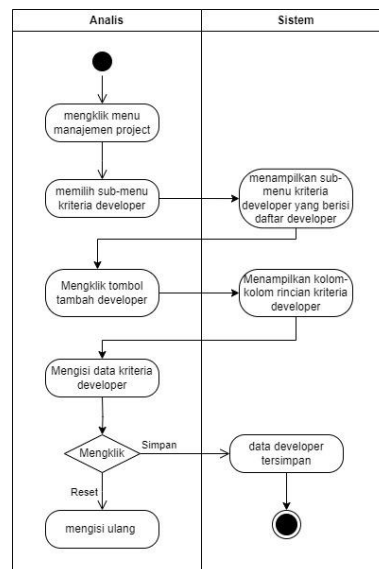
Gambar 4. 10
Diagram *Activity* sub-menu Report
Sumber: Dokumen Pribadi

Berdasarkan gambar di atas, diuraikan hak akses Analis, Kepala Seksi, dan Programmer terhadap sistem. Analis, Kepala Seksi dan Programmer dapat melihat laporan pengerjaan project pada periode waktu tertentu. Analis dan Kepala Seksi dapat melihat semua data project, sedangkan programmer hanya dapat melihat project yang sedang dikerjakan oleh programmer yang bersangkutan.

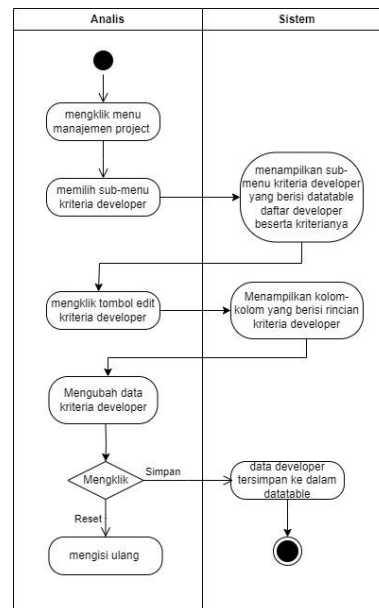
5) Diagram Activity sub-menu Kriteria Developer



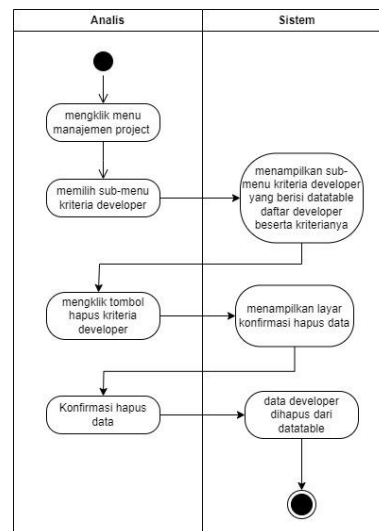
Gambar 4. 11
Diagram *Activity* sub-menu Kriteria Developer
Sumber: Dokumen Pribadi



Gambar 4. 12
Diagram *Activity* Tambah Kriteria Developer
Sumber: Dokumen Pribadi



Gambar 4. 13
Diagram *Activity* Ubah Kriteria Developer
Sumber: Dokumen Pribadi



Gambar 4. 14
Diagram *Activity* Hapus Kriteria Developer
Sumber: Dokumen Pribadi

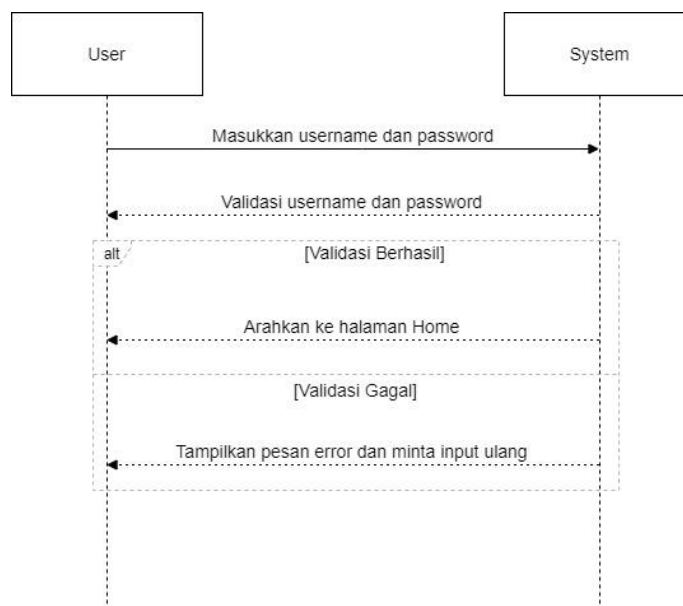
Berdasarkan Gambar 4.11 diuraikan hak akses Kepala Seksi terhadap sistem. Kepala Seksi dapat melihat rincian data kriteria developer yang sudah disimpan pada *database*. Berdasarkan

Gambar 4.11, Gambar 4.12, Gambar 4.13, dan Gambar 4.14, diuraikan hak akses Analis terhadap sistem. Analis dapat menambahkan, mengubah, dan menghapus data kriteria developer yang selanjutnya akan disimpan dalam database dan menjadi dasar bahan perhitungan SAW untuk menentukan penugasan pengerjaan project.

c. *Sequence Diagram*

1) *Sequence Diagram Login*

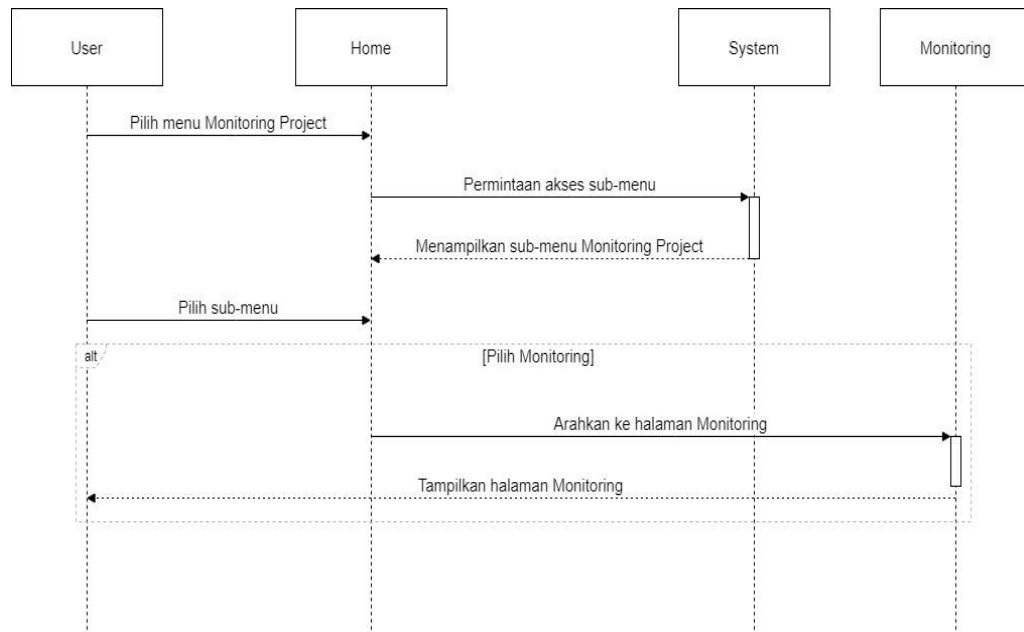
Berikut merupakan *sequence diagram* login pada Sistem penugasan pengerjaan *project*.



Gambar 4. 15
Diagram *Sequence* Login
Sumber: Dokumen Pribadi

2) *Sequence Diagram Monitoring Project*

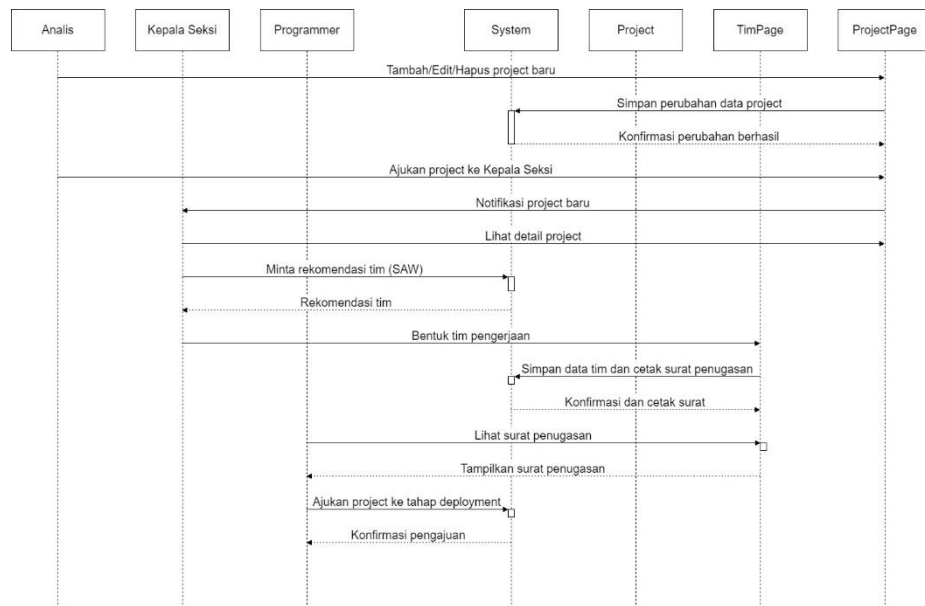
Berikut merupakan *sequence diagram* sub-menu monitoring project pada sistem



Gambar 4. 16
Diagram *Sequence* Monitoring Project
Sumber: Dokumen Pribadi

3) *Sequence Diagram Request Project*

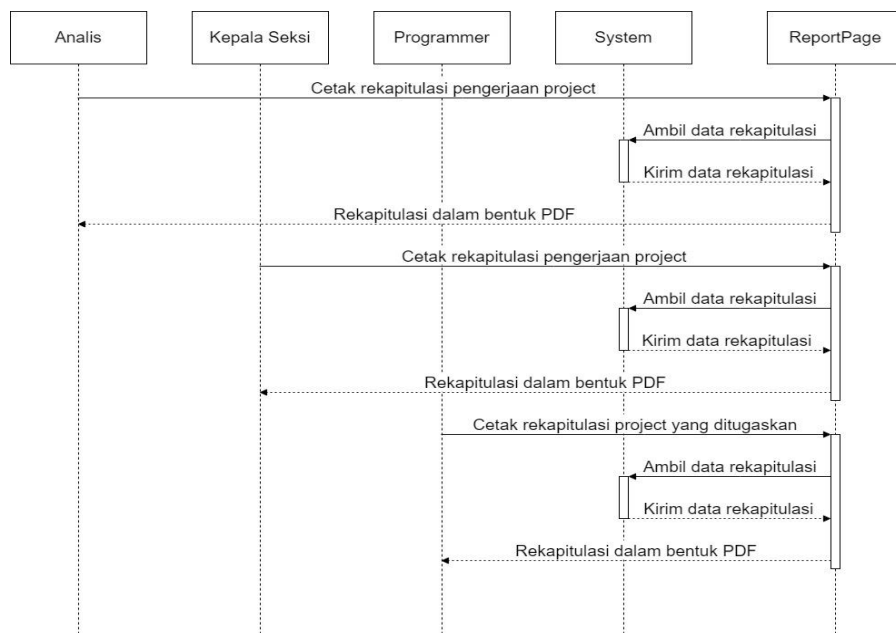
Berikut merupakan *sequence diagram* sub-menu request project pada sistem yang terdiri dari proses tambah, edit, dan hapus data project, memilih tim developer berdasarkan perhitungan SAW, mencetak surat penugasan, mengajukan project untuk dilanjutkan ke tahap selanjutnya.



Gambar 4. 17
Diagram Sequence Request Project
Sumber: Dokumen Pribadi

4) Sequence Diagram Report

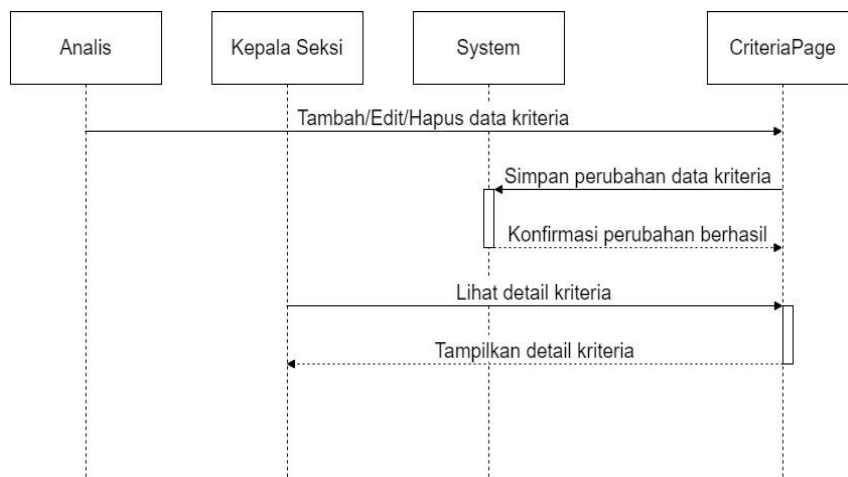
Berikut merupakan sequence diagram sub-menu report yang terdiri dari proses melihat rekapitulasi perkembangan project dan mencetak hasil rekapitulasi perkembangan pengerjaan project.



Gambar 4. 18
Diagram Sequence Report
Sumber: Dokumen Pribadi

5) *Sequence Diagram* Kriteria Developer

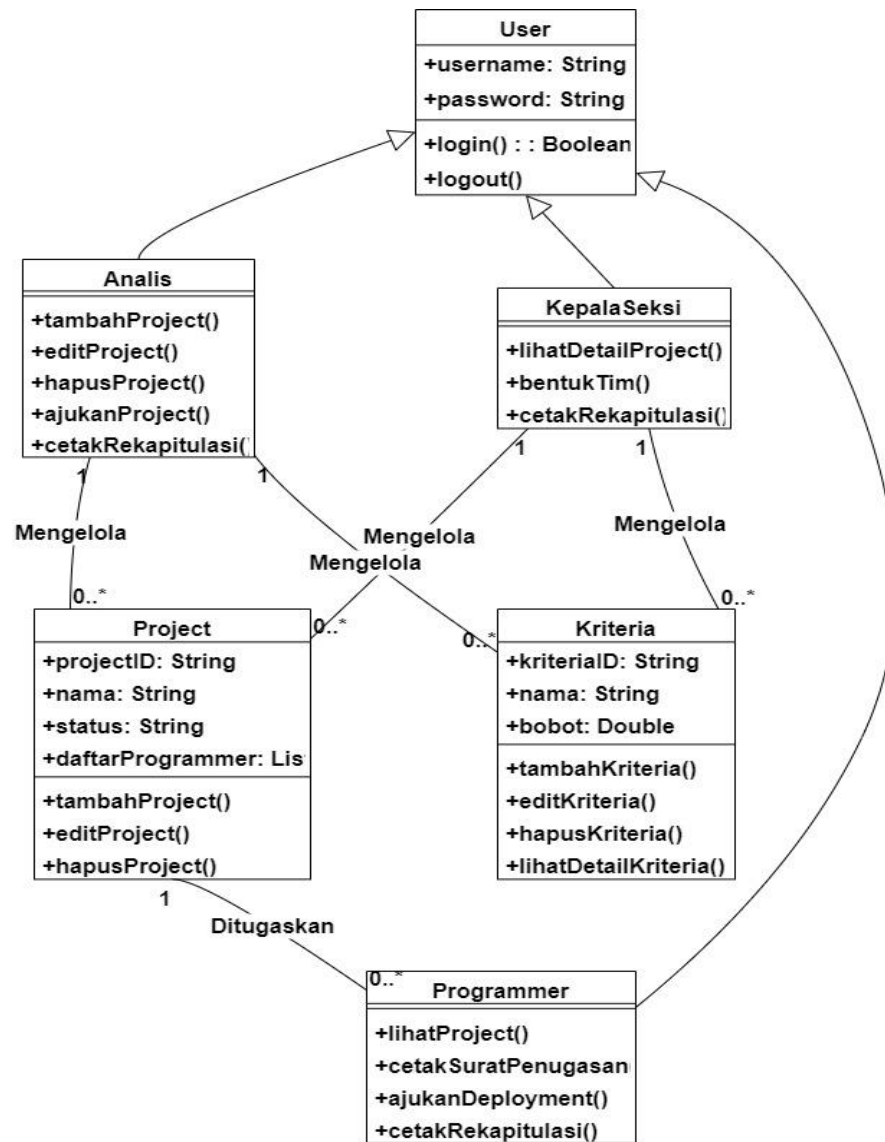
Berikut merupakan sequence diagram sub-menu kriteria developer pada sistem yang terdiri dari proses menambahkan, menghapus, dan mengubah data developer.



