

**PERANGKINGAN PEMILIHAN SKILL HERO PADA GAME “PEDJOEANG”  
MENGUNAKAN METODE TECHNIQUE FOR ORDER PERFORMANCE  
BY SIMILARITY TO IDEAL SOLUTION (TOPSIS) BERBASIS ORDERED  
WEIGHTED AVERAGING (OWA)**

**SKRIPSI**

Oleh:  
**MUHAMMAD ZIDAN**  
**NIM. 200605110051**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

**PERANGKINGAN PEMILIHAN SKILL HERO PADA GAME “PEDJOEANG”  
MENGUNAKAN METODE TECHNIQUE FOR ORDER PERFORMANCE  
BY SIMILARITY TO IDEAL SOLUTION (TOPSIS) BERBASIS ORDERED  
WEIGHTED AVERAGING (OWA)**

**SKRIPSI**

Diajukan Kepada:  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:  
**MUHAMMAD ZIDAN**  
**NIM. 200605110051**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2024**

## HALAMAN PERSETUJUAN

### PERANGKINGAN PEMILIHAN SKILL HERO PADA GAME "PEDJOEANG" MENGUNAKAN METODE TECHNIQUE FOR ORDER PERFORMANCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS) BERBASIS ORDERED WEIGHTED AVERAGING (OWA)

#### SKRIPSI

Oleh:  
**MUHAMMAD ZIDAN**  
NIM. 200605110051

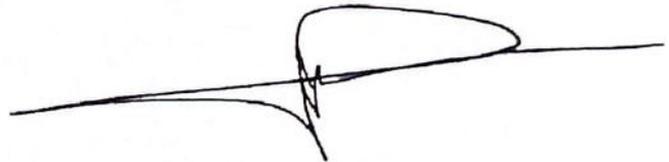
Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: 2 Mei 2024

Pembimbing I



Dr. Fresy Nugroho, M.T  
NIP. 19710722 201101 1 001

Pembimbing II



Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

## HALAMAN PENGESAHAN

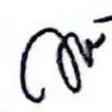
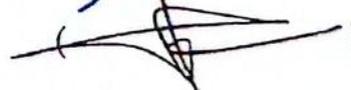
### PERANGKINGAN PEMILIHAN SKILL HERO PADA GAME "PEDJOEANG" MENGUNAKAN METODE TECHNIQUE FOR ORDER PERFORMANCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS) BERBASIS ORDERED WEIGHTED AVERAGING (OWA)

#### SKRIPSI

Oleh:  
**MUHAMMAD ZIDAN**  
NIM. 200605110051

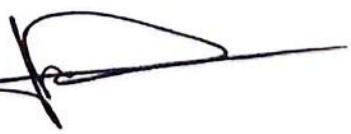
Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Tanggal: 7 Mei 2024

#### Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. Muhammad Faisal, M.T</u> NIP. 19740510 200501 1 007	(  )
Anggota Penguji I	: <u>Ahmad Fahmi Karami, M.Kom</u> NIP. 19870909 202012 1 001	(  )
Anggota Penguji II	: <u>Dr. Fresy Nugroho, M.T</u> NIP. 19710722 201101 1 001	(  )
Anggota Penguji III	: <u>Dr. Fachrul Kurniawan, M.MT, IPM</u> NIP. 19771020 200912 1 001	(  )

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
Dr. Fachrul Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Zidan

NIM : 200605110051

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Perangkingan Pemilihan *Skill Hero* Pada Game "PEDJOEANG" Menggunakan Metode *Technique for Order Performance by Similarity to Ideal Solution* (TOPSIS) Berbasis *Ordered Weighted Averaging* (OWA)

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 2 Mei 2024

Yang membuat pernyataan,



Muhammad Zidan  
NIM.200605110051

## HALAMAN MOTTO

” إِنَّ مَعَ الْعُسْرِ يُسْرًا ”

## **HALAMAN PERSEMBAHAN**

Alhamdulillah Rabbil Aalamin, puji syukur kepada Allah SWT. Terimakasih atas karunia-Mu yang telah memberikan kemudahan dan kelancaran sehingga skripsi ini dapat terselesaikan dengan baik.

Halaman persembahan ini ditujukan sebagai ungkapan terimakasih terutama kepada kedua orang tua penulis, serta keluarga, seluruh dosen, dan teman-teman yang telah mendoakan dan memberikan dukungan penuh selama perjuangan menempuh pendidikan.

## KATA PENGANTAR

*Assalamualaikum Wr. Wb*

Puji dan syukur penulis panjatkan atas ke hadirat Allah SWT, Tuhan Semesta Alam, yang telah melimpahkan Rahmat, hidayah, dan karunia-Nya yang telah memudahkan penulis dalam menyelesaikan skripsi yang berjudul “Perangkingan Pemilihan Skill Hero Pada Game “PEDJOEANG” Menggunakan Metode Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) Berbasis Ordered Weighted Averaging (OWA)” dengan baik. Shalawat dan salam senantiasa terlimpah kepada Nabi Muhammad Sallallahu ‘Alaihi wa Sallam. Dan semoga kita semua mendapat syafaatnya di hari kiamat nanti, Aamiin.

Penulis mengucapkan rasa syukur dan terima kasih kepada banyak pihak yang selalu memberikan semangat, bantuan dan motivasi kepada penulis. Untuk itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Fresy Nugroho, M.T., selaku Dosen Pembimbing I yang telah memberikan banyak bimbingan dan bantuan selama penulisan skripsi ini.
5. Dr. Fachrul Kurniawan, M.MT., IPM., selaku Dosen Pembimbing II yang telah memberikan bimbingan dalam penulisan skripsi ini.
6. Dr. Muhammad Faisal, M.T., selaku Ketua Penguji yang telah memberikan masukan dan dukungan dalam menyelesaikan skripsi ini.
7. Ahmad Fahmi Karami, M.Kom., selaku Dosen Penguji I yang telah menguji serta memberikan masukan dalam menyelesaikan skripsi ini.
8. Nia Faricha, S.Si selaku Admin Program Studi Teknik Informatika yang dengan sabar membantu, memberikan arahan, informasi terkait perkuliahan.

9. Seluruh Dosen dan Jajaran Staf Program Studi Teknik Informatika yang telah memberikan banyak bantuan dalam skripsi ini.
10. Kedua orang tua, Bapak Saifudin Zuhri dan Ibu Erni Dwi Sulistyowati, serta keluarga yang telah memberikan dukungan, bantuan, serta doa yang tiada henti kepada penulis dalam menyelesaikan skripsi ini.
11. Dewan asatidz Madrasah Diniyah Masjid Agung An-nuur Kota Batu, teman-teman “The Ninth”, “Enter New Subject” dan teman-teman Angkatan 2020 Teknik Informatika “INTEGER” yang telah memberikan banyak bantuan baik material maupun ilmu, semangat, serta motivasi dalam menyelesaikan penelitian skripsi ini.
12. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan kontribusi, saran, dan dukungan dalam penyusunan penulisan skripsi ini.

Malang, 2 Mei 2024

Penulis

## DAFTAR ISI

<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>v</b>
<b>HALAMAN MOTTO</b> .....	<b>vi</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vii</b>
<b>KATA PENGANTAR</b> .....	<b>viii</b>
<b>DAFTAR ISI</b> .....	<b>x</b>
<b>DAFTAR GAMBAR</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>ABSTRAK</b> .....	<b>xiv</b>
<b>ABSTRACT</b> .....	<b>xv</b>
البحث مستخلص.....	<b>xvi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
<b>BAB II STUDI PUSTAKA</b> .....	<b>7</b>
2.1 Penelitian Terdahulu.....	7
2.2 Role Playing Game.....	7
2.3 Decision Support System.....	9
2.4 <i>Technique for Order Preference by Similarity to Ideal Solution</i> (TOPSIS).....	10
2.5 <i>Ordered Weighted Averaging</i> (OWA).....	12
<b>BAB III DESAIN DAN IMPLEMENTASI</b> .....	<b>14</b>
3.1 Analisis dan Perancangan <i>Turn-Based</i> RPG.....	14
3.1.1 Analisis <i>Turn-Based</i> RPG.....	14
3.1.2 Analisis dan Perancangan <i>Turn-Based</i> RPG.....	15
3.1.3 Rancangan Antarmuka.....	17
3.2 Desain Sistem.....	20
3.3 Rancangan Perhitungan TOPSIS berbasis OWA.....	22
3.3.1 Alternatif.....	22
3.3.2 Kriteria.....	23
3.3.2.1 Menentukan Matriks Keputusan.....	24
3.3.2.2 Normalisasi Matriks Keputusan.....	25
3.3.2.3 Perhitungan TOPSIS berbasis OWA.....	26
3.3.2.4 Perankingan.....	28
3.4 Implementasi Perhitungan TOPSIS-OWA.....	29
3.5 Desain Pengujian Sistem.....	31
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>33</b>
4.1 Implementasi Sistem.....	33
4.1.1 Implementasi Perhitungan Metode.....	33
4.1.1.1 Implementasi Perhitungan TOPSIS.....	33