Gambar 4. 19
Diagram Sequence Kriteria Developer
Sumber: Dokumen Pribadi

d. *Class Diagram*

Berikut merupakan class diagram dari sistem pendukung keputusan untuk menentukan penugasan pengerjaan project (Gambar 4. 20)

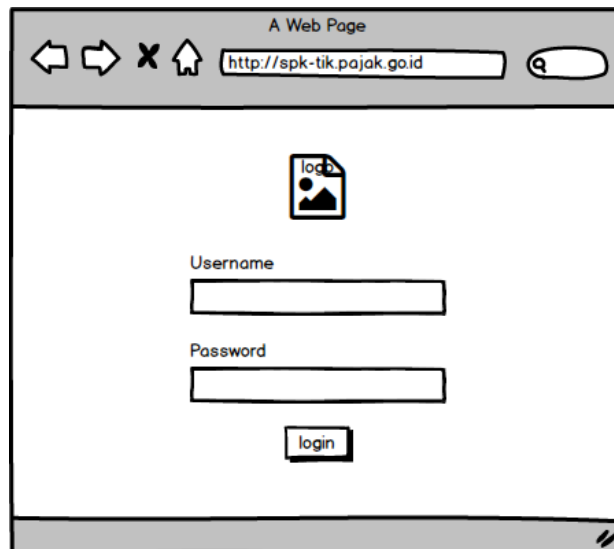


Gambar 4. 20
 Class Diagram SPK Menentukan Penugasan Project
 Sumber: Dokumen Pribadi

2. Rancangan Layar

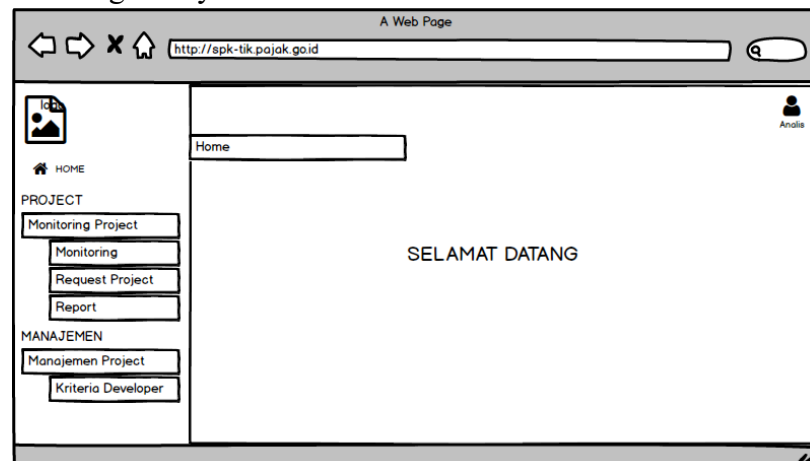
Rancangan layar merupakan rancangan bangun percakapan antara pengguna dengan komputer yang terdiri dari proses memasukkan data ke sistem kemudian menampilkan kembali keluaran informasi kepada pengguna.

a. Rancangan Layar Form Login



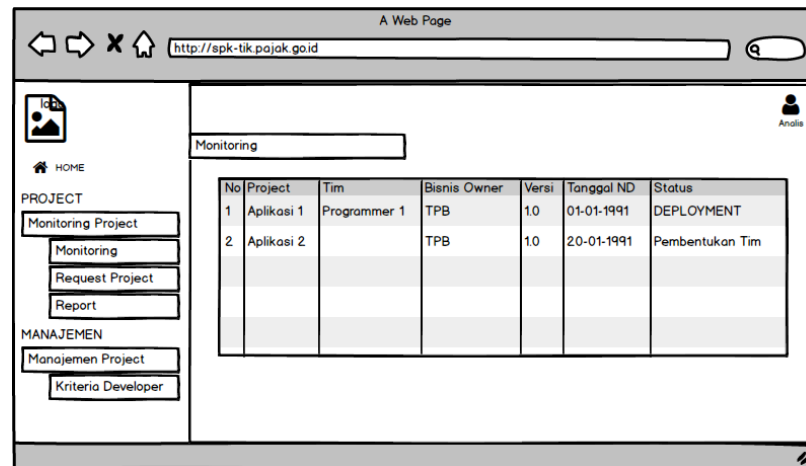
Gambar 4. 21
Rancangan *Form Login*
Sumber: Dokumen Pribadi

b. Rancangan Layar Menu Home



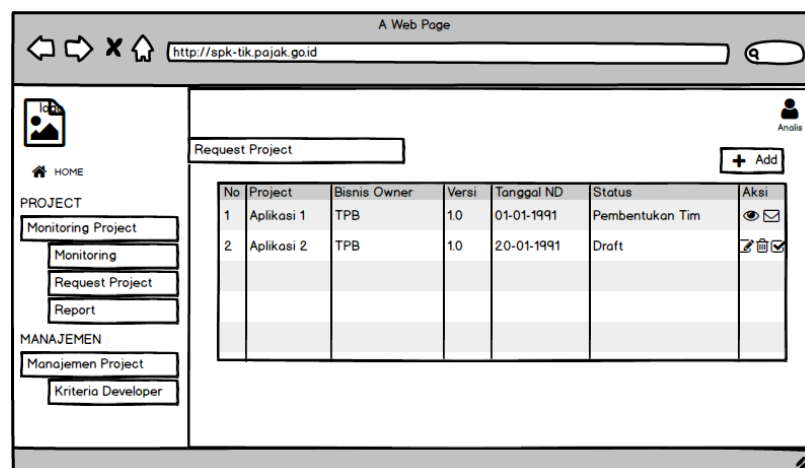
Gambar 4. 22
Rancangan Layar Menu *Home*
Sumber: Dokumen Pribadi

c. Rancangan Layar sub-menu *Monitoring*

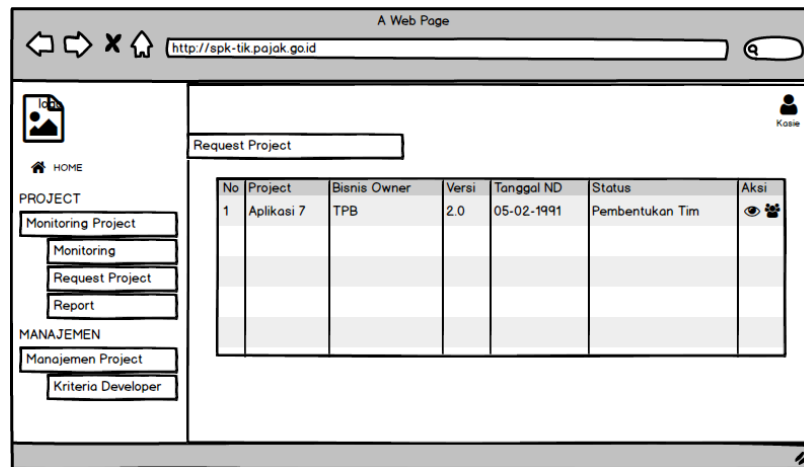


Gambar 4. 23
Rancangan Layar sub-menu *Monitoring*
Sumber: Dokumen Pribadi

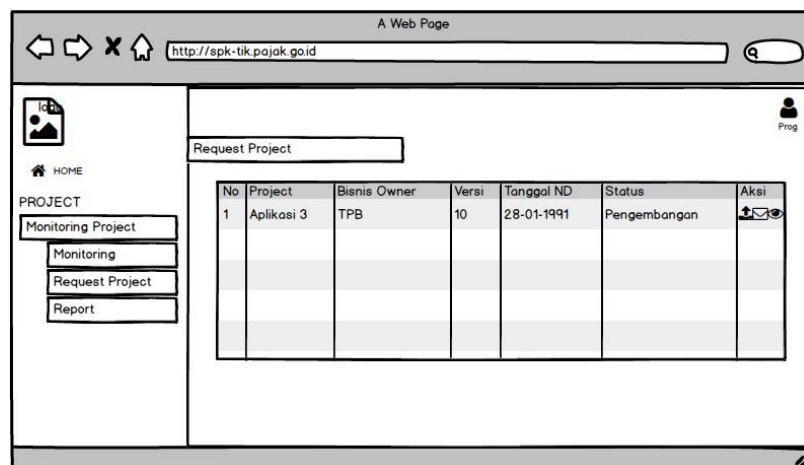
d. Rancangan Layar sub-menu *Request Project*



Gambar 4. 24
Rancangan Layar sub-menu *Request Project* Aktor Analis
Sumber: Dokumen Pribadi



Gambar 4. 25
Rancangan Layar sub-menu *Request Project* Aktor Kasi
Sumber: Dokumen Pribadi



Gambar 4. 26
Rancangan Layar sub-menu *Request Project* Aktor *Programmer*
Sumber: Dokumen Pribadi

e. Rancangan Layar Tambah *Project*

The screenshot shows a web browser window with the URL <http://spk-tik.pajak.go.id>. The main content area displays a sidebar with navigation links under 'PROJECT' (Monitoring Project, Monitoring, Request Project, Report) and 'MANAJEMEN' (Manajemen Project, Kriteria Development). A modal window titled 'Add Project' is open, containing the following form elements:

- Nama Project:
- Stack yang digunakan:
- Bisnis Owner:
- Versi:
- No. ND Permintaan:
- Tanggal Nota Dinas:
- Jenis:
- Buttons: RESET, SIMPAN

Gambar 4. 27
Rancangan Layar Tambah Project
Sumber: Dokumen Pribadi

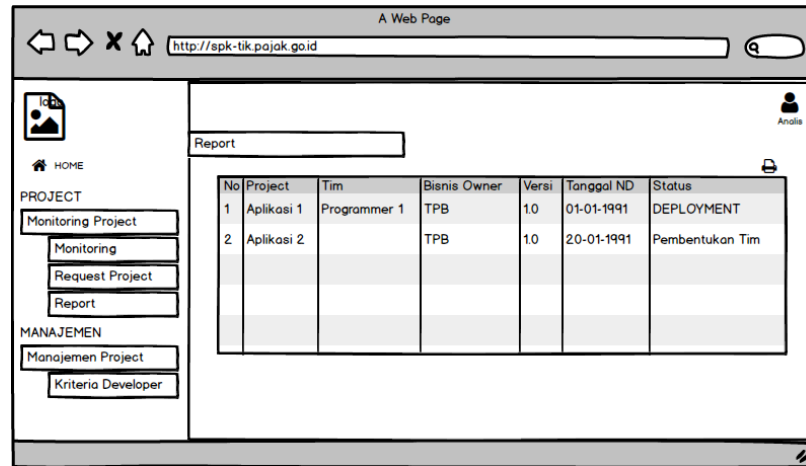
f. Rancangan Layar *Generate Team*

The screenshot shows the same web browser window. The modal window titled 'Preferensi Rekomendasi Pegawai' is open, displaying a table of employee recommendations:

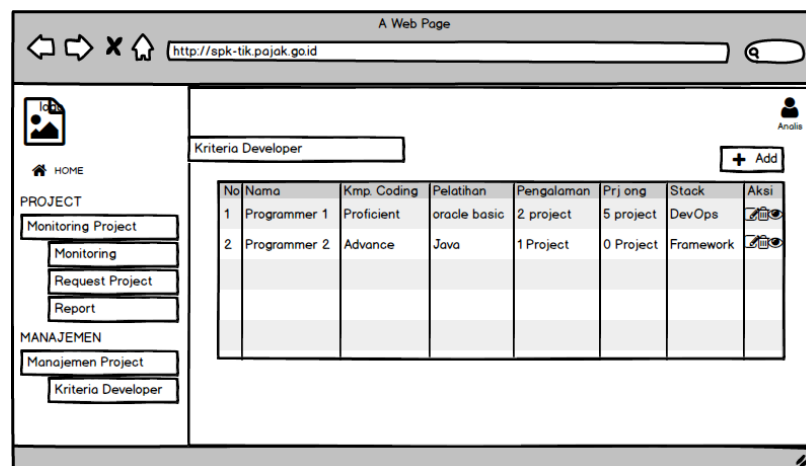
No	Nama	NIP	Jabatan	Rating	Aksi
1	Programmer 5	1234567	Programmer	0.79	<input checked="" type="checkbox"/>
2	Programmer 10	2345678	Programmer	0.54	<input type="checkbox"/>
3	Programmer 4	5678934	Programmer	0.43	<input type="checkbox"/>

Buttons: Close, Generate Team

Gambar 4. 28
Rancangan Layar *Generate Team*
Sumber: Dokumen Pribadi

g. Rancangan Layar sub-menu *Report*

Gambar 4. 29
Rancangan Layar sub-menu *Report*
Sumber: Dokumen Pribadi

h. Rancangan Layar sub-menu *Kriteria Developer*

Gambar 4. 30
Rancangan Layar sub-menu *Kriteria Developer*
Sumber: Dokumen Pribadi

i. Rancangan Layar Tambah Kriteria *Developer*

The screenshot shows a web browser window with the address bar displaying 'http://spk-tik.pajak.go.id'. The main content area features a sidebar with navigation links: HOME, PROJECT (Monitoring Project, Monitoring, Request Project, Report), and MANAJEMEN (Manajemen Project, Kriteria Developer). A modal window titled 'Add Kriteria Developer' is open, containing four dropdown menus: 'Nama Developer', 'Skill Programming Level', 'Penguasaan Stack', and 'Pelatihan'. At the bottom of the modal are 'RESET' and 'SIMPAN' buttons. A small 'Add' button is also visible in the top right corner of the modal.

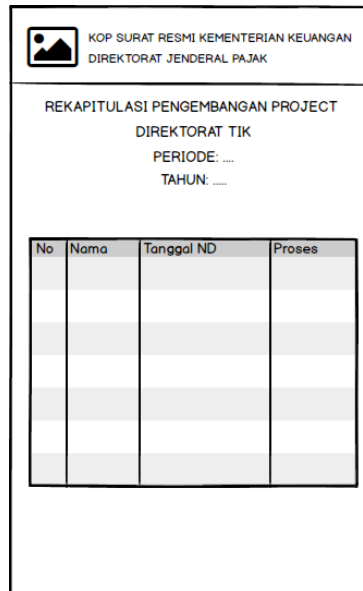
Gambar 4. 31
Rancangan Layar Tambah Kriteria *Developer*
Sumber: Dokumen Pribadi

j. Rancangan Layar Cetak Surat Penugasan

The screenshot shows a document template for a formal assignment letter. At the top left is a logo and the text 'KOP SURAT RESMI KEMENTERIAN KEUANGAN DIREKTORAT JENDERAL PAJAK'. Below this is the heading 'Nomor Surat'. The main body contains the following text: 'Hal : Penugasan Pengembangan Aplikasi', 'Tanggal:', 'Isi Surat terdiri dari:', 'Spesifikasi Project', 'Bahasa Pemrograman :', 'Database :', 'Jenis :', 'Ditetapkan tim pengembang', 'Nama Analis :', 'Nama Programmer :', and 'NAMA ATASAN' at the bottom right.