4.1.1.2 Implementasi Perhitungan TOPSIS-OWA.....	37
4.2 Pengujian Sistem .....	41
4.2.1 Uji Coba Game .....	42
4.2.2 Hasil Uji Coba .....	45
4.3 Integrasi dengan Islam.....	50
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>54</b>
5.1 Kesimpulan.....	54
5.2 Saran .....	54
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 3. 1 Desain FSM Pedjoeang .....	16
Gambar 3. 2 Tampilan Mainmenu .....	18
Gambar 3. 3 Tampilan Pengaturan.....	18
Gambar 3. 4 Tampilan Pilihan Pedjoeang.....	19
Gambar 3. 5 <i>Gameplay</i> .....	20
Gambar 3. 6 <i>BattleScene</i> .....	20
Gambar 3. 7 Desain Sistem.....	21
Gambar 4. 1 Hasil Perhitungan Metode dalam Unity .....	41
Gambar 4. 2 Tombol Hint .....	42
Gambar 4. 3 Sistem Rekomendasi .....	43
Gambar 4. 4 Battlescene Prajurit Hijau.....	43
Gambar 4. 5 Battlescene Prajurit Merah .....	44
Gambar 4. 6 Battlescene Jendral de Kock .....	45

## DAFTAR TABEL

Tabel 2. 1 Perbandingan Penelitian.....	7
Tabel 3. 1 Alternatif .....	22
Tabel 3. 2 Kriteria .....	23
Tabel 3. 3 Matriks Keputusan .....	25
Tabel 3. 4 Normalisasi Matriks Keputusan.....	26
Tabel 3. 5 Hasil Perhitungan TOPSIS berbasis OWA.....	27
Tabel 3. 6 Perankingan.....	28
Tabel 4. 1 Hasil Sistem Rekomendasi Setiap Scene .....	43
Tabel 4. 2 Matriks Keputusan Battlescene Prajurit Hijau .....	45
Tabel 4. 3 Bobot Battlescene Prajurit Hijau.....	46
Tabel 4. 4 Perbandingan Metode Battlescene Prajurit Hijau .....	46
Tabel 4. 5 Matriks Keputusan <i>Battlescene</i> Prajurit Merah .....	46
Tabel 4. 6 Bobot <i>Battlescene</i> Prajurit Merah .....	46
Tabel 4. 7 Perbandingan Metode <i>Battlescene</i> Prajurit Merah.....	47
Tabel 4. 8 Matriks Keputusan <i>Battlescene</i> Jendral de Kock.....	47
Tabel 4. 9 Bobot <i>Battlescene</i> Jendral de Kock .....	47
Tabel 4. 10 Perbandingan Metode <i>Battlescene</i> Jendral de Kock.....	48

## ABSTRAK

Zidan, Muhammad. 2024. *Perangkingan Pemilihan Skill Hero Pada Game “PEDJOEANG” Menggunakan Metode Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) Berbasis Ordered Weighted Averaging (OWA)*. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Fresy Nugroho, M.T (II) Dr. Fachrul Kurniawan, M.MT, IPM.

**Kata Kunci:** Skill, Technique for Order Performance by Similarity to Ideal Solution, Ordered Weighted Averaging, Game, Pedjoeang.

Sistem rekomendasi dapat menjadi suatu hal yang penting dalam sebuah *game*, untuk membantu *player* dalam menentukan sebuah keputusan. Dalam penelitian ini sistem rekomendasi dikembangkan dengan metode *Technique for Order Performance by Similarity to Ideal Solution* (TOPSIS) berbasis *Ordered Weighted Averaging* (OWA). Sistem tersebut akan dimasukkan dalam *game* “PEDJOEANG” yakni pada pemilihan *skill* untuk membantu *player* dalam mengalahkan musuhnya. Nantinya hasil perhitungan dari metode TOPSIS-OWA akan dibandingkan dengan hasil perhitungan metode TOPSIS. Didapatkan bahwa hasil perhitungan metode TOPSIS-OWA mendapatkan hasil yang lebih efektif dibanding metode TOPSIS dalam perankingan pemilihan *skill* dalam *game* “PEDJOEANG”.

## ABSTRACT

Zidan, Muhammad. 2024. *Ranking Skill Selection for Hero in the Game "PEDJOEANG" Using the Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) Method Based on Ordered Weighted Averaging (OWA)*. Undergraduate Thesis, Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University Malang. Promotor: (I) Dr. Fresy Nugroho, M.T (II) Dr. Fachrul Kurniawan, M.MT, IPM.

**Keywords:** Skill, Technique for Order Performance by Similarity to Ideal Solution, Ordered Weighted Averaging, Game, Pedjoeang.

Recommendation system plays an important part in a game to help players to make decision. In the research, the recommendation system was develop using Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) method based on Ordered Weighted Averaging (OWA). The system will be installed in the game of "PEDJOEANG" to help players to choose skill to beat their enemies. The calculation result of TOPSIS-OWA will be compared with that of TOPSIS method. The research result shows that the calculation result of TOPSIS-OWA method is more effective than that of TOPSIS method in skill-choosing ranking in the game of "PEDJOEANG".

## البحث مستخلص

زيدان، محمد. 2024. ترتيب اختيار مهارة البطل في لعبة "PEDJOEANG" باستخدام تقنية أداء الطلب عن طريق التشابه مع الحل المثالي (TOPSIS) بناء على المتوسط المرجح المطلوب (OWA). رسالة الماجستير. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. فريسي نوغروهو، الماجستير. المشرف الثاني: د. فخر الكورنياوان، الماجستير.

**الكلمات الرئيسية:** مهارة، تقنية أداء الطلب عن طريق التشابه مع الحل المثالي، المتوسط المرجح المطلوب، لعبة، pedjoeang.

يمكن أن يكون نظام التوصية شيئًا مهمًا في اللعبة، لمساعدة اللاعب على اتخاذ القرار. في هذا البحث، تم تطوير نظام التوصية باستخدام تقنية أداء الطلب عن طريق التشابه مع الحل المثالي (TOPSIS) بناء على المتوسط المرجح المطلوب (OWA). سيتم تضمين النظام في لعبة "PEDJOEANG" عند اختيار المهارة لمساعدة اللاعب على هزيمة أعدائه. في وقت لاحق، سيتم مقارنة نتائج حساب طريقة TOPSIS-OWA بنتائج حساب طريقة TOPSIS. وجد أن نتائج حساب طريقة TOPSIS-OWA حصلت على نتائج أكثر فعالية من طريقة TOPSIS في ترتيب اختيار المهارة في لعبة "PEDJOEANG".

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Game* kini tidak lagi hanya dianggap sebagai hiburan semata, tetapi juga sebagai bentuk ekspresi budaya, karya seni, dan media sosial. *Game* menyediakan wadah bagi *player* untuk berinteraksi, berkolaborasi, dan mengejar prestasi dalam sebuah dunia virtual yang dinamis (Qomariah et al., 2020). Hal ini telah mengubah bagaimana masyarakat modern di Indonesia mengkonsumsi hiburan dan menjadikan *game* sebagai bagian tak terpisahkan dari kehidupan sehari-hari.

Produksi *game* lokal terus berkembang. Banyak pengembang *game* Indonesia telah membuat *game* yang diakui secara nasional dan internasional. Ini termasuk *game* yang mengangkat budaya dan mitologi Indonesia, serta *game* dengan cerita dan karakter lokal yang menarik perhatian *playernya*. Secara tidak langsung pengembangan ini menciptakan lapangan kerja di industri *game* dan meningkatkan kebanggaan nasional terhadap industri kreatif (Bahtiar et al., 2019).

*Game* dapat dikatakan menyenangkan dan menarik bagi *player* apabila *game* tersebut dapat memberikan pengalaman yang menantang dan memunculkan rasa penasaran sehingga *player* akan terus memainkan *game* tersebut sampai akhir. (Arif & Yulianto, 2014)

Salah satu *genre game* yang populer di pasar internasional adalah RPG (*Role Playing Game*). RPG merupakan *genre game* di mana *player* mengambil peran atau karakter dalam cerita atau dunia fiksi dan berinteraksi dengan lingkungan serta

karakter lain sesuai dengan peran yang mereka pilih. *Player* biasanya memiliki kendali atas keputusan dan tindakan karakter yang mereka perankan (Kaban et al., 2021). Meskipun pada umumnya game RPG membutuhkan strategi untuk memainkannya, *game* RPG tetap disukai oleh seluruh kalangan baik anak-anak, remaja maupun orang dewasa (Romadhona & Yundra, 2018).