Gambar 4. 32
Rancangan Layar Cetak Surat Penugasan
Sumber: Dokumen Pribadi

k. Rancangan Layar Cetak Rekapitulasi Pengembangan *Project*



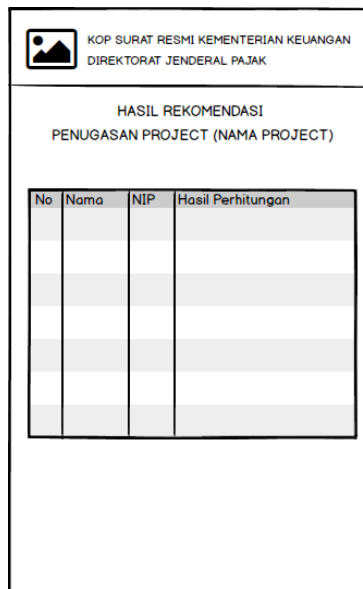
KOP SURAT RESMI KEMENTERIAN KEUANGAN
DIREKTORAT JENDERAL PAJAK

REKAPITULASI PENGEMBANGAN PROJECT
DIREKTORAT TIK
PERIODE:
TAHUN:

No	Nama	Tanggal ND	Proses

Gambar 4. 33
Rancangan Layar Cetak Rekapitulasi Pengembangan Project
Sumber: Dokumen Pribadi

l. Rancangan Layar Cetak Hasil Rekomendasi Penugasan *Project*



KOP SURAT RESMI KEMENTERIAN KEUANGAN
DIREKTORAT JENDERAL PAJAK

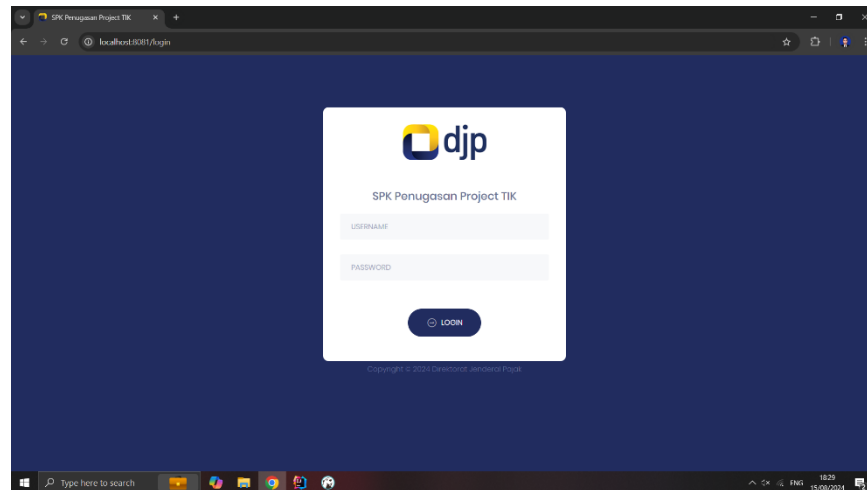
HASIL REKOMENDASI
PENUGASAN PROJECT (NAMA PROJECT)

No	Nama	NIP	Hasil Perhitungan

Gambar 4. 34
Rancangan Layar Cetak Hasil Rekomendasi Penugasan Project
Sumber: Dokumen Pribadi

3. Tampilan Layar

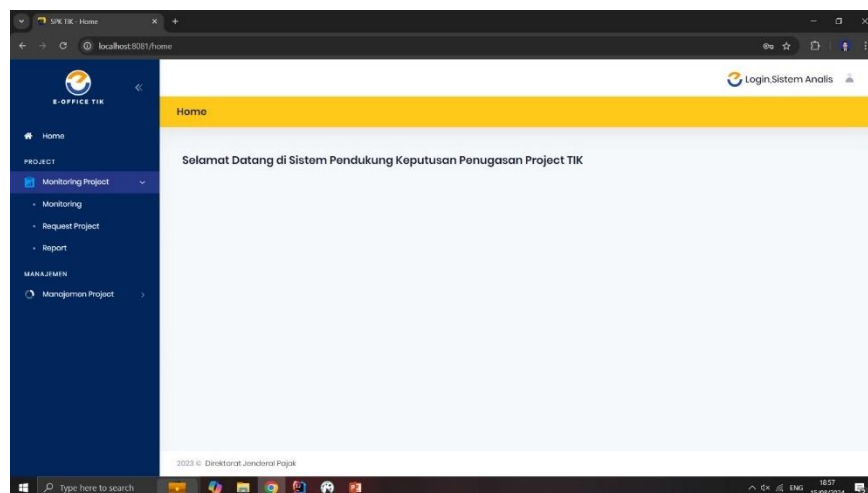
a. Tampilan Layar Menu Login



Gambar 4. 35
Tampilan Layar Menu Login
Sumber: Dokumen Pribadi

Form login ini akan muncul saat pertama kali mengakses aplikasi. Form akan diisi oleh aktor analis, kepala seksi, atau *programmer* dengan memasukkan *username* dan *password*. Apabila *username* dan *password* valid, maka akan ditampilkan menu *Home* aplikasi.

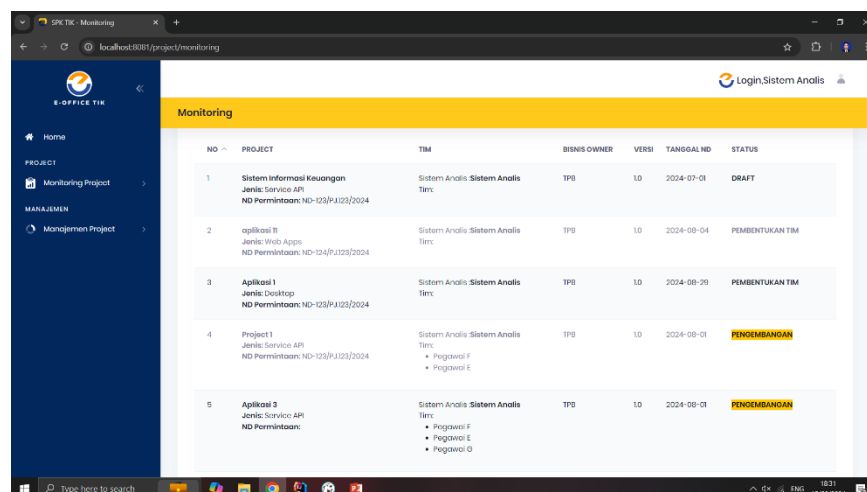
b. Tampilan Layar Menu Home



Gambar 4. 36
Tampilan Layar Menu Home
Sumber: Dokumen Pribadi

Layar Home merupakan tampilan pertama yang akan diakses oleh pengguna ketika berhasil login ke aplikasi. Semua aktor memiliki tampilan *Home* yang sama. Pada layar Home ini juga dapat dilihat ada menu utama yaitu Monitoring Project dengan sub-menu monitoring, request project, dan report. Serta menu utama lainnya adalah Manajemen project dengan sub-menu kriteria *developer*.

c. Tampilan Layar sub-menu *Monitoring*

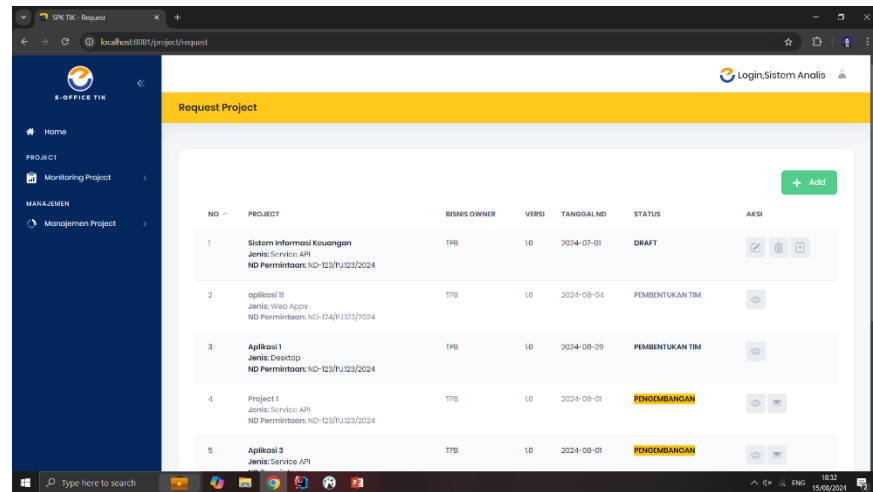


NO	PROJECT	TM	BISNIS OWNER	VERSI	TANGGAL ND	STATUS
1	Sistem Informasi Keuangan Jenis: Service API ND Permintaan: ND-123/P/123/2024	Sistem Analisis Sistem Analisis Tim:	TPB	1.0	2024-07-01	DRAFT
2	Aplikasi II Jenis: Web Apps ND Permintaan: ND-124/P/123/2024	Sistem Analisis Sistem Analisis Tim:	TPB	1.0	2024-08-04	PEMBENTUKAN TIM
3	Aplikasi I Jenis: Desktop ND Permintaan: ND-124/P/123/2024	Sistem Analisis Sistem Analisis Tim:	TPB	1.0	2024-08-09	PEMBENTUKAN TIM
4	Project 1 Jenis: Service API ND Permintaan: ND-123/P/123/2024	Sistem Analisis Sistem Analisis Tim: • Pegawai F • Pegawai E	TPB	1.0	2024-08-01	PENDEMBANGAN
5	Aplikasi 3 Jenis: Service API ND Permintaan:	Sistem Analisis Sistem Analisis Tim: • Pegawai F • Pegawai E • Pegawai O	TPB	1.0	2024-08-01	PENDEMBANGAN

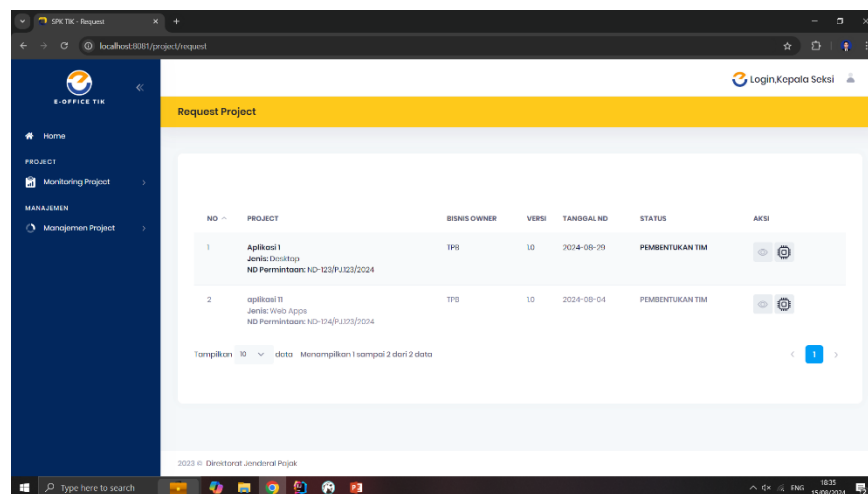
Gambar 4. 37
Tampilan Layar sub-menu *Monitoring*
Sumber: Dokumen Pribadi

Layar sub-menu monitoring project merupakan tampilan yang muncul ketika aktor mengakses menu monitoring project. Pada sub-menu ini akan ditampilkan datatable rincian *project* yang sudah diinput. Tampilan layar menu monitoring ini, sama pada setiap aktor.

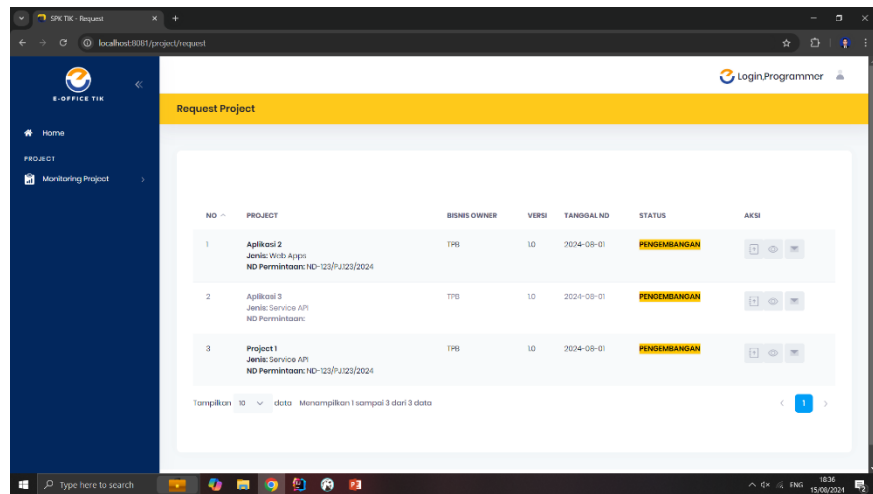
d. Tampilan Layar sub-menu *Request Project*



Gambar 4. 38
Tampilan Layar sub-menu *Request Project* Aktor Analis
Sumber: Dokumen Pribadi



Gambar 4. 39
Tampilan Layar sub-menu *Request Project* Aktor Kasi
Sumber: Dokumen Pribadi



Gambar 4. 40
Tampilan Layar sub-menu Request Project Aktor Programmer
Sumber: Dokumen Pribadi

Layar sub-menu *request project* berisi tampilan datatable project yang sudah diinput. Jika pengguna login sebagai aktor analis, maka dapat melihat, mengubah, dan mengajukan pengerjaan project. Pada halaman ini analis juga dapat menambahkan daftar project baru. Jika pengguna login sebagai aktor kepala seksi, maka dapat melakukan pembentukan tim. Jika pengguna login sebagai aktor *programmer*, maka dapat mengajukan deploy project, dan mencetak surat penugasan project tersebut.

e. Tampilan Layar Tambah *Project*

Gambar 4. 41
Tampilan Layar Tambah *Project*
Sumber: Dokumen Pribadi

Layar tambah project merupakan formulir yang berisi kolom-kolom rincian project yang akan ditampilkan pada menu monitoring project. Tampilan layar ini hanya berlaku oleh pengguna aktor analis.

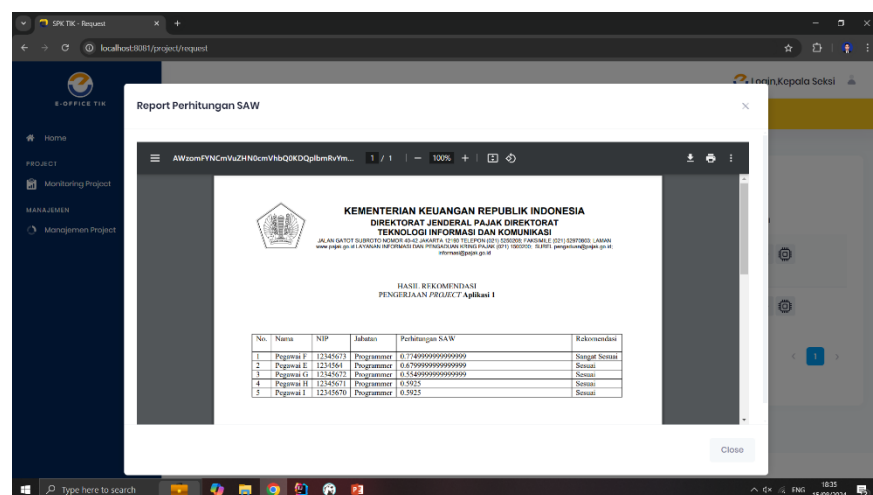
f. Tampilan Layar *Generate Team*

NO	NAMA	NIP	JABATAN	PERHITUNGAN REKOMENDASI	REKOMENDASI	AKSI
1	Pegawai F	12345673	Programmer	0.7743999999999999	Sangat sesuai	<input type="checkbox"/>
2	Pegawai E	1234564	Programmer	0.6799999999999999	Sesuai	<input type="checkbox"/>
3	Pegawai H	1234567	Programmer	0.5925	Sesuai	<input type="checkbox"/>
4	Pegawai I	12345670	Programmer	0.5925	Sesuai	<input type="checkbox"/>
5	Pegawai G	12345672	Programmer	0.5543999999999999	Sesuai	<input type="checkbox"/>

Gambar 4. 42
Tampilan Layar *Generate Team*
Sumber: Dokumen Pribadi

Layar preferensi rekomendasi pegawai berisi datatable nama *programmer* beserta hasil perhitungan dengan metode SAW sebagai bahan pertimbangan. Tampilan layar ini hanya berlaku oleh pengguna aktor kepala seksi. Pada tampilan layar ini, pengguna dapat memilih salah satu atau lebih *programmer* yang akan ditugaskan menjadi tim untuk mengerjakan *project* tersebut. Pada tampilan layar ini juga, pengguna dapat mencetak hasil rekomendasi penugasan *project* tersebut.

g. Tampilan Layar Cetakan Hasil Rekomendasi Penugasan *Project*

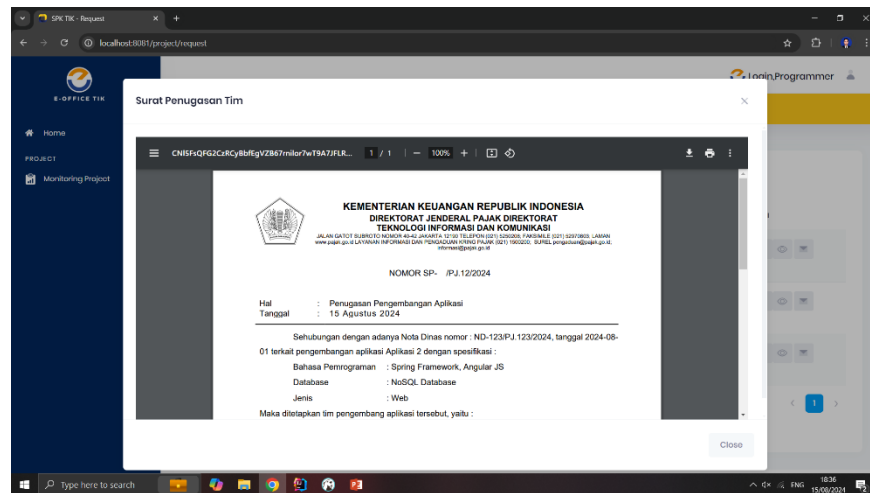


Gambar 4. 43
Tampilan Layar Cetakan Hasil Rekomendasi Penugasan Project
Sumber: Dokumen Pribadi

Tampilan layar hasil rekomendasi penugasan project berbentuk *pop-up* pdf yang berisi kop surat, datatable terdiri dari nama, NIP, Jabatan, nilai hasil perhitungan SAW, dan status rekomendasi masing-masing pegawai. Pada tampilan layar ini, pengguna dapat mengunduh atau

mencetak dokumen yang telah ditampilkan. Tampilan layar ini hanya berlaku oleh pengguna aktor kepala seksi.