Pada penelitian sebelumnya, sebuah *game* RPG bernama “Grand Line” yang dirancang oleh (Muttakin et al., 2020), dengan sub tema *Turn-based* bercerita tentang seorang karakter pahlawan yang berjuang untuk mengalahkan musuhnya, dengan *skill* yang dimilikinya. *Player* harus bisa memilih mode menyerang yang cocok, agar bisa mengalahkan musuh. Namun, pada game ini belum diterapkan sistem rekomendasi untuk memilih mode menyerang yang efektif untuk melawan musuh, jadi pada game tersebut *player* harus sedikit memikirkan strategi yang baik agar bisa mengalahkan musuhnya.

Untuk membantu *player* mengalahkan musuh pada game RPG Turn-Based, sistem rekomendasi dapat menjadi suatu hal yang penting, karena sistem tersebut dapat menyarankan penggunaan *skill* karakter secara efektif sesuai dengan musuh yang dihadapi, sehingga *player* dapat lebih mudah memilih *skill* dengan tepat dan terhindar dari pilihan yang salah. Hal ini sejalan dengan prinsip kepedulian terhadap sesama makhluk Allah. Firman Allah dalam Surah Al-Baqarah [2]: 216 yang berbunyi:

وَعَسَىٰ أَنْ تَكْرَهُوا شَيْئًا وَهُوَ خَيْرٌ لَّكُمْ وَعَسَىٰ أَنْ تُحِبُّوا شَيْئًا وَهُوَ شَرٌّ لَّكُمْ وَاللَّهُ يَعْلَمُ وَأَنْتُمْ لَا تَعْلَمُونَ

“Mungkin kamu membenci sesuatu, padahal ia amat baik bagimu, dan mungkin (pula) kamu menyukai sesuatu, padahal ia amat buruk bagimu; Allah mengetahui, sedangkan kamu tidak mengetahui.” (QS. Al-Baqarah:216)

Dalam tafsir Kemenag RI, disampaikan bahwa ada suatu hal yang mungkin dianggap buruk tapi hal tersebut malah yang lebih baik, sebaliknya terdapat hal yang mungkin dianggap baik, namun ternyata hal tersebut buruk (Lajnah Pentashihan Mushaf Al-Qur'an, 2022a). Dengan demikian, sistem rekomendasi untuk memilih *skill* yang tepat dalam game RPG Turn-Based bukan hanya tentang mengoptimalkan kinerja dalam *game*, tetapi juga tentang menerapkan prinsip-prinsip moral dan etika yang diajarkan oleh agama. Sistem ini diharapkan dapat memberikan saran *skill* terbaik yang dapat digunakan *player*, sehingga *player* dapat mengalahkan musuhnya dengan mudah. Sistem ini memberikan panduan berdasarkan data dan algoritma. Dengan kata lain, sistem rekomendasi bukan hanya alat untuk meningkatkan keterampilan dalam *game*, tetapi juga peluang untuk menghormati prinsip-prinsip moral yang mendalam.

Penulis mengembangkan suatu sistem rekomendasi yang mampu menghitung dan menganalisis berbagai alternatif dalam pemilihan *skill* karakter yang bervariasi dengan menerapkan metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) yang berbasis OWA (*Ordered Weighted Averaging*). Perbedaan mendasar antara TOPSIS dan TOPSIS berbasis OWA adalah terletak pada pendekatan penentuan bobot atribut. Dalam metode TOPSIS, bobot atribut bersifat tetap dan telah ditentukan sebelumnya oleh pengambil keputusan, sedangkan dalam TOPSIS berbasis OWA, bobot atribut bersifat fleksibel, disusun berdasarkan preferensi pengambil keputusan melalui fungsi distribusi yang menggambarkan struktur preferensial. (Liu & Wang, 2020).

Metode OWA dapat digunakan untuk menggabungkan hasil dari perhitungan kriteria berdasarkan bobot yang telah ditentukan. OWA memungkinkan penulis untuk mengatur tingkat ketidakpastian dalam pemilihan *skill* karakter, sehingga sistem rekomendasi dapat memberikan rekomendasi yang lebih fleksibel dan sesuai dengan preferensi *player*, terutama ketika situasi dalam *game* berubah (Ghasemkhani et al., 2020).

Setelah analisis yang mendalam dengan penerapan TOPSIS berbasis OWA selesai, sistem memberikan rekomendasi kepada *player*, menyajikan daftar *skill* yang diurutkan berdasarkan relevansi atau nilai hasil perhitungan yang telah disesuaikan dengan bobot kriteria. Dengan demikian, sistem rekomendasi ini bertujuan membantu *player* membuat keputusan yang lebih tepat, dan disesuaikan dengan preferensi dalam memilih *skill* karakter dari berbagai alternatif yang tersedia dalam *game*. Pendekatan ini meningkatkan pengalaman bermain dan mendukung *player* dalam mengambil keputusan yang lebih baik dalam konteks *game*, dengan memanfaatkan teknik TOPSIS berbasis OWA untuk memberikan rekomendasi *skill* karakter yang lebih efektif.

Pada penelitian sebelumnya, metode OWA digunakan oleh (Simangunsong & Hamdani, 2020) untuk menentukan desa terbaik pada pelaksanaan perlombaan desa dan kelurahan, tujuan dari pada kegiatan ini adalah mengevaluasi dan menilai perkembangan desa dan kelurahan menuju desa yang berkembang dan maju. beberapa aspek yang digunakan sebagai acuan dalam penilaian adalah kesehatan, Kebersihan desa, Pendidikan, Ekonomi, Keamanan dan Ketertiban, Peranan PKK, Administrasi Pemerintah Desa dipadukan menjadi satu kesatuan dalam kompetisi

untuk menjadi salah satu bagian dari desa terbaik pada tingkatan tertentu. Metode OWA dipilih karena metode ini mampu memberikan penilaian berdasarkan rata-rata perhitungan dari nilai yang di inputkan.

Maka, dalam penelitian ini, metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) berbasis OWA digunakan sebagai landasan untuk mengembangkan sistem rekomendasi dalam konteks pemilihan *skill* karakter dalam *game* "Pedjoeang." Penggunaan TOPSIS berbasis OWA menghadirkan pendekatan yang matematis dan fleksibel, di mana sistem dapat mempertimbangkan berbagai faktor atau kriteria yang relevan dalam proses pemilihan *skill* karakter. Metode ini memungkinkan sistem untuk memberikan rekomendasi *skill* karakter yang lebih tepat sesuai dengan preferensi *player*, dengan memperhitungkan bobot kriteria yang telah ditentukan.

## 1.2 Rumusan Masalah

Apakah sistem yang dibangun menggunakan metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) berbasis OWA (*Ordered Weighted Averaging*) lebih baik daripada sistem yang dibangun dengan metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) dan pemilihan *skill* karakter pada *game* "Pedjoeang".

## 1.3 Batasan Masalah

Adapun pembatasan masalah dalam penelitian ini sebagai berikut.

1. Variabel kriteria yang digunakan antara lain seperti *Enemy Health Point*, *Skill Effectiveness*, *Blue Attack*, *Red Attack*, *Green Attack*, *Skill Energy Cost*.

2. Variabel alternatif yang akan diteliti adalah *skill* karakter yang akan dievaluasi dan dibandingkan berdasarkan kriteria yang telah ditentukan, Alternatif *skill* karakter yang akan diteliti dan dibandingkan adalah A1 hingga A6.

#### **1.4 Tujuan Penelitian**

Tujuan utama dari penelitian ini adalah mengembangkan sebuah sistem rekomendasi yang menggunakan metode TOPSIS berbasis OWA untuk memudahkan *player* dalam pemilihan *skill* karakter yang optimal dalam *game* “Pedjoeang”.

#### **1.5 Manfaat Penelitian**

Memberikan rekomendasi *skill* karakter yang optimal bagi *player* “Pedjoeang”, menghemat waktu, serta memberikan fleksibilitas dalam pemilihan *skill*.

## BAB II

### STUDI PUSTAKA

#### 2.1 Penelitian Terdahulu

Berikut merupakan beberapa penelitian terdahulu yang berkaitan dengan penelitian yang saat ini dilakukan oleh penulis

Tabel 2. 1 Perbandingan Penelitian

No	Peneliti	Judul	Persamaan	Perbedaan
1.	(Prasetyo et al., 2022)	Metode K-Means Berbasis <i>Ordered Weighted Averaging</i> (OWA) pada Data Potensi Desa untuk Penentuan Status Desa	Penggunaan Metode OWA	Dikombinasikan dengan Metode K-Means
2.	(H. Ilyas et al., 2022)	<i>Analysis Of Learning Web Application Testing With Weighting Product And Ordered Weighting Averaging On Group Decision Maker</i>	Penggunaan Metode OWA	Dikombinasikan dengan Metode WP
3.	(Zhao et al., 2022)	<i>Analysis of Ranking Consistency in Linguistic Multiple Attribute Decision Making: The Roles of Granularity and Decision Rules</i>	Penggunaan Metode OWA	Membandingkan dengan Metode lain
4.	(Mutmainah & Yunita, 2021)	Penerapan Metode TOPSIS Dalam Pemilihan Jasa Ekspedisi	Penggunaan Metode TOPSIS	Tidak diimplementasikan pada <i>Game</i>
5.	(Setiawansyah, 2022)	Sistem Pendukung Keputusan Rekomendasi Tempat Wisata Menggunakan Metode TOPSIS	Penggunaan Metode TOPSIS	Tidak dikombinasikan dengan Metode OWA

#### 2.2 Role Playing Game

*Game* adalah sebuah aktivitas atau cara untuk menghibur *player* dengan tujuan tertentu. Dalam aktivitas ini, ada beberapa aturan yang harus diikuti *player*

selama bermain. Aturan ini membuat *game* berjalan dengan baik dan membantu *player* mencapai tujuan tertentu atau sebuah kemenangan.

Selama bermain *game*, *player* berusaha mencapai berbagai hasil, seperti mencetak poin, mencapai tujuan tertentu, atau mengalahkan *player* lainnya. Ini menciptakan unsur kompetisi di dalam *game*. Kemenangan dalam *game* bisa dicapai dengan berbagai cara, tergantung pada jenis *game* dan aturan yang berlaku (Juvrud et al., 2022).

*Game* biasanya dimainkan dalam suasana yang santai dan tidak serius, di mana *player* dapat menikmati waktu luang mereka dan merilekskan pikiran. Sebagai hiburan, *game* dapat menghilangkan stres, memberikan kesenangan, dan memungkinkan *player* untuk melupakan kekhawatiran sehari-hari sementara mereka terlibat dalam pengalaman bermain yang menyenangkan (Klang et al., 2021).

*Game* adalah bagian penting dari budaya manusia yang telah ada selama berabad-abad. Mereka tidak hanya menyediakan hiburan, tetapi juga membantu membangun keterampilan, meningkatkan kerja sama tim, dan memberikan cara yang menyenangkan untuk menghabiskan waktu bersama teman dan keluarga.

Terdapat banyak *genre game* yang telah dibuat, salah satunya adalah *Role Playing Game*, dimana *player* akan mengontrol penuh karakter yang dimainkannya sesuai rangkaian cerita pada *game* yang dimainkan. Genre RPG cukup populer dikalangan *gamer*, karena ciri khasnya yang menekankan pada pengembangan cerita yang mendalam dan juga memungkinkan para *player* untuk mengontrol

kemampuan dan status dari masing-masing karakter mereka dalam *game* (Latifah et al., 2022).

RPG juga memiliki *sub-genre*, salah satunya adalah *sub-genre Turn-Based* RPG, yang berfokus pada mengalahkan musuh secara bergantian serangan dengan musuh tersebut, secara otomatis *player* harus merencanakan setiap langkah dengan cermat, memikirkan strategi yang tepat untuk mengalahkan musuh (Wayan Alit Rangkan Nuaja et al., 2022). *Turn-Based* RPG menawarkan pengalaman bermain yang kaya, strategis, dan mendalam bagi para *player* yang menyukai tantangan taktis dalam dunia imajinatif (Ashlock et al., 2021).

### ***2.3 Decision Support System***

*Decision Support System* (DSS) adalah sistem komputer atau perangkat lunak yang dimaksudkan untuk membantu dalam situasi tertentu untuk membuat keputusan (Sutyawati & Daniawan, 2024). DSS mengumpulkan dan mengintegrasikan data, menganalisis data dengan alat statistik dan model matematis, dan menawarkan berbagai skenario keputusan. Tujuannya adalah untuk memberikan informasi dan analisis yang relevan bagi pengambil keputusan agar mereka dapat membuat keputusan yang lebih baik dan lebih efisien. DSS dapat digunakan dalam berbagai bidang seperti bisnis, kesehatan, keuangan, dan *game*, membantu pengguna membuat keputusan sehari-hari hingga keputusan yang lebih kompleks dengan lebih baik (Da Silva Oliveira & De Lima Neto, 2023).

Sistem pendukung keputusan pada dasarnya adalah pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang untuk berinteraksi dengan pengguna. Sifat interaktif dimaksudkan untuk memudahkan

integrasi berbagai bagian proses pengambilan keputusan, seperti prosedur, kebijakan, dan teknik analisis, bersama dengan pengalaman dan wawasan manajemen untuk membuat kerangka keputusan yang fleksibel (Wibowo & Thyo Priandika, 2021).

#### **2.4 *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)***

TOPSIS adalah salah satu metode yang digunakan untuk menyelesaikan masalah *Multi Attribute Decision Making (MADM)* (Amalia et al., 2019). MADM merupakan salah satu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari MADM yaitu menentukan nilai bobot setiap atribut yang dilanjutkan dengan meranking dengan menyeleksi alternatif yang sudah diberikan (Rinaldi et al., 2021).

Pada TOPSIS Alternatif yang dipilih didasarkan pada jarak terkecil dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Namun, alternatif yang paling dekat (jarak terdekat) dengan solusi ideal positif tidak selalu mempunyai jarak terbesar dari solusi ideal negatif (Abootalebi et al., 2022).

Menurut (Surahaman & Nursadi, 2019) tahapan – tahapan dari TOPSIS adalah sebagai berikut :

1. Membuat matriks keputusan yang ternormalisasi
2. Membuat matriks keputusan yang ternormalisasi terbobot
3. Membuat matriks solusi ideal positif dan matriks solusi ideal negatif
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.
5. Menentukan nilai preferensi untuk setiap alternatif:

- a. *Decision Matrix* D mengacu terhadap m alternatif yang akan dievaluasi berdasarkan kriteria yang didefinisikan
- b. Dengan  $x_{ij}$  menyatakan performansi dari perhitungan untuk alternatif ke-i terhadap atribut ke-j.

Menurut (Trise Putra et al., 2020) perumusan TOPSIS adalah sebagai berikut:

1. Merangking tiap alternatif

Dibutuhkan ranking kinerja setiap alternatif  $A_i$  pada setiap kriteria  $C_j$  yang ternormalisasi yaitu

$$R_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

2. Matriks keputusan ternormalisasi tebobot

$$Y_{ij} = w_i \cdot r_i$$

3. Solusi ideal positif  $A^+$  dan solusi ideal negatif  $A^-$  dapat ditentukan berdasarkan ranking bobot ternormalisasi ( $y_{ij}$ ) sebagai berikut

$$A^+ = \max(y_1^+, y_2^+, \dots, y_n^+)$$

$$A^- = \min(y_1^-, y_2^-, \dots, y_n^-)$$

4. Jarak dengan solusi ideal

Jarak adalah alternatif dengan solusi ideal positif dirumuskan sebagai :

$$D_1^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

Jarak adalah alternatif  $A_i$  dengan solusi ideal negatif dirumuskan sebagai:

$$D_1^- = \sqrt{\sum_{j=1}^n (y_i^- - y_{ij})^2}$$

#### 5. Nilai Preferensi Untuk Setiap Alternatif

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

Nilai  $V_i$  yang lebih besar menunjukkan bahwa alternatif  $A_i$  lebih dipilih.

### 2.5 *Ordered Weighted Averaging (OWA)*

Sejak Ronald R. Yager pertama kali memperkenalkan *Ordered Weighted Averaging (OWA)* pada tahun 1988, operator ini telah digunakan untuk menggabungkan berbagai sikap pengambil keputusan untuk menghasilkan keputusan keseluruhan (Shu, 2022). OWA merupakan sebuah operator agregasi dengan sifat yang komunikatif, kontinyu, monoton dan cocok digunakan pada transformasi linear. Singkatnya cara kerja operator OWA adalah dengan mengumpulkan nilai respon yang telah diberikan kemudian, mengurutkan argument-argumen dan mengagregasikannya berdasarkan nilai respon yang telah diberikan (Tom, 2021).

Menurut (Luukka & Stoklasa, 2022) operator OWA di definisikan sebagai berikut :

1. OWA berdimensi  $n$  adalah pemetaan dari  $F: R_n \rightarrow R$  yang memiliki asosiasi vektor pembobotan, sehingga  $w_i \in [0, 1], 1 \leq i \leq n, \sum_{i=1}^n w_i = 1$  dan dapat dirumuskan:

$$F(a_1, \dots, a_n)^n = \sum_{j=1}^n w_j b_j = w_1 b_1 + \dots + w_n b_n$$

Dimana  $b_j$  adalah elemen terbesar  $j$  dari kumpulan objek  $a_1, a_2, \dots, a_n$

2. Subset *fuzzy*  $Q$  dari unit interval disebut dengan pembilang *Regular Growing Monotone* (RIM), jika memenuhi kondisi berikut

- 1)  $Q(0) = 0$ ,
- 2)  $Q(1) = 1$ ,
- 3)  $Q(x) \geq Q(y)$ , if  $x > y$ .