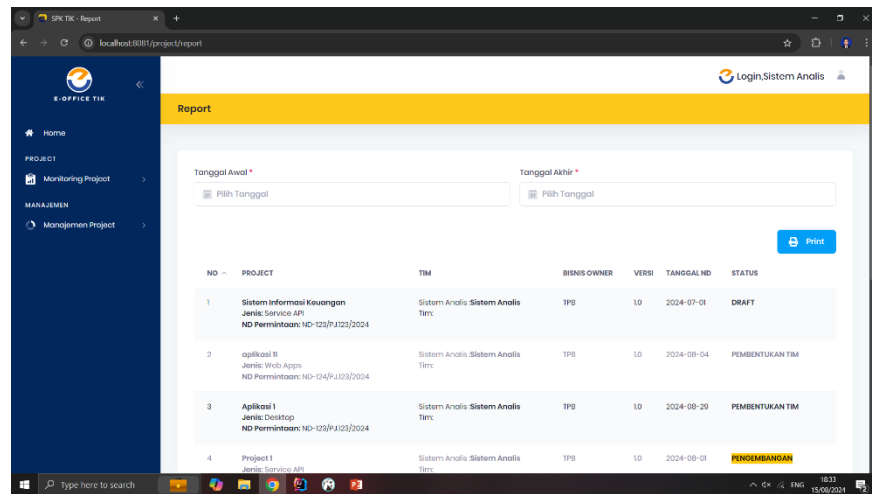
h. Tampilan Layar Cetak Surat Penugasan



Gambar 4. 44
Tampilan Layar Cetak Surat Penugasan
Sumber: Dokumen Pribadi

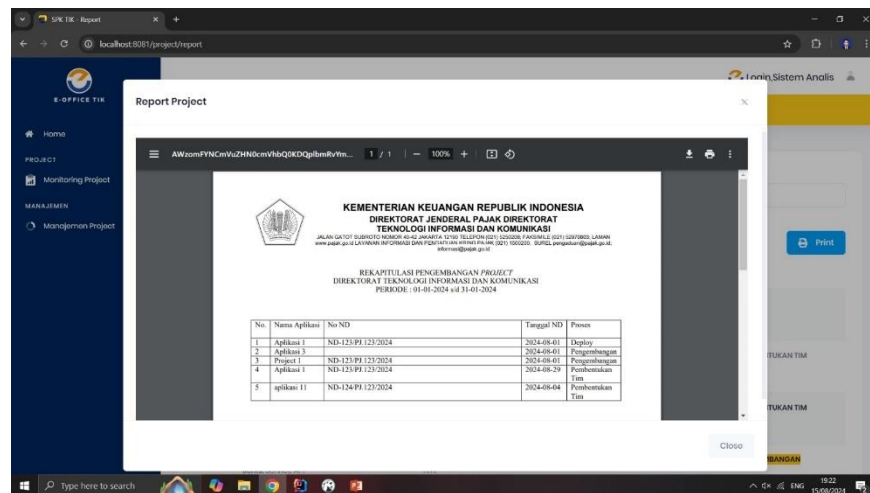
Tampilan layar surat penugasan pengerjaan *project* berbentuk *pop-up* pdf yang berisi kop surat, nomor surat, dan isi surat. Pada tampilan layar ini, pengguna dapat mengunduh dan mencetak dokumen yang telah ditampilkan. Tampilan layar ini hanya berlaku oleh pengguna aktor analis, dan *programmer*.

i. Tampilan Layar sub-menu *Report*



Gambar 4. 45
Tampilan Layar sub-menu Report
Sumber: Dokumen Pribadi

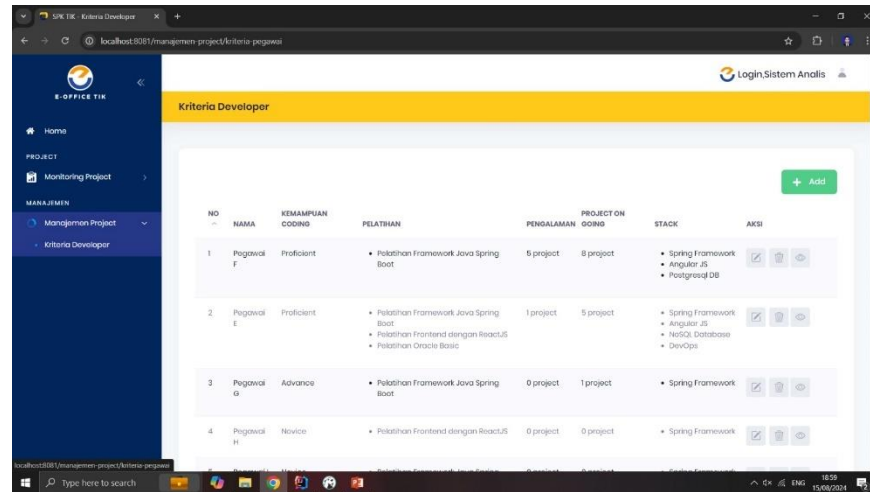
Tampilan layar sub-menu report ini, data yang ditampilkan pada layar untuk aktor analis sama dengan aktor kepala seksi. Layar sub-menu *report* berisi datatable semua *project-project* yang sedang dikerjakan di Direktorat TIK. Sedangkan, tampilan layar sub-menu *report* pada aktor *programmer*, berisi datatable *project* yang sedang dikerjakan oleh pengguna tersebut. Pada tampilan layar sub-menu *report* ini, pengguna juga dapat mencetak rekapitulasi pengembangan project.

j. Tampilan Layar Cetakan Rekapitulasi Pengembangan *Project*

Gambar 4. 46
Tampilan Layar Cetakan Rekapitulasi Pengembangan *Project*
Sumber: Dokumen Pribadi

Tampilan layar rekapitulasi pengembangan *project* berbentuk *pop-up* pdf yang berisi kop surat, datatable rincian project terdiri dari nama *project*, No. Nota Dinas, tanggal Nota Dinas, dan proses pengerjaan *project*. Pada tampilan layar ini, pengguna dapat mengunduh dan mencetak dokumen yang telah ditampilkan. Tampilan layar sub-menu *report* ini sama pada setiap aktor.

k. Tampilan Layar sub-menu Kriteria *Developer*



NO	NAMA	KEMAMPUAN CODING	PELATIHAN	PENGALAMAN	PROJECT ON GOING	STACK	AKSI
1	Pegawai F	Proficient	<ul style="list-style-type: none"> • Pelatihan Framework Java Spring Boot 	6 project	8 project	<ul style="list-style-type: none"> • Spring Framework • Angular JS • PostgreSQL DB 	[Edit] [Delete] [Add]
2	Pegawai E	Proficient	<ul style="list-style-type: none"> • Pelatihan Framework Java Spring Boot • Pelatihan Frontend dengan ReactJS • Pelatihan Oracle Base 	1 project	5 project	<ul style="list-style-type: none"> • Spring Framework • Angular JS • NoSQL Database • DevOps 	[Edit] [Delete] [Add]
3	Pegawai G	Advance	<ul style="list-style-type: none"> • Pelatihan Framework Java Spring Boot 	0 project	1 project	<ul style="list-style-type: none"> • Spring Framework 	[Edit] [Delete] [Add]
4	Pegawai H	Novice	<ul style="list-style-type: none"> • Pelatihan Frontend dengan ReactJS 	0 project	0 project	<ul style="list-style-type: none"> • Spring Framework 	[Edit] [Delete] [Add]

Gambar 4. 47
Tampilan Layar sub-menu Kriteria *Developer*
Sumber: Dokumen Pribadi

Tampilan layar ini berisi datatable *programmer/developer* yang sudah diinput, pengguna dapat melihat rincian data, mengubah, dan menghapus data yang sudah ada. Tampilan layar sub-menu kriteria developer hanya bisa diakses oleh aktor analis dan kepala seksi. Aktor analis dapat menambahkan data baru. Sedangkan, aktor kepala seksi hanya bisa melihat daftar *developer* beserta kriteria yang diinput oleh aktor analis.

1. Tampilan Layar Tambah Kriteria *Developer*

Gambar 4. 48
Tampilan Layar Tambah Kriteria *Developer*
Sumber: Dokumen Pribadi

Tampilan layar ini berisi kolom-kolom rincian *developer* yang menjadi bahan perhitungan menggunakan metode SAW. Tampilan layar tambah kriteria developer hanya bisa diakses oleh pengguna aktor analis.

m. Tampilan Layar Cetak Surat

KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK DIREKTORAT
TEKNOLOGI INFORMASI DAN KOMUNIKASI
JALAN GATOT SUBROTO NOMOR 40-42, JAKARTA 12190 TELEPON (021) 5202008 FAKSIMILE (021) 5270803, LAMAH
 www.pajak.go.id LAYANAN INFORMASI DAN PETUGAS KIRI: 021 1500200, SURvei_pajak@pajak.go.id, informasi@pajak.go.id

REKAPITULASI PENGEMBANGAN *PROJECT*
 DIREKTORAT TEKNOLOGI INFORMASI DAN KOMUNIKASI
 PERIODE : 01-07-2024 s/d 31-08-2024

No.	Nama Aplikasi	No ND	Tanggal	Proses
1	Sistem Informasi Keuangan	ND-123/PJ.123/2024	2024-08-15	Pembentukan Tim
2	Project 1	ND-123/PJ.123/2024	2024-08-16	Pengembangan
3	Aplikasi Project A	ND-123/PJ.123/2024	2024-08-16	Pembentukan Tim
4	Project Bersi 1	ND-123/PJ.123/2024	2024-08-16	Deploy

Jakarta, 20 Agustus 2024
 Sistem Analis

Yam Suseno

Gambar 4. 49
Surat Rekapitulasi Pengembangan Project
Sumber: Dokumen Pribadi



**KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK DIREKTORAT
TEKNOLOGI INFORMASI DAN KOMUNIKASI**

JALAN GATOT SUBROTO NOMOR 40-42, JAKARTA 12190 TELEPON (021) 5250208, FAKSIMILE (021) 52970803, LAMAH
www.pajak.go.id LAYANAN INFORMASI DAN PENGADUAN KRING PAJAK (021) 1500200, SUREL: pengaduan@pajak.go.id,
informasi@pajak.go.id

**HASIL REKOMENDASI
PENGERJAAN *PROJECT* Sistem Informasi Keuangan**

No.	Nama	NIP	Jabatan	Perhitungan SAW	Rekomendasi
1	Erwin	12345673	Programmer	0.6383333333333333	Sesuai
2	Novita	1234564	Programmer	0.61	Sesuai
3	Teguh	12345672	Programmer	0.3866666666666667	Kurang Sesuai
4	Natali	12345671	Programmer	0.4416666666666667	Kurang Sesuai
5	Febri	12345670	Programmer	0.4416666666666667	Kurang Sesuai
6	Joko	12345679	Programmer	0.4816666666666667	Kurang Sesuai
7	Faisal	12345676	Programmer	0.5733333333333334	Sesuai
8	Vina	12345666	Programmer	0.5966666666666667	Sesuai

Jakarta, 20 Agustus 2024

Kepala Seksi

Asep Hidayat

Gambar 4. 50
Surat Hasil Rekomendasi Programmer
Sumber: Dokumen Pribadi



**KEMENTERIAN KEUANGAN REPUBLIK INDONESIA
DIREKTORAT JENDERAL PAJAK DIREKTORAT
TEKNOLOGI INFORMASI DAN KOMUNIKASI**

JALAN GATOT SUBROTO NOMOR 40-42, JAKARTA 12190 TELEPON (021) 5250208, FAKSIMILE (021) 52970803, LAMAH
www.pajak.go.id LAYANAN INFORMASI DAN PENGADUAN KRING PAJAK (021) 1500200, SUREL: pengaduan@pajak.go.id,
informasi@pajak.go.id

NOMOR SP- /PJ.12/2024

Hal : Penugasan Pengembangan Aplikasi
Tanggal : 20 Agustus 2024

Sehubungan dengan adanya Nota Dinas nomor : ND-124/PJ.123/2024, tanggal 2024-08-04 terkait pengembangan aplikasi aplikasi 11 dengan spesifikasi :

Bahasa Pemrograman : Angular JS
Database : Postgresql DB
Jenis : Web

Maka ditetapkan tim pengembang aplikasi tersebut, yaitu :

Sistem Analis : Yan Suseno
Programmer : Vina, Erwin, Novita, Joko

Demikian disampaikan agar ditindaklanjuti, terimakasih.

Jakarta, 20 Agustus 2024

Kepala Seksi PSPM

Asep Hidayat

Gambar 4. 51
Surat Penugasan Pengerjaan Project
Sumber: Dokumen Pribadi

D. Kelebihan dan Kelemahan Penelitian

1. Kelebihan Penelitian

- a. Kelebihan yang dimiliki metode SAW adalah dapat menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan pemberian ranking yang berguna untuk menyeleksi alternatif terbaik dari sejumlah alternatif
- b. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan sebelumnya.

2. Kelemahan Penelitian

Data yang digunakan dalam metode SAW harus benar dan tepat, agar tidak menimbulkan kesalahan pada saat pembobotan dan perankingan kriteria.

BAB V

PENUTUP

A. SIMPULAN

Berdasarkan pembahasan dan hasil yang diperoleh, maka dapat disimpulkan bahwa Sistem Pendukung Keputusan dengan menggunakan metode SAW untuk menentukan penugasan pengerjaan *project* pada Direktorat TIK, Direktorat Jenderal Pajak sebagai berikut:

1. Sistem aplikasi yang dirancang dapat membantu proses manajemen *project* dalam menentukan penugasan pengerjaan *project* secara tepat sasaran dengan memilih pegawai yang direkomendasikan oleh sistem yang menggunakan Metode SAW.
2. Penerapan Metode SAW dalam menentukan kandidat pegawai (*developer/programmer*) untuk menentukan penugasan pengerjaan *project* berdasarkan kriteria kemampuan *coding*, jumlah *project* yang sedang dikerjakan, jumlah pelatihan yang diikuti, pengalaman, dan penguasaan *stack*.
3. Data dalam penerapan Metode SAW pada aplikasi Sistem Pendukung Keputusan untuk menentukan penugasan pengerjaan *project* disajikan dalam bentuk laporan hasil rekomendasi penugasan *project* yang telah diurutkan sesuai dengan perhitungan, Selain itu, aplikasi ini juga akan menyajikan surat tugas pengerjaan *project*, dan rekapitulasi pengembangan *project*.

B. SARAN

Saran yang dapat diberikan penulis pada akhir skripsi ini adalah sebagai berikut:

1. Aplikasi Sistem Pendukung Keputusan ini dapat diintegrasikan sebagai modul baru pada Sistem manajemen *project* yang sudah ada saat ini di Direktorat TIK, Direktorat Jenderal Pajak.
2. Sebelum sistem baru dijalankan, sebaiknya seluruh pihak yang terkait dalam sistem ini harus diberikan penjelasan mengenai proses kerja sistem yang akan diterapkan sehingga tidak terjadi kekeliruan.
3. Meskipun telah menggunakan sistem yang terkomputerisasi, ketelitian dalam menginput data perlu diperhatikan.