RIM dapat digunakan untuk menyatakan istilah seperti 'semua', 'sebagian besar', 'banyak' dan 'setidaknya  $k$ ', di mana  $k$  adalah bilangan bulat. Quantifier yang sering digunakan adalah  $Q(x) = x^\alpha$ ,  $\alpha \geq 0$  dimana bobotnya dihitung sebagai berikut :

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), i = 1, \dots, n$$

Dengan menggunakan rumus tersebut dan pemilihan  $\alpha$  yang tepat, dapat dimodelkan berbagai aturan linguistik.

## **BAB III**

### **DESAIN DAN IMPLEMENTASI**

#### **3.1 Analisis dan Perancangan *Turn-Based* RPG**

##### **3.1.1 Analisis *Turn-Based* RPG**

Pedjoeang merupakan sebuah *game* tiga dimensi dengan *genre turn-based* RPG yang dapat dimainkan pada platform PC. Cerita dalam *game* ini mengisahkan tentang perjuangan seorang pahlawan dalam suatu dunia yang dikuasai oleh penjajah yang kejam. Di tanah air yang dulunya makmur, kehadiran kekuatan asing telah menyebabkan penderitaan bagi rakyat serta penindasan yang melanda. Pahlawan yang memiliki sifat-sifat heroik dan bertekad untuk membebaskan tanah airnya, memimpin perlawanan melawan penjajah tersebut.

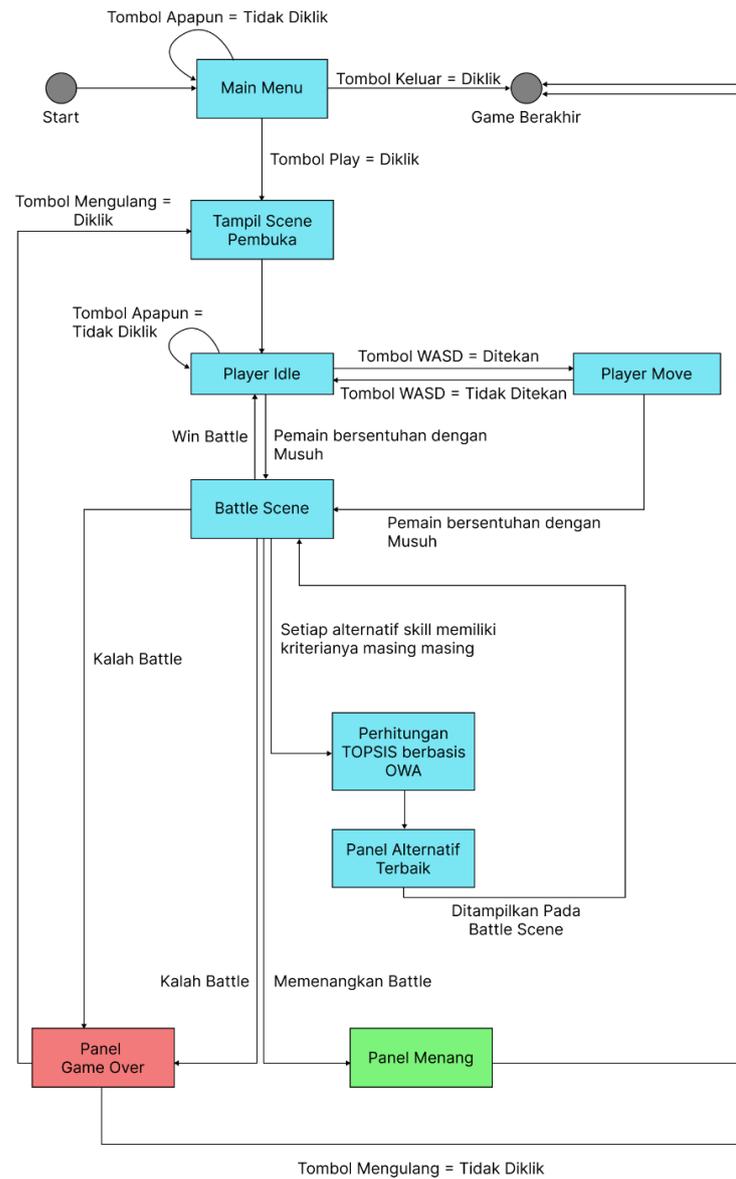
Dalam *game* ini, *player* akan menghadapi musuh-musuh kuat yang merupakan representasi dari kekuatan penjajah. *Player* akan menggunakan beberapa *skill* yang dimiliki pahlawan untuk mengalahkan musuh-musuh tersebut. Namun, *player* harus bijak dalam memilih *skill* yang digunakan, karena tidak semua *skill* dapat memberikan serangan yang efektif terhadap musuh tertentu. Setiap musuh dari pahlawan memiliki kelemahan masing-masing yang harus dipahami dan dieksplorasi oleh *player* untuk mencapai kemenangan.

Selain itu, *game* ini memiliki mekanisme pertarungan bergantian (*turn-based*) antara *player* dan musuh. Artinya, *player* akan memiliki giliran untuk menyerang atau menggunakan *skill*, kemudian musuh akan melakukan hal yang

sama. *Player* harus menggunakan strategi yang tepat dan memanfaatkan kelemahan musuh serta kekuatan pahlawan dengan baik untuk memenangkan pertarungan.

### **3.1.2 Analisis dan Perancangan *Turn-Based* RPG**

*Gameplay* dari *game* Pedjoeang dimulai pada *scene* utama, setelah *player* memilih karakter utamanya. *Player* akan menjelajahi map dan akan bertemu beberapa musuh, ketika musuh bersentuhan dengan pahlawan yang dimainkan, maka *player* akan dibawa kedalam *battlescene*, disinilah *player* akan saling menyerang dengan musuh. Implementasi dari metode TOPSIS berbasis OWA diterapkan dalam *battle scene* yakni memberikan saran atas *skill* yang paling efektif untuk mengalahkan musuh. Untuk mendapatkan saran dari metode TOPSIS berbasis OWA *player* dapat mengklik tombol rekomendasi yang telah disediakan pada antarmuka *battle scene*. Dan berikut ini merupakan *Finite State Machine* dari *game* Pedjoeang, yang merupakan gambaran perjalanan *player* dari awal *Main Menu* sampai akhir *game*.



Gambar 3. 1 Desain FSM Pedjoeang

*Finite State Machine* (FSM) adalah model matematis yang digunakan dalam pengembangan *game* untuk menggambarkan berbagai keadaan atau kondisi yang mungkin dialami oleh *game* dari awal hingga akhir. Dalam *game* Pedjoeang, FSM digunakan untuk menggambarkan perjalanan *player* dari awal *Main Menu* sampai akhir *game*.

Berikut ini merupakan alur dari FSM diatas pertama *player* akan masuk dalam tampilan *mainmenu* dan apabila *player* tidak memilih tombol apapun maka *player* akan tetap berada dalam tampilan *mainmenu*. *Player* akan dibawa keluar *game* apabila *player* memilih tombol keluar, dan akan dibawa pada tampilan *scene* pembuka apabila memilih tombol *Start* atau *Play* dipilih. Saat *player* masuk dalam *scene* utama untuk pertama kali, *player* akan masuk dalam kondisi *idle*, *player* akan bergerak apabila tombol WASD pada *keyboard* ditekan dan akan kembali kedalam posisi *idle* apabila *player* tidak menekan tombol apa-apa pada *keyboardnya*. Pada saat menjelajahi *scene* utama *player* akan dihadapkan dengan 3 musuh yakni Prajurit Hijau, Prajurit Merah dan Jendral de Kock sebagai bosnya. Jika *player* bertabrakan dengan Prajurit Hijau dan Prajurit Merah *player* akan dibawa kedalam *Side Battle Scene* apabila *player* kalah dalam *scene* tersebut maka akan ditampilkan *Panel Game Over* dimana terdapat 2 tombol didalamnya yakni tombol mengulang dan tombol keluar, apabila *player* memilih tombol mengulang maka *player* akan dibawa kembali kedalam *scene* utama, sedangkan apabila *player* memilih tombol keluar maka *player* dibawa keluar dari *game*. Kondisi yang lain pada saat *player* menjelajahi *scene* utama kemudian *player* bertabrakan dengan Jendral de Kock atau bos dalam *game* ini maka *player* akan dibawa masuk kedalam *Boss Battle Scene* dimana apabila *player* berhasil mengalahkan bos tersebut maka akan muncul *Panel Menang*, dan sebaliknya apabila *player* tidak berhasil mengalahkan bos tersebut maka akan muncul *Panel Game Over* seperti pada *Battle Scene* sebelumnya.

### **3.1.3 Rancangan Antarmuka**

Rancangan dari antarmuka *game* Pedjoeang adalah sebagai berikut:

### 1. Tampilan *Mainmenu*

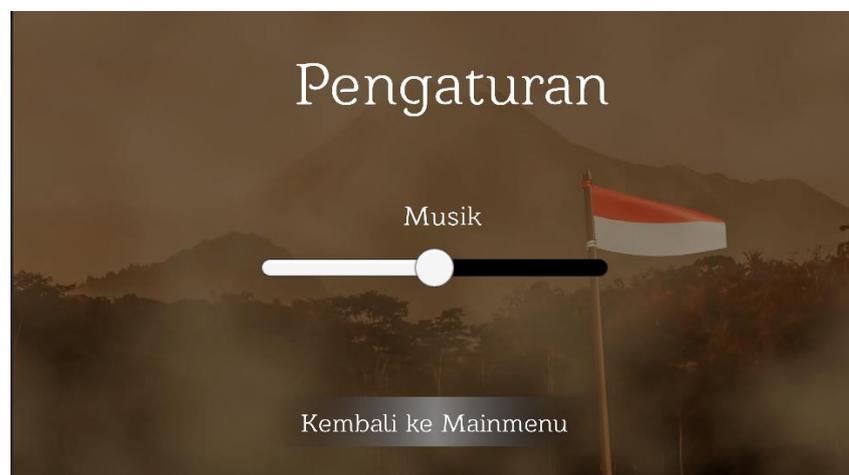
Pada saat pertama kali membuka game, *player* akan disuguhkan dengan tampilan *mainmenu* yang didalamnya terdapat tombol Mulai, Pengaturan, Tentang dan *Exit*.



Gambar 3. 2 Tampilan Mainmenu

### 2. Tampilan Pengaturan

Disini *player* dapat mengatur seberapa besar volume *game*.



Gambar 3. 3 Tampilan Pengaturan

### 3. Tampilan Pilihan Pedjoeang

Setelah *player* mengklik tombol mulai maka *player* akan dibawa ke menu yang digunakan untuk memilih karakter pahlawan yang akan dimainkan.



Gambar 3. 4 Tampilan Pilihan Pedjoeang

### 4. *Gameplay*

Setelah memilih karakter, *player* akan masuk ke scene utama dan akan mendapat kendali penuh atas karakter yang dipilihnya, *player* dapat menggunakan tombol WASD pada keyboard untuk menggerakkan karakter.

Gambar 3. 5 *Gameplay*

### 5. Tampilan *Battle Scene*

Ketika menjelajahi dunia game Pedjoeang *player* akan bertemu musuh, dan saat *player* terlalu dekat atau menyentuh karakter musuh maka *player* akan dibawa ke tampilan battle.

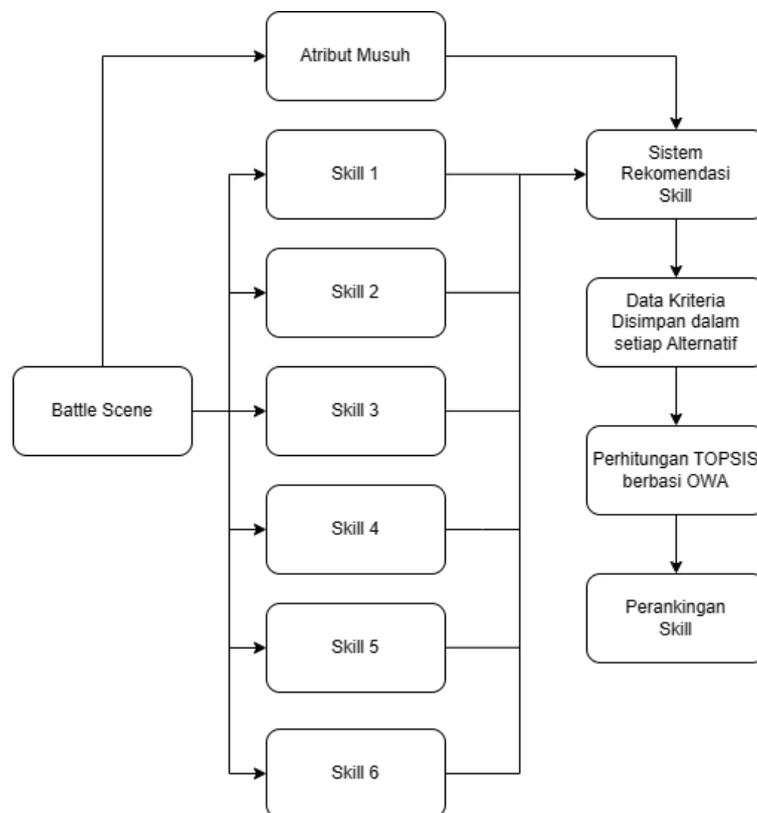
Gambar 3. 6 *BattleScene*

## 3.2 Desain Sistem

Rancangan sistem untuk pengimplementasian metode TOPSIS berbasis OWA dalam *game* Pedjoeang meliputi beberapa elemen kunci, termasuk struktur

dan sistem yang akan digunakan dalam implementasi metode tersebut. Proses desain sistem ini dilakukan setelah tahap analisis sistem selesai dan keputusan untuk menerapkan metode TOPSIS berbasis OWA telah diambil. Salah satu hasil dari desain sistem ini adalah diagram alur kerja sistem dengan bentuk blok diagram yang berisikan proses rekomendasi *skill* dalam game Pedjoeang.

Diagram alur kerja sistem ini akan menggambarkan langkah-langkah yang terjadi dalam proses rekomendasi *skill*. Dan berikut ini merupakan diagram desain sistemnya.



Gambar 3. 7 Desain Sistem

Diagram alur kerja sistem akan memvisualisasikan proses sistem yang menerapkan metode TOPSIS berbasis OWA secara sistematis, dengan

menggambarkan hubungan antara input data, proses analisis TOPSIS berbasis OWA, dan output rekomendasi *skill*. Hal ini akan membantu dalam memahami bagaimana sistem menghasilkan rekomendasi *skill* yang tepat untuk membantu *player* dalam menghadapi musuh dalam *game* Pedjoeang.

### 3.3 Rancangan Perhitungan TOPSIS berbasis OWA

#### 3.3.1 Alternatif

Data alternatif berisikan himpunan alternatif atau kandidat yang akan dievaluasi atau dibandingkan untuk mengambil keputusan. Data alternatif biasanya terdiri dari sejumlah alternatif atau solusi yang memiliki atribut atau kriteria yang relevan untuk masalah atau keputusan yang sedang dipertimbangkan. Setiap alternatif didefinisikan oleh sejumlah nilai atribut yang menggambarkan kinerjanya terhadap kriteria tersebut. Disini data alternatif merupakan kumpulan *skill* yang dimiliki oleh karakter. Berikut alternatif yang digunakan dalam *game* Pedjoeang :

Tabel 3. 1 Alternatif

Alternatif	Arti
A1	<i>Skill 1</i> (Portal Kehidupan)
A2	<i>Skill 2</i> (Topan Kegelapan)
A3	<i>Skill 3</i> (Aura Alam)
A4	<i>Skill 4</i> (Bayangan Maut)
A5	<i>Skill 5</i> (Bola Maut)
A6	<i>Skill 6</i> (Cakra Emas)

Pemilihan *skill* oleh *player* merupakan pemilihan alternatif dari tabel tersebut. *player* dapat melihat variasi kombinasi *skill* dalam Tabel 3.1 sebagai opsi awal, dan nilai-nilai yang terkait dengan masing-masing alternatif kemudian ditentukan berdasarkan penilaian yang terperinci dalam Tabel 3.3.

### 3.3.2 Kriteria

Kriteria merupakan atribut atau faktor-faktor yang digunakan untuk mengevaluasi alternatif atau solusi yang tersedia. Kriteria ini merupakan aspek-aspek yang relevan dengan masalah atau keputusan yang sedang dipertimbangkan. Kriteria ini dapat bervariasi tergantung pada jenis masalah atau keputusan yang akan diselesaikan, Dengan mempertimbangkan kriteria-kriteria ini, *player* dapat membuat keputusan yang lebih cerdas dalam memilih *skill* yang sesuai dengan situasi pertempuran yang dihadapi dalam *game* Pedjoeang. dan berikut kriteria yang digunakan pada *game* Pedjoeang :

Tabel 3. 2 Kriteria

<b>Kriteria</b>	<b>Arti</b>	<b>Jenis Kriteria</b>
K1	<i>Enemy Health Point</i>	Min
K2	<i>Skill Effectiveness</i>	Max
K3	<i>Blue Attack</i>	Max
K4	<i>Red Attack</i>	Max
K5	<i>Green Attack</i>	Max
K6	<i>Skill Energy Cost</i>	Min

Dalam Tabel 3.2, terdapat lima kriteria evaluasi yang digunakan dalam proses pengambilan keputusan. Kriteria-kriteria tersebut adalah *Enemy Health Point*, *Skill Effectiveness*, *Blue Attack*, *Red Attack*, *Green Attack*, dan *Skill Energy Cost*. Kriteria-kriteria ini dapat dikategorikan sebagai kriteria tipe max atau *benefit*, serta terdapat juga kriteria yang bertipe min atau *cost*.

Kriteria tipe max menunjukkan bahwa semakin tinggi nilai kriteria tersebut, semakin baik. Dengan kata lain, peningkatan nilai pada kriteria-kriteria ini akan menghasilkan dampak positif pada evaluasi keseluruhan. Misalnya, semakin tinggi nilai *Skill Effectiveness*, semakin efektif *skill* tersebut dalam pertempuran.