DAFTAR PUSTAKA

- Ahmad, N., Krisnanik, E., Rupilele, F. G. J., Muliawati, A., Syamsiyah, N., Dwi Cahyono, B., Sriyeni, Y., Kristanto, T., Irwanto, & Guntoro. (2022). *ANALISA & PERANCANGAN SISTEM INFORMASI BERORIENTASI OBJEK*. Penerbit Widina Media Utama. www.penerbitwidina.com
- Belferik, R., Andiyan, Zulkarnain, I., Munizu, M., Samosir, J. M., Afriyadi, H., Rusmiatmoko, D., Adhicandra, I., Syamil, A., Ichsan, M., & Prasetyo, A. (2023). *MANAJEMEN PROYEK: Teori & Penerapannya*. Sonpedia Publishing Indonesia. <https://www.researchgate.net/publication/371175194>
- Fernando, J. G., & Baldelovar, M. (2022). Decision Support System: Overview, Different Types and Elements. *Technoarete Transactions on Intelligent Data Mining and Knowledge Discovery*, 2(2), 13–18.
- Gautama, K., & Suwarno. (2021). Analisis Dan Perancangan Sistem Pendukung Keputusan Untuk Menentukan Prioritas Pengerjaan Aplikasi Menggunakan Metode Simple Additive Weighting. *Journal of Information System and Technology*, 02(01), 1–23.
- Görsch, C., Seppänen, O., Peltokorpi, A., & Lavikka, R. (2024). Task planning and control in construction: revealing workers as early and late planners. *Construction Management and Economics*, 42(5), 431–450. <https://doi.org/10.1080/01446193.2023.2270080>
- Jiang, Y. (2020). Research on Application Value of Computer Software Development in Java Programming Language. *Journal of Physics: Conference Series*, 1648(3). <https://doi.org/10.1088/1742-6596/1648/3/032152>
- Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021). *UML Diagrams in Software Engineering Research: A Systematic Literature Review*. 13. <https://doi.org/10.3390/proceedings2021074013>
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2018). *DATABASE PROCESSING FUNDAMENTALS, DESIGN, AND IMPLEMENTATION*. www.pearson.com
- Martinez, D. D., Remegio, A. H. P., & Lincopinis, D. R. (2023). *A Review on Java Programming Language*. <https://orcid.org/0000-0001-9503-8965>,
- Musfikar, R., Akbar, I., Dewi, S. V., & Aziz, A. S. (2023). E-Module Bahasa Pemrograman Java Berbasis Exe-Learning. *Jurnal PROCESSOR*, 18(1). <https://doi.org/10.33998/processor.2023.18.1.704>
- Noval, Q., Handrianto, Y., & Supendar, H. (2020). Sistem Pendukung Keputusan Dalam Menentukan Karyawan Terbaik Menggunakan Metode Simple

- Additive Weighting. *Jurnal Infortech*, 2(1), 116–121.
<http://ejournal.bsi.ac.id/ejurnal/index.php/infortech116>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering A PRACTITIONER'S APPROACH Ninth Edition*. McGrawHill.
- Putri, R. A. (2022). *Buku Ajar BASIS DATA*. Media Sains Indonesia.
www.medsan.co.id
- Rahmansyah, N., Kom, S., Kom, M., & Lusia, S. A. (2021). *Buku Ajar SISTEM PENDUKUNG KEPUTUSAN*.
- Santiara, N. (2017). *PENERAPAN METODE SAW (SIMPLE ADDITIVE WEIGHTING) PADA MANAJEMEN PROYEK UNTUK MENENTUKAN PEGAWAI DALAM Pengerjaan Suatu Proyek (STUDI KASUS PT. DELIMAN INTEGRA TEKNOLOGI)*. <https://digilib.uinsgd.ac.id/14986/>
- Saputra, M. A. W., Aslam, M. N., Ridha, M., & Yaqin, M. A. (2021). Sistem Pendukung Keputusan Untuk Menentukan Manajer Proyek Menggunakan Metode Simple Additive Weighting (SAW). *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 3(3), 373–386.
<https://doi.org/10.28926/ilkomnika.v3i3.386>
- Saputra, M. H. (2019). *Sistem Pendukung Keputusan Penentuan Prioritas Penyelesaian Proyek Dengan Metode Simple Additive Weighting (SAW) di PT Nindya Karya (Persero) Kanwil 1 Palembang*. https://repository.unsri.ac.id/817/1/RAMA_57201_09031181520024_0013096707_0013058302_01_front_ref.pdf
- Sharda, Ramesh., Delen, Dursun., Turban, Efraim., Aronson, J. E. ., Liang, T.-Peng., & King, David. (2018). *Business intelligence, analytics, and data science: a managerial perspective*. Pearson.
- Silalahi, F. D. (2022). *Manajemen Database MySQL (Structured Query Language)*. Yayasan Prima Agus Teknik.
- Spezie, M., & Bragantini, D. (2023). The Development of a Task Management Software (TMS): A bridge between Project Management's sub-activities (especially in multi-project context) and "ordinary" assignments to follow. *PM World Journal*, XII, 2330–4480.
www.pmworljournal.com
- Su, J., Wang, J., Liu, S., Zhang, N., & Li, C. (2020). A Method for Efficient Task Assignment Based on the Satisfaction Degree of Knowledge. *Complexity*, 2020. <https://doi.org/10.1155/2020/3543782>
- Sumirat, L. P., Cahyono, D., Kristyawan, Y., & Kacung, S. (2023). *DASAR-DASAR Rekayasa Perangkat Lunak*. Madza Media. www.madzamedia.co.id
- Taherdoost, H. (2023). Analysis of Simple Additive Weighting Method (SAW) as a MultiAttribute Decision-Making Technique: A Step-by-Step Guide.

Journal of Management Science & Engineering Research, 6(1), 21–24.
<https://doi.org/10.30564/jmser.v6i1.5400>

Utama, D. N. (2017). *Sistem Penunjang Keputusan*. Penerbit Garudhawaca.
<https://books.google.co.id/books?id=4oihDgAAQBAJ&printsec=copyright&hl=id#v=onepage&q&f=false>

DAFTAR NARASUMBER

1.



Nama : Asep Hidayat
Tempat/ Tanggal Lahir : -
Pekerjaan : PNS
Waktu Wawancara : Mei 2024
Tempat Wawancara : Direktorat TIK, DJP
Kompetensi sesuai objek penelitian : Kepala Seksi
2.



Nama : Yan Suseno
Tempat/ Tanggal Lahir : -
Pekerjaan : PNS
Waktu Wawancara : Mei 2024
Tempat Wawancara : Direktorat TIK, DJP
Kompetensi sesuai objek penelitian : Sistem Analis
3.



Nama : Erwin Adinugraha
Tempat/ Tanggal Lahir : -
Pekerjaan : PNS
Waktu Wawancara : Mei 2024
Tempat Wawancara : Direktorat TIK, DJP
Kompetensi sesuai objek penelitian : Programmer

DAFTAR RIWAYAT HIDUP



Fauzan Saef, lahir di Payakumbuh, 15 Januari 1995. Saat ini tinggal di Jalan Veteran RT.9/RW.6, Duren Tiga, Pancoran, Jakarta Selatan. Pendidikan Dasar saya tempuh di SD Klapanunggal Bogor dari tahun 2001-2002, SD Deli Maju Medan dari tahun 2002-2006, SD Muhammadiyah 11 Medan dari tahun 2006-2007, Pendidikan menengah di SMPN 16 Medan dari tahun 2007-2010, SMAN 1 Medan dari tahun 2010-2013. Pendidikan tinggi di Universitas Sumatera Utara dari tahun 2013-2016. Saat ini saya bekerja sebagai Pegawai Negeri Sipil.

LISTING PROGRAM

A. Monitoring Project

```
1. package com.skripsi.skripsi.service;
2.
3. import com.skripsi.skripsi.auth.UserDetailsImpl;
4. import com.skripsi.skripsi.dto.AplikasiDTO;
5. import com.skripsi.skripsi.dto.ReportDTO;
6. import com.skripsi.skripsi.dto.SKepDTO;
7. import com.skripsi.skripsi.dto.TbPerangkinganDTO;
8. import com.skripsi.skripsi.entity.*;
9. import com.skripsi.skripsi.repository.*;
10. import com.skripsi.skripsi.utility.ReportUtil;
11. import com.skripsi.skripsi.utility.SAWUtil;
12. import com.skripsi.skripsi.utility.MessageResponse;
13. import org.apache.catalina.User;
14. import org.modelmapper.ModelMapper;
15. import org.springframework.beans.factory.annotation.Autowired;
16. import org.springframework.data.domain.Sort;
17. import org.springframework.security.core.context.SecurityContextHolder;
18. import org.springframework.stereotype.Service;
19. import org.springframework.transaction.annotation.Transactional;
20.
21. import java.time.LocalDate;
22. import java.util.ArrayList;
23. import java.util.Arrays;
24. import java.util.List;
25. import java.util.stream.Collectors;
26.
27. @Service
28. @Transactional
29. public class ProjectService implements IProjectService {
30.     private final TbAplikasiRepo tbAplikasiRepo;
31.     private final TrxBahasaPemrogramanRepo trxBahasaPemrogramanRepo;
32.     private final TrxJenisDatabaseRepo trxJenisDatabaseRepo;
33.     private final TbKriteriaPegawaiMatrixRepo tbKriteriaPegawaiMatrixRepo;
34.     private final TbTimRepo tbTimRepo;
35.     private final ModelMapper modelMapper;
36.     private final SAWUtil SAWUtil;
37.     private final ReportUtil reportUtil;
38.     private final TbKriteriaPegawaiRepo tbKriteriaPegawaiRepo;
39.     private final RefStackRepo refStackRepo;
40.
41.
42.     @Autowired
43.     public ProjectService(TbAplikasiRepo tbAplikasiRepo, TrxBahasaPemrogramanRepo
trxBahasaPemrogramanRepo, TrxJenisDatabaseRepo
trxJenisDatabaseRepo,
TbKriteriaPegawaiMatrixRepo
tbKriteriaPegawaiMatrixRepo, TbTimRepo
tbTimRepo,
ModelMapper modelMapper, SAWUtil SAWUtil, ReportUtil reportUtil, TbKriteriaPegawaiRepo
tbKriteriaPegawaiRepo,
RefStackRepo refStackRepo) {
44.
45.         this.tbAplikasiRepo = tbAplikasiRepo;
46.         this.trxBahasaPemrogramanRepo = trxBahasaPemrogramanRepo;
```

```

47.     this.trxJenisDatabaseRepo = trxJenisDatabaseRepo;
48.     this.tbKriteriaPegawaiMatrixRepo = tbKriteriaPegawaiMatrixRepo;
49.     this.tbTimRepo = tbTimRepo;
50.     this.modelMapper = modelMapper;
51.     this.SAWUtil = SAWUtil;
52.     this.reportUtil = reportUtil;
53.     this.tbKriteriaPegawaiRepo = tbKriteriaPegawaiRepo;
54.     this.refStackRepo = refStackRepo;
55. }
56.
57. @Override
58. public List<TbAplikasi> getAllProject() {
59.     return tbAplikasiRepo.findAll(Sort.by(Sort.Direction.DESC, "id"));
60. }
61.
62. @Override
63. public List<TbAplikasi> getAllProjectByProses(int proses) {
64.     return tbAplikasiRepo.findAllByProses(proses);
65. }
66.
67. @Override
68. public List<TbAplikasi> getAllProjectByProsesPegawai(int proses) {
69.     UserDetailsImpl userDetails = (UserDetailsImpl)
SecurityContextHolder.getContext().getAuthentication().getDetails();
70.     List<TbTim> tbTimList =
tbTimRepo.findAllByIdPegawai(userDetails.getPegawai().getId());
71.     List<TbAplikasi> tbAplikasiList = new ArrayList<>();
72.
73.
74.     if (proses == 0) {
75.         System.out.println(tbTimList.size());
76.         if (!tbTimList.isEmpty()) {
77.             tbTimList.forEach(tbTim -> {
78.                 tbAplikasiList.add(tbTim.getTbAplikasi());
79.             });
80.             return tbAplikasiList;
81.         }
82.     } else {
83.         if (!tbTimList.isEmpty()) {
84.             tbTimList.forEach(tbTim -> {
85.                 if (tbTim.getTbAplikasi().getProses() == proses)
86.                     tbAplikasiList.add(tbTim.getTbAplikasi());
87.             });
88.             return tbAplikasiList;
89.         }
90.     }
91.
92.     return tbAplikasiList;
93. }
94.
95. @Override
96. public MessageResponse getProjectById(int id) {
97.     if (!tbAplikasiRepo.existsById(id)) {
98.         return new MessageResponse(0, "Project tidak ditemukan", null);
99.     }

```

```

100.
101.     return new MessageResponse(1, "Success", tbAplikasiRepo.findById(id).get());
102. }
103.
104. @Override
105. public MessageResponse saveProject(AplikasiDTO aplikasiDTO) {
106.     UserDetailsImpl userDetails = (UserDetailsImpl)
SecurityContextHolder.getContext().getAuthentication().getDetails();
107.     TbAplikasi tbAplikasi = modelMapper.map(aplikasiDTO, TbAplikasi.class);
108.     tbAplikasi.setProses(0);//0:draft, 1:pengembangan, 2:testing, 3:deploy
109.     tbAplikasi.setStack(Arrays.toString(aplikasiDTO.getListStack()));
110.     tbAplikasi.setAnalisis(userDetails.getPegawai().getId());
111.     tbAplikasi.setCreatedAt(java.time.LocalDate.now());
112.     tbAplikasiRepo.save(tbAplikasi);
113.
114. //     setListBahasaPemrogramanDanDatabase(aplikasiDTO, tbAplikasi);
115.
116.     return new MessageResponse(1, "Project berhasil disimpan", tbAplikasi);
117. }
118.
119. @Override
120. public MessageResponse updateProject(int id, AplikasiDTO aplikasiDTO) {
121.
122.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
123.
124.     if (tbAplikasi == null) {
125.         return new MessageResponse(0, "Project tidak ditemukan", null);
126.     }
127.
128.
129.     if (!trxBahasaPemrogramanRepo.findAllByIdAplikasi(id).isEmpty()) {
130.         trxBahasaPemrogramanRepo.findAllByIdAplikasi(id).forEach(trxBahasaPemrograman
-> {
131.             trxBahasaPemrogramanRepo.deleteById(trxBahasaPemrograman.getId());
132.         });
133.     }
134.
135.     if (!trxJenisDatabaseRepo.findAllByIdAplikasi(id).isEmpty()) {
136.         trxJenisDatabaseRepo.findAllByIdAplikasi(id).forEach(trxJenisDatabase -> {
137.             trxJenisDatabaseRepo.deleteById(trxJenisDatabase.getId());
138.         });
139.     }
140.
141. //     setListBahasaPemrogramanDanDatabase(aplikasiDTO, tbAplikasi);
142.     tbAplikasi.setStack(Arrays.toString(aplikasiDTO.getListStack()));
143.     modelMapper.map(aplikasiDTO, tbAplikasi);
144.     tbAplikasiRepo.save(tbAplikasi);
145.
146.     return new MessageResponse(1, "Project berhasil diupdate", tbAplikasi);
147. }
148.
149.     private void setListBahasaPemrogramanDanDatabase(AplikasiDTO aplikasiDTO,
TbAplikasi tbAplikasi) {

```



```

150.
Arrays.stream(aplikasiDTO.getListBahasaPemrograman()).collect(Collectors.toList()).forEach(ba
hasaPemrograman -> {
151.     TrxBahasaPemrograman trxBahasaPemrograman = new TrxBahasaPemrograman();
152.     trxBahasaPemrograman.setIdAplikasi(tbAplikasi.getId());
153.     trxBahasaPemrograman.setIdBahasaPemrograman(bahasaPemrograman);
154.     trxBahasaPemrogramanRepo.save(trxBahasaPemrograman);
155. });
156.
157.
Arrays.stream(aplikasiDTO.getListJenisDatabase()).collect(Collectors.toList()).forEach(jenisData
base -> {
158.     TrxJenisDatabase trxJenisDatabase = new TrxJenisDatabase();
159.     trxJenisDatabase.setIdAplikasi(tbAplikasi.getId());
160.     trxJenisDatabase.setIdDatabase(jenisDatabase);
161.     trxJenisDatabaseRepo.save(trxJenisDatabase);
162. });
163. }
164.
165. @Override
166. public MessageResponse deleteProject(int id) {
167.     if (tbAplikasiRepo.existsById(id)) {
168.         tbAplikasiRepo.deleteById(id);
169.         return new MessageResponse(1, "Project berhasil dihapus", null);
170.     } else {
171.         return new MessageResponse(0, "Project tidak ditemukan", null);
172.     }
173. }
174.
175. @Override
176. public MessageResponse ajukanProject(int id) {
177.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
178.     if (tbAplikasi == null) {
179.         return new MessageResponse(0, "Project tidak ditemukan", null);
180.     }
181.
182.     if (tbAplikasi.getProses() > 0) {
183.         return new MessageResponse(0, "Project sudah diajukan", null);
184.     }
185.
186.     tbAplikasi.setProses(1);
187.     tbAplikasiRepo.save(tbAplikasi);
188.     return new MessageResponse(1, "Project berhasil diajukan pengembangan", tbAplikasi);
189. }
190.
191. @Override
192. public MessageResponse ajukanDeployment(int id) {
193.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
194.     if (tbAplikasi == null) {
195.         return new MessageResponse(0, "Project tidak ditemukan", null);
196.     }
197.
198.     if (tbAplikasi.getProses() == 4) {
199.         return new MessageResponse(0, "Project sudah di deploy", null);
200.     }