Sementara itu, kriteria tipe min menunjukkan bahwa semakin tinggi nilai kriteria tersebut, semakin buruk. Dengan kata lain, peningkatan nilai pada kriteria-kriteria ini akan menghasilkan dampak negatif pada evaluasi keseluruhan. Contohnya, semakin tinggi nilai *Skill Energy Cost*, semakin banyak energi yang diperlukan untuk menggunakan *skill* tersebut, yang dapat berdampak negatif pada efisiensi dalam pertempuran.

Pemahaman tipe kriteria ini penting dalam pengambilan keputusan karena memungkinkan untuk dapat mengenali apakah suatu peningkatan nilai pada kriteria akan memberikan kontribusi positif atau negatif terhadap evaluasi keseluruhan, serta membantu dalam menetapkan prioritas yang tepat dalam pengembangan atau perbaikan sistem.

### **3.3.2.1 Menentukan Matriks Keputusan**

Dalam matriks keputusan game Pedjoeang, penulis menetapkan skala kepentingan dari 1 hingga 5. Skala ini digunakan untuk menilai tingkat kepentingan setiap kriteria dalam mengevaluasi skill-skill alternatif (alternatif dalam hal ini adalah skill yang dimiliki oleh karakter pahlawan). Nilai 1 mengindikasikan tingkat kepentingan yang sangat rendah, sedangkan nilai 5 menunjukkan tingkat kepentingan yang sangat tinggi.

Matriks keputusan ini disusun dengan setiap baris mewakili alternatif (skill) yang dievaluasi, dan setiap kolom mewakili kriteria evaluasi yang telah ditetapkan sebelumnya, seperti kekuatan serangan, kemampuan pertahanan, kecepatan serangan, efektivitas terhadap jenis musuh, dan durasi efek. Dengan menggunakan matriks keputusan ini, player dapat dengan jelas melihat penilaian atas setiap skill

berdasarkan kriteria yang relevan, dan dengan demikian membuat keputusan yang lebih terinformasi dan strategis saat memilih skill dalam pertempuran.

Tabel 3. 3 Matriks Keputusan

	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	<b>K5</b>	<b>K6</b>
<b>A1</b>	4	5	2	4	1	5
<b>A2</b>	4	1	3	3	2	3
<b>A3</b>	4	4	3	5	2	5
<b>A4</b>	3	2	4	5	4	4
<b>A5</b>	4	5	3	2	3	3
<b>A6</b>	4	3	5	2	3	5

Tabel 3.3 menampilkan matriks keputusan yang mencerminkan penilaian relatif dari setiap alternatif terhadap setiap kriteria. Setiap baris dalam matriks ini mewakili satu alternatif, sementara setiap kolom mencakup satu kriteria. Nilai di setiap sel matriks mengindikasikan tingkat kontribusi atau kecocokan relatif dari setiap alternatif terhadap kriteria yang bersangkutan.

### 3.3.2.2 Normalisasi Matriks Keputusan

Setelah matriks keputusan dibuat, langkah selanjutnya adalah melakukan normalisasi. Normalisasi diperlukan untuk memastikan bahwa setiap elemen dalam matriks memiliki skala yang seragam, sehingga perbandingan antara elemen-elemen tersebut dapat dilakukan dengan benar. Proses normalisasi dilakukan dengan menghitung nilai normalisasi dari setiap elemen matriks keputusan. Ini biasanya melibatkan pembagian setiap elemen dengan jumlah total dari baris atau kolom tempat elemen tersebut berada, tergantung pada jenis normalisasi yang dipilih. Dengan normalisasi, dapat diperoleh representasi yang lebih akurat dari

bobot atau pentingnya setiap alternatif atau kriteria dalam proses pengambilan keputusan.

Tabel 3. 4 Normalisasi Matriks Keputusan

	K1	K2	K3	K4	K5	K6
A1	1	1	0	0.666667	0	1
A2	1	0	0.333333	0.333333	0.333333	0
A3	1	0.75	0.333333	1	0.333333	1
A4	0	0.25	0.666667	1	1	0.5
A5	1	1	0.333333	0	0.666667	0
A6	1	0.5	1	0	0.666667	1

Dalam Tabel 3.4, proses normalisasi dilakukan terhadap matriks keputusan dengan tujuan untuk mengubah nilai-nilai dalam matriks ke skala 0 hingga 1. Normalisasi ini penting untuk mengatasi perbedaan skala antar kriteria yang ada dalam matriks keputusan. Dengan melakukan normalisasi, setiap kriteria menjadi memiliki kontribusi yang seimbang dalam proses pengambilan keputusan, sehingga tidak ada kriteria yang mendominasi keputusan hanya karena skala nilainya yang lebih besar atau lebih kecil dibandingkan kriteria lainnya. Hal ini memastikan bahwa evaluasi terhadap alternatif-alternatif dilakukan secara objektif dan adil, dengan mempertimbangkan setiap kriteria dengan proporsi yang tepat sesuai dengan bobotnya dalam pengambilan keputusan.

### 3.3.2.3 Perhitungan TOPSIS berbasis OWA

Setelah data keputusan dinormalisasi, langkah selanjutnya dalam metode TOPSIS berbasis OWA adalah melakukan perhitungan untuk menentukan kedekatan setiap alternatif dengan solusi ideal positif (PIS) dan solusi ideal negatif (NIS). Proses ini menghasilkan koefisien kedekatan (*closeness coefficient*) untuk

setiap alternatif, yang mengindikasikan seberapa dekat atau baiknya alternatif tersebut dengan solusi ideal positif dan seberapa jauh atau buruknya alternatif tersebut dari solusi ideal negatif. Koefisien kedekatan ini digunakan untuk merangkingkan alternatif-alternatif tersebut, dengan nilai yang lebih tinggi menunjukkan tingkat kedekatan yang lebih baik dengan solusi ideal positif dan nilai yang lebih rendah menunjukkan tingkat kedekatan yang lebih baik dengan solusi ideal negatif.

Tabel 3. 5 Hasil Perhitungan TOPSIS berbasis OWA

	<b>Closeness Coefficient (cc)</b>	<b>PIS</b>	<b>NIS</b>
<b>A1</b>	0.0793031	0.083333	0.967487
<b>A2</b>	0.1488173	0.166667	0.953275
<b>A3</b>	0.175788	0.201389	0.944246
<b>A4</b>	0.4646863	0.581019	0.669327
<b>A5</b>	0.2240884	0.268519	0.929752
<b>A6</b>	0.1380544	0.152778	0.953872

Dalam Tabel 3.5, disajikan hasil perhitungan TOPSIS berbasis OWA, yang mencakup *Closeness Coefficient*, PIS (*Positive Ideal Solution*), dan NIS (*Negative Ideal Solution*). *Closeness Coefficient* mengindikasikan seberapa dekat setiap alternatif dengan solusi ideal positif (PIS) dan seberapa jauh dari solusi ideal negatif (NIS).

*Closeness Coefficient* memberikan gambaran tentang seberapa baik atau buruknya performa relatif setiap alternatif dalam konteks kriteria yang telah ditetapkan. PIS mencerminkan nilai maksimum yang dapat dicapai oleh setiap kriteria, sementara NIS mencerminkan nilai minimum yang diinginkan. Oleh karena itu, *Closeness Coefficient* menyajikan ukuran sejauh mana setiap alternatif mendekati solusi ideal positif dan sejauh mana terpisah dari solusi ideal negatif.

### 3.3.2.4 Perankingan

Hasil dari perhitungan TOPSIS berbasis OWA, yang menghasilkan koefisien kedekatan untuk setiap alternatif, adalah informasi yang berharga dalam pengambilan keputusan. Koefisien kedekatan ini memungkinkan untuk mengurutkan alternatif-alternatif berdasarkan seberapa dekat atau seberapa jauh mereka dari solusi ideal, yang dapat memberikan pandangan yang lebih jelas tentang kualitas relatif dari setiap alternatif. Dengan demikian, hasil dari perhitungan TOPSIS berbasis OWA dapat digunakan untuk menyusun prioritas, mengidentifikasi alternatif yang paling sesuai dengan tujuan atau kriteria tertentu, dan mendukung proses pengambilan keputusan yang lebih informasional dan terarah.