```

```

201.
202.     tbAplikasi.setProses(4);
203.     tbAplikasiRepo.save(tbAplikasi);
204.
205.         UserDetailsImpl userDetails = (UserDetailsImpl)
SecurityContextHolder.getContext().getAuthentication().getDetails();
206.     //update jumlah pengalaman project
207.         TbKriteriaPegawai tbKriteriaPegawai =
tbKriteriaPegawaiRepo.findByIdPegawai(userDetails.getPegawai().getId()).orElse(null);
208.     if (tbKriteriaPegawai != null) {
209.         tbKriteriaPegawai.setJumlahPengalaman(tbKriteriaPegawai.getJumlahPengalaman() +
1);
210.         tbKriteriaPegawaiRepo.save(tbKriteriaPegawai);
211.     }
212.         TbKriteriaPegawaiMatrix tbKriteriaPegawaiMatrix =
tbKriteriaPegawaiMatrixRepo.findByIdPegawai(userDetails.getPegawai().getId()).orElse(null);
213.     if (tbKriteriaPegawaiMatrix != null) {
214.         //C3
215.         if (tbKriteriaPegawai.getJumlahPengalaman() <= 1) {
216.             tbKriteriaPegawaiMatrix.setPengalaman(1);
217.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 2) {
218.             tbKriteriaPegawaiMatrix.setPengalaman(2);
219.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 3) {
220.             tbKriteriaPegawaiMatrix.setPengalaman(3);
221.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 4) {
222.             tbKriteriaPegawaiMatrix.setPengalaman(4);
223.         } else {
224.             tbKriteriaPegawaiMatrix.setPengalaman(5);
225.         }
226.         tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
227.     }
228. }
229.
230.
231.     return new MessageResponse(1, "Project berhasil diajukan deployment", tbAplikasi);
232. }
233.
234. @Override
235. public MessageResponse perhitunganSaw(int id) {
236.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
237.     if (tbAplikasi == null) {
238.         return new MessageResponse(0, "Project tidak ditemukan", null);
239.     }
240.         List<TbKriteriaPegawaiMatrix> tbKriteriaPegawaiMatrixList =
tbKriteriaPegawaiMatrixRepo.findAll();
241.
242.         tbKriteriaPegawaiMatrixList.forEach(tbKriteriaPegawaiMatrix -> {
243.             TbKriteriaPegawai tbKriteriaPegawai =
tbKriteriaPegawaiRepo.findByIdPegawai(tbKriteriaPegawaiMatrix.getIdPegawai()).orElse(null);
244.             String[] arrayStackAplikasi = tbAplikasi.getStack().replace("[", "").replace("]",
"".split(","));
245.             String[] arrayStackPegawai = new String[0];
246.             if (tbKriteriaPegawai != null) {
247.                 arrayStackPegawai = tbKriteriaPegawai.getPenguasaanStack().replace("[",
"".replace("]", "").split(",");

```

```

248.     }
249.
250.         ArrayList<String> listStackPegawai = new
ArrayList<>(Arrays.asList(arrayStackPegawai));
251.         ArrayList<String> listStackAplikasi = new
ArrayList<>(Arrays.asList(arrayStackAplikasi));
252.
253.
254.     listStackPegawai.retainAll(listStackAplikasi);
255.
256.     //C5
257.     if (listStackPegawai.size() <= 1) {
258.         tbKriteriaPegawaiMatrix.setStack(1);
259.     } else if (listStackPegawai.size() == 2) {
260.         tbKriteriaPegawaiMatrix.setStack(2);
261.     } else if (listStackPegawai.size() == 3) {
262.         tbKriteriaPegawaiMatrix.setStack(3);
263.     } else if (listStackPegawai.size() == 4) {
264.         tbKriteriaPegawaiMatrix.setStack(4);
265.     } else if (listStackPegawai.size() > 4) {
266.         tbKriteriaPegawaiMatrix.setStack(5);
267.     } else {
268.         throw new RuntimeException("Error set C5");
269.     }
270.     tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
271. });
272. return SAWUtil.metodeSAW(tbKriteriaPegawaiMatrixList);
273. }
274.
275. @Override
276. public MessageResponse generateTimProject(List<Integer> idPegawais, int idAplikasi) {
277.     UserDetailsImpl userDetails = (UserDetailsImpl)
SecurityContextHolder.getContext().getAuthentication().getDetails();
278.
279.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(idAplikasi).orElse(null);
280.     if (tbAplikasi == null) {
281.         return new MessageResponse(0, "Project tidak ditemukan", null);
282.     }
283.
284.     tbAplikasi.setProses(2);
285.     tbAplikasi.setIdApproval(userDetails.getPegawai().getId());
286.     tbAplikasiRepo.save(tbAplikasi);
287.
288.     idPegawais.forEach(idPegawai -> {
289.         TbTim tbTim = new TbTim();
290.         tbTim.setIdAplikasi(idAplikasi);
291.         tbTim.setIdPegawai(idPegawai);
292.         tbTim.setRoleProject("Programmer");
293.         tbTimRepo.save(tbTim);
294.
295.         //update jumlah project ongoing
296.         TbKriteriaPegawai tbKriteriaPegawai =
tbKriteriaPegawaiRepo.findByIdPegawai(idPegawai).orElse(null);
297.         if (tbKriteriaPegawai != null) {

```

```

298.
tbKriteriaPegawai.setJumlahProjectOngoing(tbKriteriaPegawai.getJumlahProjectOngoing() + 1);
299.     tbKriteriaPegawaiRepo.save(tbKriteriaPegawai);
300.     }
301.
302.         TbKriteriaPegawaiMatrix  tbKriteriaPegawaiMatrix  =
tbKriteriaPegawaiMatrixRepo.findByIdPegawai(idPegawai).orElse(null);
303.     if (tbKriteriaPegawaiMatrix != null) {
304.         //C2
305.         switch (tbKriteriaPegawai.getJumlahProjectOngoing()) {
306.             case 0:
307.                 tbKriteriaPegawaiMatrix.setProjectOngoing(1);
308.                 break;
309.             case 1:
310.                 tbKriteriaPegawaiMatrix.setProjectOngoing(2);
311.                 break;
312.             case 2:
313.                 tbKriteriaPegawaiMatrix.setProjectOngoing(3);
314.                 break;
315.             case 3:
316.                 tbKriteriaPegawaiMatrix.setProjectOngoing(4);
317.                 break;
318.             default:
319.                 tbKriteriaPegawaiMatrix.setProjectOngoing(5);
320.                 break;
321.         }
322.     }
323.
324.
325. });
326.
327.
328.     return new MessageResponse(1, "Tim project berhasil dibuat", null);
329. }
330.
331. @Override
332. public byte[] generateSKep(int id) throws Exception {
333.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
334.
335.     String[] arrayStackAplikasi = tbAplikasi.getStack().trim().replace("[", "").replace("]",
""").split(",");
336.
337.         int[]      intArrayStackAplikasi      =
Arrays.stream(arrayStackAplikasi).map(String::trim).mapToInt(Integer::parseInt).toArray();
338.
339.
340.     SKepDTO skepDTO = new SKepDTO();
341.     skepDTO.setAnalisis(tbAplikasi.getTbPegawaiAnalisis().getNama());
342.
343.     switch (tbAplikasi.getJenis()) {
344.         case 1:
345.             skepDTO.setJenis("Services API");
346.             break;
347.         case 2:
348.             skepDTO.setJenis("Mobile");

```

```

349.         break;
350.     case 3:
351.         skepDTO.setJenis("Web");
352.         break;
353.     case 4:
354.         skepDTO.setJenis("Desktop");
355.         break;
356.     default:
357.         skepDTO.setJenis("Unknown");
358.         break;
359. }
360. skepDTO.setNoNd(tbAplikasi.getNdRequest());
361. skepDTO.setTanggalNd(tbAplikasi.getTglNd().toString());
362. skepDTO.setNamaAplikasi(tbAplikasi.getNama());
363. skepDTO.setKepalaSeksi(tbAplikasi.getTbPegawaiApproval().getNama());
364.
365. List<String> bahasa = new ArrayList<>();
366. List<String> database = new ArrayList<>();
367. List<String> programmer = new ArrayList<>();
368.
369. for (int i = 0; i < intArrayStackAplikasi.length; i++) {
370.     RefStack refStack = refStackRepo.findById(intArrayStackAplikasi[i]).orElseThrow(()
-> new RuntimeException("Stack not found"));
371.
372.     if (refStack.getJenis() == 1) {
373.         bahasa.add(refStack.getNama());
374.     } else if (refStack.getJenis() == 2) {
375.         database.add(refStack.getNama());
376.     }
377. }
378.
379. tbAplikasi.getTim().forEach(tbTim -> {
380.     if (tbTim.getRoleProject().equals("Programmer")) {
381.         programmer.add(tbTim.getTbPegawai().getNama());
382.     }
383. });
384.
385. String resultBahasa = String.join(", ", bahasa);
386. String resultDatabase = String.join(", ", database);
387. String resultProgrammer = String.join(", ", programmer);
388. skepDTO.setBahasa(resultBahasa);
389. skepDTO.setDatabase(resultDatabase);
390. skepDTO.setProgrammer(resultProgrammer);
391. skepDTO.setNamaSeksi(tbAplikasi.getTbPegawaiApproval().getUnit());
392.
393. return reportUtil.generateSKep(skepDTO);
394. }
395.
396. @Override
397. public byte[] generateReport(LocalDate tglAwal, LocalDate tglAkhir) throws Exception {
398.     UserDetailsImpl userDetails = (UserDetailsImpl)
SecurityContextHolder.getContext().getAuthentication().getDetails();
399.     List<ReportDTO> reportDTOList = new ArrayList<>();
400.     if
(userDetails.getUser().getRefRole().getNamaRole().equals("ROLE_PROGRAMMER")) {

```

```

401.         System.out.println("cek");
402.                                     List<TbTim>      tbTimList      =
tbTimRepo.findAllByIdPegawai(userDetails.getPegawai().getId());
403.
404.         List<TbAplikasi> tbAplikasiList = new ArrayList<>();
405.
406.         if (!tbTimList.isEmpty()) {
407.             tbTimList.forEach(tbTim -> {
408.                 tbAplikasiList.add(tbTim.getTbAplikasi());
409.             });
410.         }
411.
412.         for (TbAplikasi tbAplikasi : tbAplikasiList) {
413.             if (tbAplikasi.getCreatedAt().isAfter(tglAwal) &&
tbAplikasi.getCreatedAt().isBefore(tglAkhir)) {
414.                 ReportDTO reportDTO = new ReportDTO();
415.                 reportDTO.setNamaAplikasi(tbAplikasi.getNama());
416.                 reportDTO.setNoNd(tbAplikasi.getNdRequest());
417.                 reportDTO.setTanggalRequest(tbAplikasi.getCreatedAt().toString());
418.                 switch (tbAplikasi.getProses()) {
419.                     case 0:
420.                         reportDTO.setProses("Draft");
421.                         break;
422.                     case 1:
423.                         reportDTO.setProses("Pembentukan Tim");
424.                         break;
425.                     case 2:
426.                         reportDTO.setProses("Pengembangan");
427.                         break;
428.                     case 3:
429.                         reportDTO.setProses("Testing");
430.                         break;
431.                     case 4:
432.                         reportDTO.setProses("Deploy");
433.                         break;
434.                     default:
435.                         reportDTO.setProses("Unknown");
436.                         break;
437.                 }
438.                 reportDTOList.add(reportDTO);
439.             }
440.         }
441.         return reportUtil.generateReport(reportDTOList, tglAwal, tglAkhir);
442.
443.     } else {
444.
445.         List<TbAplikasi> tbAplikasiList = tbAplikasiRepo.findAll();
446.         for (TbAplikasi tbAplikasi : tbAplikasiList) {
447.             if (tbAplikasi.getCreatedAt().isAfter(tglAwal) &&
tbAplikasi.getCreatedAt().isBefore(tglAkhir)) {
448.                 ReportDTO reportDTO = new ReportDTO();
449.                 reportDTO.setNamaAplikasi(tbAplikasi.getNama());
450.                 reportDTO.setNoNd(tbAplikasi.getNdRequest());
451.                 reportDTO.setTanggalRequest(tbAplikasi.getCreatedAt().toString());
452.                 switch (tbAplikasi.getProses()) {

```

```

453.         case 0:
454.             reportDTO.setProses("Draft");
455.             break;
456.         case 1:
457.             reportDTO.setProses("Pembentukan Tim");
458.             break;
459.         case 2:
460.             reportDTO.setProses("Pengembangan");
461.             break;
462.         case 3:
463.             reportDTO.setProses("Testing");
464.             break;
465.         case 4:
466.             reportDTO.setProses("Deploy");
467.             break;
468.         default:
469.             reportDTO.setProses("Unknown");
470.             break;
471.     }
472.     reportDTOList.add(reportDTO);
473. }
474. }
475.     return reportUtil.generateReport(reportDTOList, tglAwal, tglAkhir);
476. }
477. }
478.
479. @Override
480. public byte[] generateReportPegawai(int id) throws Exception {
481.     MessageResponse messageResponse = perhitunganSaw(id);
482.     TbAplikasi tbAplikasi = tbAplikasiRepo.findById(id).orElse(null);
483.     if (messageResponse.getStatus() == 0) {
484.         throw new RuntimeException("Error generate report pegawai");
485.     }
486.     List<TbPerangkinganDTO> tbPerangkinganDTOList = (List<TbPerangkinganDTO>)
messageResponse.getData();
487.     return reportUtil.generateReportPegawai(tbPerangkinganDTOList, tbAplikasi.getNama());
488. }
489.
490. }
491.

```

B. Manajemen Project

```
1. package com.skripsi.skripsi.service;
2.
3. import com.skripsi.skripsi.dto.KriteriaPegawaiDTO;
4. import com.skripsi.skripsi.dto.TbKriteriaPegawaiDTO;
5. import com.skripsi.skripsi.entity.RefPelatihan;
6. import com.skripsi.skripsi.entity.RefStack;
7. import com.skripsi.skripsi.entity.TbKriteriaPegawai;
8. import com.skripsi.skripsi.entity.TbKriteriaPegawaiMatrix;
9. import com.skripsi.skripsi.repository.*;
10. import com.skripsi.skripsi.utility.MessageResponse;
11. import org.modelmapper.ModelMapper;
12. import org.springframework.beans.factory.annotation.Autowired;
13. import org.springframework.stereotype.Service;
14. import org.springframework.transaction.annotation.Transactional;
15.
16. import java.util.ArrayList;
17. import java.util.Arrays;
18. import java.util.List;
19.
20. @Service
21. @Transactional
22. public class KriteriaPegawaiService implements IKriteriaPegawaiService {
23.
24.     private final TbKriteriaPegawaiRepo tbKriteriaPegawaiRepo;
25.     private final TbKriteriaPegawaiMatrixRepo tbKriteriaPegawaiMatrixRepo;
26.     private final TbTimRepo tbTimRepo;
27.     private final RefPelatihanRepo refPelatihanRepo;
28.     private final RefStackRepo refStackRepo;
29.     private final ModelMapper modelMapper;
30.
31.     @Autowired
32.     public KriteriaPegawaiService(TbKriteriaPegawaiRepo tbKriteriaPegawaiRepo,
33. TbKriteriaPegawaiMatrixRepo tbKriteriaPegawaiMatrixRepo, TbTimRepo tbTimRepo,
34. RefPelatihanRepo refPelatihanRepo, RefStackRepo refStackRepo, ModelMapper
35. modelMapper) {
36.         this.tbKriteriaPegawaiRepo = tbKriteriaPegawaiRepo;
37.         this.tbKriteriaPegawaiMatrixRepo = tbKriteriaPegawaiMatrixRepo;
38.         this.tbTimRepo = tbTimRepo;
39.         this.refPelatihanRepo = refPelatihanRepo;
40.         this.refStackRepo = refStackRepo;
41.         this.modelMapper = modelMapper;
42.     }
43.
44.     @Override
45.     public List<TbKriteriaPegawaiDTO> getKriteriaPegawai() {
46.         List<TbKriteriaPegawaiDTO> listTbKriteriaPegawaiDTO = new ArrayList<>();
47.         List<TbKriteriaPegawai> tbKriteriaPegawai = tbKriteriaPegawaiRepo.findAll();
48.
49.         tbKriteriaPegawai.forEach(kriteriaPegawai -> {
50.             TbKriteriaPegawaiDTO tbKriteriaPegawaiDTO = new TbKriteriaPegawaiDTO();
51.             tbKriteriaPegawaiDTO.setId(kriteriaPegawai.getId());
52.             tbKriteriaPegawaiDTO.setNamaPegawai(kriteriaPegawai.getTbPegawai().getNama());
```



```

51. tbKriteriaPegawaiDTO.setLvlKemampuanCoding(kriteriaPegawai.getRefSkillProgramming(
52. ).getNama());
53. String strArrPelatihan = kriteriaPegawai.getJumlahPelatihan();
54. strArrPelatihan = strArrPelatihan.substring(1, strArrPelatihan.length() - 1);
55. String[] strArrayPelatihan = strArrPelatihan.split(", ");
56. int[] intIdPelatihan = new int[strArrayPelatihan.length];
57.
58. for (int i = 0; i < strArrayPelatihan.length; i++) {
59.     intIdPelatihan[i] = Integer.parseInt(strArrayPelatihan[i]);
60. }
61.
62. List<String> listPelatihan = new ArrayList<>();
63. for (int i = 0; i < intIdPelatihan.length; i++) {
64.     RefPelatihan refPelatihan = refPelatihanRepo.findById(intIdPelatihan[i]).get();
65.     listPelatihan.add(refPelatihan.getNama());
66.     tbKriteriaPegawaiDTO.setListPelatihan(listPelatihan);
67. }
68.
69. tbKriteriaPegawaiDTO.setJumlahPengalaman(kriteriaPegawai.getJumlahPengalaman());
70.
71. tbKriteriaPegawaiDTO.setJumlahProjectOngoing(kriteriaPegawai.getJumlahProjectOngoing(
72. ));
73. String strArrPenguasaanStack = kriteriaPegawai.getPenguasaanStack();
74. strArrPenguasaanStack = strArrPenguasaanStack.substring(1,
75. strArrPenguasaanStack.length() - 1);
76. String[] strArray = strArrPenguasaanStack.split(", ");
77. int[] intIdStack = new int[strArray.length];
78.
79. List<String> listStack = new ArrayList<>();
80. for (int i = 0; i < strArray.length; i++) {
81.     intIdStack[i] = Integer.parseInt(strArray[i]);
82. }
83.
84. for (int i = 0; i < intIdStack.length; i++) {
85.     RefStack refStack = refStackRepo.findById(intIdStack[i]).get();
86.     listStack.add(refStack.getNama());
87.     tbKriteriaPegawaiDTO.setListStack(listStack);
88. }
89.
90. listTbKriteriaPegawaiDTO.add(tbKriteriaPegawaiDTO);
91. });
92. return listTbKriteriaPegawaiDTO;
93. }
94.
95. @Override
96. public MessageResponse getKriteriaPegawaiById(int id) {
97.
98.     if (!tbKriteriaPegawaiRepo.existsById(id)) {
99.         return new MessageResponse(0, "kriteria pegawai tidak ditemukan", null);

```

```

100.     }
101.
102.         return new MessageResponse(1, "Success",
tbKriteriaPegawaiRepo.findById(id).get());
103.     }
104.
105.     @Override
106.     public MessageResponse simpanKriteriaPegawai(KriteriaPegawaiDTO
kriteriaPegawaiDTO) {
107.         if
(tbKriteriaPegawaiRepo.findByIdPegawai(kriteriaPegawaiDTO.getIdPegawai()).isPresent())
{
108.             return new MessageResponse(0, "Kriteria Pegawai sudah di input", null);
109.         }
110.
111.
112.             Integer projectDeploy =
tbTimRepo.jumlahProjectByStatusProses(kriteriaPegawaiDTO.getIdPegawai(), 4);
113.             Integer projectOngoing =
tbTimRepo.jumlahProjectByStatusProses(kriteriaPegawaiDTO.getIdPegawai(), 2);
114.
115.         TbKriteriaPegawai tbKriteriaPegawai = new TbKriteriaPegawai();
116.         tbKriteriaPegawai.setIdPegawai(kriteriaPegawaiDTO.getIdPegawai());
117.
tbKriteriaPegawai.setKemampuanCoding(kriteriaPegawaiDTO.getKemampuanCoding());
118.         tbKriteriaPegawai.setJumlahPengalaman(projectDeploy);
119.         tbKriteriaPegawai.setJumlahProjectOngoing(projectOngoing);
120.
tbKriteriaPegawai.setJumlahPelatihan(Arrays.toString(kriteriaPegawaiDTO.getJumlahPelati
han()));
121.
tbKriteriaPegawai.setPenguasaanStack(Arrays.toString(kriteriaPegawaiDTO.getPenguasaanS
tack()));
122.
123.
124.         if
(tbKriteriaPegawaiMatrixRepo.findByIdPegawai(kriteriaPegawaiDTO.getIdPegawai()).isPre
sent()) {
125.             TbKriteriaPegawaiMatrix tbKriteriaPegawaiMatrix =
tbKriteriaPegawaiMatrixRepo.findByIdPegawai(kriteriaPegawaiDTO.getIdPegawai()).get();
126.
127.             //C1
128.             switch (tbKriteriaPegawai.getKemampuanCoding()) {
129.                 case 1:
130.                     tbKriteriaPegawaiMatrix.setKemampuanCoding(1);
131.                     break;
132.                 case 2:
133.                     tbKriteriaPegawaiMatrix.setKemampuanCoding(2);
134.                     break;
135.                 case 3:
136.                     tbKriteriaPegawaiMatrix.setKemampuanCoding(3);
137.                     break;
138.                 case 4:
139.                     tbKriteriaPegawaiMatrix.setKemampuanCoding(4);
140.                     break;

```

```

141.         case 5:
142.             tbKriteriaPegawaiMatrix.setKemampuanCoding(5);
143.             break;
144.         }
145.
146.         //C2
147.         switch (tbKriteriaPegawai.getJumlahProjectOngoing()) {
148.             case 0:
149.                 tbKriteriaPegawaiMatrix.setProjectOngoing(1);
150.                 break;
151.             case 1:
152.                 tbKriteriaPegawaiMatrix.setProjectOngoing(2);
153.                 break;
154.             case 2:
155.                 tbKriteriaPegawaiMatrix.setProjectOngoing(3);
156.                 break;
157.             case 3:
158.                 tbKriteriaPegawaiMatrix.setProjectOngoing(4);
159.                 break;
160.             default:
161.                 tbKriteriaPegawaiMatrix.setProjectOngoing(5);
162.                 break;
163.         }
164.
165.         //C3
166.         if (tbKriteriaPegawai.getJumlahPengalaman() <= 1) {
167.             tbKriteriaPegawaiMatrix.setPengalaman(1);
168.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 2) {
169.             tbKriteriaPegawaiMatrix.setPengalaman(2);
170.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 3) {
171.             tbKriteriaPegawaiMatrix.setPengalaman(3);
172.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 4) {
173.             tbKriteriaPegawaiMatrix.setPengalaman(4);
174.         } else {
175.             tbKriteriaPegawaiMatrix.setPengalaman(5);
176.         }
177.
178.
179.         //C4
180.         String strArrPelatihan = tbKriteriaPegawai.getJumlahPelatihan();
181.         strArrPelatihan = strArrPelatihan.substring(1, strArrPelatihan.length() - 1); //
Remove the brackets
182.         String[] strArrayPelatihan = strArrPelatihan.split(", "); // Split the string into an
array of strings
183.         int[] intIdPelatihan = new int[strArrayPelatihan.length]; // Create an int array of
the same length
184.
185.         // Parse the strings into integers and store them in the int array
186.         for (int i = 0; i < strArrayPelatihan.length; i++) {
187.             intIdPelatihan[i] = Integer.parseInt(strArrayPelatihan[i]);
188.         }
189.
190.         if (intIdPelatihan.length <= 1) {
191.             tbKriteriaPegawaiMatrix.setPelatihan(1);
192.         } else if (intIdPelatihan.length > 1 && intIdPelatihan.length <= 3) {

```

```

193.         tbKriteriaPegawaiMatrix.setPelatihan(2);
194.     } else if (intIdPelatihan.length > 3 && intIdPelatihan.length <= 5) {
195.         tbKriteriaPegawaiMatrix.setPelatihan(3);
196.     } else if (intIdPelatihan.length > 5 && intIdPelatihan.length <= 7) {
197.         tbKriteriaPegawaiMatrix.setPelatihan(4);
198.     } else if (intIdPelatihan.length > 7) {
199.         tbKriteriaPegawaiMatrix.setPelatihan(5);
200.     } else {
201.         return new MessageResponse(0, "Pilihan pelatihan tidak sesuai", null);
202.     }
203.
204.
205.     tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
206. } else {
207.     TbKriteriaPegawaiMatrix tbKriteriaPegawaiMatrix = new
TbKriteriaPegawaiMatrix();
208.     tbKriteriaPegawaiMatrix.setIdPegawai(kriteriaPegawaiDTO.getIdPegawai());
209.
210.     //C1
211.     switch (tbKriteriaPegawai.getKemampuanCoding()) {
212.         case 1:
213.             tbKriteriaPegawaiMatrix.setKemampuanCoding(1);
214.             break;
215.         case 2:
216.             tbKriteriaPegawaiMatrix.setKemampuanCoding(2);
217.             break;
218.         case 3:
219.             tbKriteriaPegawaiMatrix.setKemampuanCoding(3);
220.             break;
221.         case 4:
222.             tbKriteriaPegawaiMatrix.setKemampuanCoding(4);
223.             break;
224.         case 5:
225.             tbKriteriaPegawaiMatrix.setKemampuanCoding(5);
226.             break;
227.     }
228.
229.     //C2
230.     switch (tbKriteriaPegawai.getJumlahProjectOngoing()) {
231.         case 0:
232.             tbKriteriaPegawaiMatrix.setProjectOngoing(1);
233.             break;
234.         case 1:
235.             tbKriteriaPegawaiMatrix.setProjectOngoing(2);
236.             break;
237.         case 2:
238.             tbKriteriaPegawaiMatrix.setProjectOngoing(3);
239.             break;
240.         case 3:
241.             tbKriteriaPegawaiMatrix.setProjectOngoing(4);
242.             break;
243.         default:
244.             tbKriteriaPegawaiMatrix.setProjectOngoing(5);
245.             break;
246.     }

```

```

247.
248.         //C3
249.         if (tbKriteriaPegawai.getJumlahPengalaman() <= 1) {
250.             tbKriteriaPegawaiMatrix.setPengalaman(1);
251.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 2) {
252.             tbKriteriaPegawaiMatrix.setPengalaman(2);
253.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 3) {
254.             tbKriteriaPegawaiMatrix.setPengalaman(3);
255.         } else if (tbKriteriaPegawai.getJumlahPengalaman() == 4) {
256.             tbKriteriaPegawaiMatrix.setPengalaman(4);
257.         } else {
258.             tbKriteriaPegawaiMatrix.setPengalaman(5);
259.         }
260.
261.         //C4
262.         String strArrPelatihan = tbKriteriaPegawai.getJumlahPelatihan();
263.         strArrPelatihan = strArrPelatihan.substring(1, strArrPelatihan.length() - 1); //
Remove the brackets
264.         String[] strArrayPelatihan = strArrPelatihan.split(", "); // Split the string into an
array of strings
265.         int[] intIdPelatihan = new int[strArrayPelatihan.length]; // Create an int array of
the same length
266.
267.         // Parse the strings into integers and store them in the int array
268.         for (int i = 0; i < strArrayPelatihan.length; i++) {
269.             intIdPelatihan[i] = Integer.parseInt(strArrayPelatihan[i]);
270.         }
271.
272.         if (intIdPelatihan.length <= 1) {
273.             tbKriteriaPegawaiMatrix.setPelatihan(1);
274.         } else if (intIdPelatihan.length > 1 && intIdPelatihan.length <= 3) {
275.             tbKriteriaPegawaiMatrix.setPelatihan(2);
276.         } else if (intIdPelatihan.length > 3 && intIdPelatihan.length <= 5) {
277.             tbKriteriaPegawaiMatrix.setPelatihan(3);
278.         } else if (intIdPelatihan.length > 5 && intIdPelatihan.length <= 7) {
279.             tbKriteriaPegawaiMatrix.setPelatihan(4);
280.         } else if (intIdPelatihan.length > 7) {
281.             tbKriteriaPegawaiMatrix.setPelatihan(5);
282.         } else {
283.             return new MessageResponse(0, "Pilihan pelatihan tidak sesuai", null);
284.         }
285.
286.
287.         tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
288.     }
289.
290.     tbKriteriaPegawaiRepo.save(tbKriteriaPegawai);
291.
292.     return new MessageResponse(1, "Kriteria Pegawai berhasil disimpan",
tbKriteriaPegawai);
293. }
294.
295. @Override
296. public MessageResponse updateKriteriaPegawai(int id, KriteriaPegawaiDTO
kriteriaPegawaiDTO) {

```

```

297.                                     Integer    projectDeploy    =
tbTimRepo.jumlahProjectByStatusProses(kriteriaPegawaiDTO.getIdPegawai(), 4);
298.                                     Integer    projectOngoing    =
tbTimRepo.jumlahProjectByStatusProses(kriteriaPegawaiDTO.getIdPegawai(), 2);
299.
300.                                     TbKriteriaPegawai    tbKriteriaPegawai    =
tbKriteriaPegawaiRepo.findById(id).orElse(null);
301.     if (tbKriteriaPegawai == null) {
302.         return new MessageResponse(0, "Kriteria Pegawai tidak ditemukan", null);
303.     }
304.     tbKriteriaPegawai.setIdPegawai(kriteriaPegawaiDTO.getIdPegawai());
305.
tbKriteriaPegawai.setKemampuanCoding(kriteriaPegawaiDTO.getKemampuanCoding());
306.     tbKriteriaPegawai.setJumlahPengalaman(projectDeploy);
307.     tbKriteriaPegawai.setJumlahProjectOngoing(projectOngoing);
308.
tbKriteriaPegawai.setJumlahPelatihan(Arrays.toString(kriteriaPegawaiDTO.getJumlahPelati
han()));
309.
tbKriteriaPegawai.setPenguasaanStack(Arrays.toString(kriteriaPegawaiDTO.getPenguasaanS
tack()));
310.
311.
312.                                     if
(tbKriteriaPegawaiMatrixRepo.findByIdPegawai(kriteriaPegawaiDTO.getIdPegawai()).isPre
sent()) {
313.                                     TbKriteriaPegawaiMatrix    tbKriteriaPegawaiMatrix    =
tbKriteriaPegawaiMatrixRepo.findByIdPegawai(kriteriaPegawaiDTO.getIdPegawai()).get();
314.
315.         //C1
316.         switch (tbKriteriaPegawai.getKemampuanCoding()) {
317.             case 1:
318.                 tbKriteriaPegawaiMatrix.setKemampuanCoding(1);
319.                 break;
320.             case 2:
321.                 tbKriteriaPegawaiMatrix.setKemampuanCoding(2);
322.                 break;
323.             case 3:
324.                 tbKriteriaPegawaiMatrix.setKemampuanCoding(3);
325.                 break;
326.             case 4:
327.                 tbKriteriaPegawaiMatrix.setKemampuanCoding(4);
328.                 break;
329.             case 5:
330.                 tbKriteriaPegawaiMatrix.setKemampuanCoding(5);
331.                 break;
332.         }
333.
334.         //C2
335.         switch (tbKriteriaPegawai.getJumlahProjectOngoing()) {
336.             case 0:
337.                 tbKriteriaPegawaiMatrix.setProjectOngoing(1);
338.                 break;
339.             case 1:
340.                 tbKriteriaPegawaiMatrix.setProjectOngoing(2);

```

```

341.         break;
342.     case 2:
343.         tbKriteriaPegawaiMatrix.setProjectOngoing(3);
344.         break;
345.     case 3:
346.         tbKriteriaPegawaiMatrix.setProjectOngoing(4);
347.         break;
348.     default:
349.         tbKriteriaPegawaiMatrix.setProjectOngoing(5);
350.         break;
351. }
352.
353. //C3
354. if (tbKriteriaPegawai.getJumlahPengalaman() <= 1) {
355.     tbKriteriaPegawaiMatrix.setPengalaman(1);
356. } else if (tbKriteriaPegawai.getJumlahPengalaman() == 2) {
357.     tbKriteriaPegawaiMatrix.setPengalaman(2);
358. } else if (tbKriteriaPegawai.getJumlahPengalaman() == 3) {
359.     tbKriteriaPegawaiMatrix.setPengalaman(3);
360. } else if (tbKriteriaPegawai.getJumlahPengalaman() == 4) {
361.     tbKriteriaPegawaiMatrix.setPengalaman(4);
362. } else {
363.     tbKriteriaPegawaiMatrix.setPengalaman(5);
364. }
365.
366.
367. //C4
368. String strArrPelatihan = tbKriteriaPegawai.getJumlahPelatihan();
369. strArrPelatihan = strArrPelatihan.substring(1, strArrPelatihan.length() - 1); //
Remove the brackets
370. String[] strArrayPelatihan = strArrPelatihan.split(", "); // Split the string into an
array of strings
371. int[] intIdPelatihan = new int[strArrayPelatihan.length]; // Create an int array of
the same length
372.
373. // Parse the strings into integers and store them in the int array
374. for (int i = 0; i < strArrayPelatihan.length; i++) {
375.     intIdPelatihan[i] = Integer.parseInt(strArrayPelatihan[i]);
376. }
377.
378. if (intIdPelatihan.length <= 1) {
379.     tbKriteriaPegawaiMatrix.setPelatihan(1);
380. } else if (intIdPelatihan.length > 1 && intIdPelatihan.length <= 3) {
381.     tbKriteriaPegawaiMatrix.setPelatihan(2);
382. } else if (intIdPelatihan.length > 3 && intIdPelatihan.length <= 5) {
383.     tbKriteriaPegawaiMatrix.setPelatihan(3);
384. } else if (intIdPelatihan.length > 5 && intIdPelatihan.length <= 7) {
385.     tbKriteriaPegawaiMatrix.setPelatihan(4);
386. } else if (intIdPelatihan.length > 7) {
387.     tbKriteriaPegawaiMatrix.setPelatihan(5);
388. } else {
389.     return new MessageResponse(0, "Pilihan pelatihan tidak sesuai", null);
390. }
391.
392.

```

```

393.         //C5
394.         String strArrPenguasaanStack = tbKriteriaPegawai.getPenguasaanStack();
395.             strArrPenguasaanStack = strArrPenguasaanStack.substring(1,
strArrPenguasaanStack.length() - 1); // Remove the brackets
396.         String[] strArray = strArrPenguasaanStack.split(", "); // Split the string into an
array of strings
397.         int[] intIdStack = new int[strArray.length]; // Create an int array of the same length
398.
399.         // Parse the strings into integers and store them in the int array
400.         for (int i = 0; i < strArray.length; i++) {
401.             intIdStack[i] = Integer.parseInt(strArray[i]);
402.         }
403.
404.         int[] jenisStack = new int[intIdStack.length];
405.
406.         for (int i = 0; i < intIdStack.length; i++) {
407.             RefStack refStack = refStackRepo.findById(intIdStack[i]).get();
408.             jenisStack[i] = refStack.getJenis();
409.         }
410.
411.         int count1 = (int) Arrays.stream(jenisStack)
412.             .filter(value -> value == 1)
413.             .count();
414.
415.         int count2 = (int) Arrays.stream(jenisStack)
416.             .filter(value -> value == 2)
417.             .count();
418.
419.         int count3 = (int) Arrays.stream(jenisStack)
420.             .filter(value -> value == 3)
421.             .count();
422.
423.         if (count1 == 1 && count2 == 0 && count3 == 0) {
424.             tbKriteriaPegawaiMatrix.setStack(1);
425.         } else if (count1 >= 2 && count2 == 0 && count3 == 0) {
426.             tbKriteriaPegawaiMatrix.setStack(2);
427.         } else if (count1 >= 1 && count2 >= 1 && count3 == 0) {
428.             tbKriteriaPegawaiMatrix.setStack(3);
429.         } else if (count1 >= 1 && count2 >= 1 && count3 == 1) {
430.             tbKriteriaPegawaiMatrix.setStack(4);
431.         } else if (count1 >= 1 && count2 >= 1 && count3 >= 1) {
432.             tbKriteriaPegawaiMatrix.setStack(5);
433.         } else if (count1 >= 1 && count2 == 0 && count3 >= 1) {
434.             return new MessageResponse(0, "Wajib memilih minimal 1 stack programming
atau pilihan stack tidak sesuai", null);
435.         } else {
436.             return new MessageResponse(0, "Pilihan stack tidak sesuai", null);
437.         }
438.         tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
439.     } else {
440.         TbKriteriaPegawaiMatrix tbKriteriaPegawaiMatrix = new
TbKriteriaPegawaiMatrix();
441.         tbKriteriaPegawaiMatrix.setIdPegawai(kriteriaPegawaiDTO.getIdPegawai());
442.
443.         //C1

```



```

444.     switch (tbKriteriaPegawai.getKemampuanCoding()) {
445.         case 1:
446.             tbKriteriaPegawaiMatrix.setKemampuanCoding(1);
447.             break;
448.         case 2:
449.             tbKriteriaPegawaiMatrix.setKemampuanCoding(2);
450.             break;
451.         case 3:
452.             tbKriteriaPegawaiMatrix.setKemampuanCoding(3);
453.             break;
454.         case 4:
455.             tbKriteriaPegawaiMatrix.setKemampuanCoding(4);
456.             break;
457.         case 5:
458.             tbKriteriaPegawaiMatrix.setKemampuanCoding(5);
459.             break;
460.     }
461.
462.     //C2
463.     switch (tbKriteriaPegawai.getJumlahProjectOngoing()) {
464.         case 0:
465.             tbKriteriaPegawaiMatrix.setProjectOngoing(1);
466.             break;
467.         case 1:
468.             tbKriteriaPegawaiMatrix.setProjectOngoing(2);
469.             break;
470.         case 2:
471.             tbKriteriaPegawaiMatrix.setProjectOngoing(3);
472.             break;
473.         case 3:
474.             tbKriteriaPegawaiMatrix.setProjectOngoing(4);
475.             break;
476.         default:
477.             tbKriteriaPegawaiMatrix.setProjectOngoing(5);
478.             break;
479.     }
480.
481.     //C3
482.     if (tbKriteriaPegawai.getJumlahPengalaman() <= 1) {
483.         tbKriteriaPegawaiMatrix.setPengalaman(1);
484.     } else if (tbKriteriaPegawai.getJumlahPengalaman() == 2) {
485.         tbKriteriaPegawaiMatrix.setPengalaman(2);
486.     } else if (tbKriteriaPegawai.getJumlahPengalaman() == 3) {
487.         tbKriteriaPegawaiMatrix.setPengalaman(3);
488.     } else if (tbKriteriaPegawai.getJumlahPengalaman() == 4) {
489.         tbKriteriaPegawaiMatrix.setPengalaman(4);
490.     } else {
491.         tbKriteriaPegawaiMatrix.setPengalaman(5);
492.     }
493.
494.     //C4
495.     String strArrPelatihan = tbKriteriaPegawai.getJumlahPelatihan();
496.     strArrPelatihan = strArrPelatihan.substring(1, strArrPelatihan.length() - 1); //
Remove the brackets

```

```

497.         String[] strArrayPelatihan = strArrPelatihan.split(" "); // Split the string into an
array of strings
498.         int[] intIdPelatihan = new int[strArrayPelatihan.length]; // Create an int array of
the same length
499.
500.         // Parse the strings into integers and store them in the int array
501.         for (int i = 0; i < strArrayPelatihan.length; i++) {
502.             intIdPelatihan[i] = Integer.parseInt(strArrayPelatihan[i]);
503.         }
504.
505.         if (intIdPelatihan.length <= 1) {
506.             tbKriteriaPegawaiMatrix.setPelatihan(1);
507.         } else if (intIdPelatihan.length > 1 && intIdPelatihan.length <= 3) {
508.             tbKriteriaPegawaiMatrix.setPelatihan(2);
509.         } else if (intIdPelatihan.length > 3 && intIdPelatihan.length <= 5) {
510.             tbKriteriaPegawaiMatrix.setPelatihan(3);
511.         } else if (intIdPelatihan.length > 5 && intIdPelatihan.length <= 7) {
512.             tbKriteriaPegawaiMatrix.setPelatihan(4);
513.         } else if (intIdPelatihan.length > 7) {
514.             tbKriteriaPegawaiMatrix.setPelatihan(5);
515.         } else {
516.             return new MessageResponse(0, "Pilihan pelatihan tidak sesuai", null);
517.         }
518.
519.
520.         //C5
521.         String strArrPenguasaanStack = tbKriteriaPegawai.getPenguasaanStack();
522.         strArrPenguasaanStack = strArrPenguasaanStack.substring(1,
strArrPenguasaanStack.length() - 1); // Remove the brackets
523.         String[] strArray = strArrPenguasaanStack.split(" "); // Split the string into an
array of strings
524.         int[] intIdStack = new int[strArray.length]; // Create an int array of the same length
525.
526.         // Parse the strings into integers and store them in the int array
527.         for (int i = 0; i < strArray.length; i++) {
528.             intIdStack[i] = Integer.parseInt(strArray[i]);
529.         }
530.
531.         int[] jenisStack = new int[intIdStack.length];
532.
533.         for (int i = 0; i < intIdStack.length; i++) {
534.             RefStack refStack = refStackRepo.findById(intIdStack[i]).get();
535.             jenisStack[i] = refStack.getJenis();
536.         }
537.
538.         int count1 = (int) Arrays.stream(jenisStack)
539.             .filter(value -> value == 1)
540.             .count();
541.
542.         int count2 = (int) Arrays.stream(jenisStack)
543.             .filter(value -> value == 2)
544.             .count();
545.
546.         int count3 = (int) Arrays.stream(jenisStack)
547.             .filter(value -> value == 3)

```

```

548.         .count();
549.
550.         if (count1 == 1 && count2 == 0 && count3 == 0) {
551.             tbKriteriaPegawaiMatrix.setStack(1);
552.         } else if (count1 >= 2 && count2 == 0 && count3 == 0) {
553.             tbKriteriaPegawaiMatrix.setStack(2);
554.         } else if (count1 >= 1 && count2 >= 1 && count3 == 0) {
555.             tbKriteriaPegawaiMatrix.setStack(3);
556.         } else if (count1 >= 1 && count2 >= 1 && count3 == 1) {
557.             tbKriteriaPegawaiMatrix.setStack(4);
558.         } else if (count1 >= 1 && count2 >= 1 && count3 >= 1) {
559.             tbKriteriaPegawaiMatrix.setStack(5);
560.         } else if (count1 >= 1 && count2 == 0 && count3 >= 1) {
561.             return new MessageResponse(0, "Wajib memilih minimal 1 stack programming
atau pilihan stack tidak sesuai", null);
562.         } else {
563.             return new MessageResponse(0, "Pilihan stack tidak sesuai", null);
564.         }
565.         tbKriteriaPegawaiMatrixRepo.save(tbKriteriaPegawaiMatrix);
566.     }
567.
568.     tbKriteriaPegawaiRepo.save(tbKriteriaPegawai);
569.
570.     return new MessageResponse(1, "Kriteria Pegawai berhasil diupdate",
tbKriteriaPegawai);
571. }
572.
573. @Override
574. public MessageResponse hapusKriteriaPegawai(int id) {
575.     TbKriteriaPegawai tbKriteriaPegawai = (TbKriteriaPegawai)
getKriteriaPegawaiById(id).getData();
576.
577.     if (!tbKriteriaPegawaiRepo.existsById(id)) {
578.         return new MessageResponse(0, "Kriteria Pegawai tidak ditemukan", null);
579.     } else {
580.         tbKriteriaPegawaiRepo.deleteById(id);
581.
582.         if
(tbKriteriaPegawaiMatrixRepo.findByIdPegawai(tbKriteriaPegawai.getIdPegawai()).isPrese
nt()) {
583.             tbKriteriaPegawaiMatrixRepo.deleteById(tbKriteriaPegawaiMatrixRepo.findByIdPegawai(tb
KriteriaPegawai.getIdPegawai()).get().getId());
584.         } else {
585.             return new MessageResponse(0, "Kriteria Pegawai Matrix tidak ditemukan",
null);
586.         }
587.
588.         return new MessageResponse(1, "Kriteria Pegawai berhasil dihapus", null);
589.     }
590. }
591. }
592.

```

C. Implementasi Algoritma SAW

```
1. package com.skripsi.skripsi.utility;
2.
3. import com.skripsi.skripsi.entity.TbKriteriaPegawaiMatrix;
4. import com.skripsi.skripsi.entity.TbPembobotan;
5. import com.skripsi.skripsi.dto.TbPerangkinganDTO;
6. import com.skripsi.skripsi.repository.TbPembobotanRepo;
7. import org.springframework.beans.factory.annotation.Autowired;
8. import org.springframework.stereotype.Component;
9.
10. import java.time.LocalDate;
11. import java.util.ArrayList;
12. import java.util.List;
13.
14. @Component
15. public class SAWUtil {
16.     private final TbPembobotanRepo tbPembobotanRepo;
17.
18.     @Autowired
19.     public SAWUtil(TbPembobotanRepo tbPembobotanRepo) {
20.         this.tbPembobotanRepo = tbPembobotanRepo;
21.     }
22.
23.
24.     public MessageResponse metodeSAW(List<TbKriteriaPegawaiMatrix>
tbKriteriaPegawaiMatrixList) {
25.
26.         List<TbPembobotan> tbPembobotanList = tbPembobotanRepo.findAll();
27.         List<TbPerangkinganDTO> tbPerangkinganDTOList = new ArrayList<>();
28.
29.         // Mengambil nilai bobot dari tbPembobotan = W
30.         double[] W = new double[tbPembobotanList.size()];
31.         int[] idPegawai = new int[tbKriteriaPegawaiMatrixList.size()];
32.         for (int i = 0; i < tbPembobotanList.size(); i++) {
33.             W[i] = tbPembobotanList.get(i).getBobot();
34.         }
35.
36.
37.         //mapping data kriteria pegawai matrix ke array 2D
38.         double[][] X = new double[tbKriteriaPegawaiMatrixList.size()][tbPembobotanList.size()];
39.         for (int i = 0; i < tbKriteriaPegawaiMatrixList.size(); i++) {
40.             idPegawai[i] = tbKriteriaPegawaiMatrixList.get(i).getIdPegawai();
41.
42.             for (int j = 0; j < tbPembobotanList.size(); j++) {
43.                 if (j == 0) {
44.                     X[i][j] = tbKriteriaPegawaiMatrixList.get(i).getKemampuanCoding(); // c1=benefit
45.                 } else if (j == 1) {
46.                     X[i][j] = tbKriteriaPegawaiMatrixList.get(i).getProjectOngoing(); // c2=cost
47.                 } else if (j == 2) {
48.                     X[i][j] = tbKriteriaPegawaiMatrixList.get(i).getPengalaman(); // c3=benefit
49.                 } else if (j == 3) {
50.                     X[i][j] = tbKriteriaPegawaiMatrixList.get(i).getPelatihan(); // c4=benefit
51.                 } else if (j == 4) {
52.                     X[i][j] = tbKriteriaPegawaiMatrixList.get(i).getStack(); // c5=benefit
53.                 }

```

```

54.     }
55.     }
56. }
57.
58.
59. //matrix normalisasi = R
60. double[][] R = new double[X.length][X[0].length];
61. for (int j = 0; j < X[0].length; j++) {
62.     double max = X[0][j];
63.     double min = X[0][1];
64.     for (int i = 1; i < X.length; i++) {
65.         if (X[i][j] > max) {
66.             max = X[i][j];
67.         }
68.         if (X[i][1] < min) {
69.             min = X[i][1];
70.         }
71.     }
72.     for (int i = 0; i < X.length; i++) {
73.         if (j == 1) {
74.             R[i][j] = min / X[i][j];
75.         } else {
76.             R[i][j] = X[i][j] / max;
77.         }
78.     }
79. }
80.
81. //hitung nilai V = R * W (nila preferensi tiap alternatif)
82. double[] V = new double[R.length];
83. for (int i = 0; i < R.length; i++) {
84.     for (int j = 0; j < R[i].length; j++) {
85.         V[i] += R[i][j] * W[j];
86.     }
87.
88.     TbPerangkinganDTO tbPerangkinganDTO = new TbPerangkinganDTO();
89.     tbPerangkinganDTO.setIdPegawai(idPegawai[i]);
90.     tbPerangkinganDTO.setPerhitunganAlternatif(V[i]);
91.     tbPerangkinganDTO.setTanggalPerangkingan(LocalDate.now());
92.     tbPerangkinganDTO.setTbPegawai(tbKriteriaPegawaiMatrixList.get(i).getTbPegawai());
93.     tbPerangkinganDTOList.add(tbPerangkinganDTO);
94. }
95.
96. return new MessageResponse(1, "Perhitungan SAW Berhasil", tbPerangkinganDTOList);
97.
98. }
99.
100.
101. }
102.

```