Tabel 3. 6 Perankingan

Alternatif	Ranking
A1	4
A2	5
A3	3
A4	2
A5	6
A6	1

Dalam Tabel 3.6, hasil perhitungan TOPSIS berbasis OWA menampilkan analisis yang mengungkapkan alternatif terbaik berdasarkan kriteria yang telah ditetapkan sebelumnya. Analisis ini melibatkan penggunaan *Closeness Coefficient*, solusi ideal positif (PIS), dan solusi ideal negatif (NIS) yang telah dihitung sebelumnya. Dari hasil tersebut, dapat disimpulkan bahwa alternatif terbaik, yang mencerminkan *skill* terbaik dalam konteks yang ditetapkan, adalah alternatif 6 atau

*skill 6*. Penentuan ini dilakukan berdasarkan perbandingan antara nilai *Closeness Coefficient* dari setiap alternatif, di mana nilai tertinggi menunjukkan tingkat kedekatan yang lebih baik dengan solusi ideal positif dan nilai terendah menunjukkan tingkat kedekatan yang lebih baik dengan solusi ideal negatif. Dengan demikian, hasil perhitungan TOPSIS memberikan informasi yang relevan dan berguna dalam menentukan alternatif terbaik berdasarkan evaluasi yang komprehensif..

### 3.4 Implementasi Perhitungan TOPSIS-OWA

Pada tahap ini, akan diimplementasikan perhitungan metode TOPSIS berbasis OWA dalam sebuah program. Berikut adalah *pseudocode* yang menjelaskan langkah-langkah yang diperlukan untuk melakukan perhitungan tersebut:

#### *Pseudocode 3.1 Implementasi TOPSIS-OWA*

```
function topsissimowa(data, crit, p, alpha1, alpha2, w=None)
    x = data
    m, n = shape of x

    if w is None:
        w = array of ones with length n

    a = (x - minimum value of x in each column) / (maximum value
of x in each column - minimum value of x in each column)
    r = element-wise multiplication of w and a

    PIS = array of zeros with length n
    NIS = array of zeros with length n

    for j in range(n):
        if crit[j] == 1:
            PIS[j] = maximum value of r in column j
            NIS[j] = minimum value of r in column j
        else:
            PIS[j] = minimum value of r in column j
            NIS[j] = maximum value of r in column j

    SPIS = simLPowa(PIS, r, p, alpha1)
    SNIS = simLPowa(NIS, r, p, alpha2)
```

```

    cc = element-wise division of SPIS by (SPIS + SNIS)

    return cc, SPIS, SNIS

function simLPowa(center, data, p, alpha)
    Ideal = outer product of array of ones with length equal to
    number of rows in data and center

    simM = element-wise calculation of (1 - absolute value of
    (data^p - Ideal^p)^(1/p))

    w = Rim1(number of columns in data, alpha)

    totsim = owamatrix(simM, w)

    return totsim

function Rim1(n, m)
    re = empty array
    for h in range(1, n + 1):
        append ((h/n)^m - ((h-1)/n)^m) to re
    return re

function owamatrix(A, w)
    n = number of rows in A
    h = array of zeros with length n
    for i in range(n):
        apu = A[i, :]
        h[i] = sum of (sorted apu in descending order) * w

    return h

crit = [1, 1, 1, 1, 1]

p = 1
alpha1 = 2
alpha2 = 0.1

cc, SPIS, SNIS = topsissimowa(data, crit, p, alpha1, alpha2)

```

*Pseudocode* diatas menggambarkan langkah-langkah yang diperlukan untuk menghitung metode TOPSIS berbasis OWA. Pertama-tama, fungsi topsissimowa menerima data keputusan, kriteria keuntungan atau biaya, serta parameter lainnya seperti  $p$ ,  $\alpha_1$ , dan  $\alpha_2$ . Pada langkah pertama di dalam fungsi ini, dilakukan normalisasi matriks keputusan untuk mengubah nilai-nilai dalam matriks ke skala antara 0 hingga 1, dengan tujuan mengatasi perbedaan skala antar kriteria.

Selanjutnya, nilai-nilai dari matriks yang telah dinormalisasi ini digunakan untuk menghitung nilai PIS (*Positive Ideal Solution*) dan NIS (*Negative Ideal Solution*) berdasarkan kategori kriteria keuntungan atau biaya.

Setelah nilai PIS dan NIS dihitung, langkah selanjutnya adalah menghitung *Similarity to Positive Ideal Solution* (SPIS) dan *Similarity to Negative Ideal Solution* (SNIS) dengan menggunakan fungsi *simLPowa*. Fungsi ini menghasilkan bobot untuk OWA menggunakan RIM quantifier dan mengagregasi matriks *similarity* yang telah dihitung sebelumnya. Bobot yang dihasilkan akan digunakan untuk menghitung *Closeness Coefficients* (cc) untuk setiap alternatif, yang merupakan ukuran seberapa dekat atau baiknya sebuah alternatif terhadap solusi ideal positif dan seberapa jauh atau buruknya alternatif tersebut dari solusi ideal negatif.

### **3.5 Desain Pengujian Sistem**

Penelitian ini bertujuan untuk mengevaluasi efektivitas implementasi metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) yang berbasis OWA (*Ordered Weighted Averaging*) dalam *game* "Pedjoeang." Metode TOPSIS berbasis OWA merupakan suatu pendekatan yang menggabungkan prinsip OWA, dengan metode TOPSIS yang digunakan untuk pemilihan alternatif terbaik.

Pengujian dilakukan dengan membandingkan hasil akhir yang dihasilkan oleh kedua metode tersebut, yaitu metode TOPSIS berbasis OWA dan metode TOPSIS. Tujuan dari perbandingan ini adalah untuk melihat perbedaan kinerja dan keunggulan yang mungkin dimiliki oleh metode berbasis OWA dalam *game*

"Pedjoeang". Metode TOPSIS berbasis OWA sedang diteliti untuk mengevaluasi apakah penggunaan pendekatan OWA dapat meningkatkan efektivitas dan efisiensi dalam pemilihan alternatif terbaik dalam *game* ini. Evaluasi hasil akhir dari kedua metode akan dilakukan secara kuantitatif dengan menganalisis data numerik yang dihasilkan. Hasil evaluasi akan disajikan dalam bentuk grafik untuk memudahkan analisis dan memperoleh pemahaman yang lebih mendalam mengenai perbedaan kinerja dan keefisienan antara kedua metode tersebut. Grafik-grafik tersebut akan memberikan gambaran yang jelas mengenai efektivitas dan keefisienan metode TOPSIS berbasis OWA dalam konteks yang telah ditetapkan, sehingga dapat diambil kesimpulan apakah metode tersebut sudah memenuhi standar yang diharapkan atau masih memerlukan peningkatan lebih lanjut.

Meskipun kedua metode mempunyai fungsi yang sama untuk mencari alternatif terbaik, namun hasil dari metode TOPSIS nantinya akan berbeda dengan TOPSIS-OWA dikarenakan adanya metode OWA sebagai operator agregat dalam perhitungan TOPSIS yang berguna sebagai generalisasi. OWA dapat menggeneralisasi metode TOPSIS, pengukuran *similarity* digantikan posisinya dengan OWA. Oleh sebab itu, muncul pendekatan yang lebih fleksibel dibanding metode TOPSIS yang tidak berbasis OWA dalam perhitungan PIS (*Positive Ideal Solution*) dan NIS (*Negative Ideal Solution*) dengan alternatif-alternatif yang telah ditentukan. Sehingga, dengan menggunakan metode OWA, nilai yang didapat dari proses evaluasi alternatif akan lebih konsisten dan akurat dibanding yang tidak menggunakan metode OWA. (Luukka & Stoklasa, 2022)

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Sistem

Uji coba dilakukan dalam *battlescene* pada *game* Pedjoeang untuk pemilihan *skill* terbaik menggunakan metode TOPSIS berbasis OWA. Pada tahap ini, perhitungan yang telah ditampilkan pada bab sebelumnya diimplementasikan dengan membuatnya sesuai kebutuhan yang telah ditetapkan, kemudian membuat program sesuai dengan desain yang telah dirancang sebelumnya.

##### 4.1.1 Implementasi Perhitungan Metode

Perhitungan antara metode TOPSIS dan yang berbasis OWA, dihitung melalui kode program C# dalam aplikasi Unity. Hasil dari perhitungannya di simpan dalam variabel dengan tipe *string* agar nanti dapat ditampilkan pada *game object* teks.

##### 4.1.1.1 Implementasi Perhitungan TOPSIS

###### 1. Normalisasi Matriks Keputusan

*Pseudocode* 4.1 Normalisasi Matriks Keputusan

```
function normalize_matrix(matrix):
    normalized_matrix = []

    for each column in transpose(matrix):
        normalized_column = []

        for each value in column:
            normalized_value = value / sum(column)
            add normalized_value to normalized_column
        end for

        add normalized_column to normalized_matrix
```

```

end for

return transpose(normalized_matrix)
end function

```

Tahap pertama dalam perhitungan TOPSIS adalah menormalisasi matriks keputusan yang telah di *input* sebelumnya, langkahnya menghitung nilai maksimum untuk setiap kolom, kemudian dilakukan pembagian setiap elemen dalam matriks oleh nilai maksimum tersebut. Hasil dari normalisasi ini adalah matriks keputusan yang memiliki semua elemen dalam rentang 0 sampai 1.

## 2. Normalisasi Terbobot

### *Pseudocode 4.2* Normalisasi Terbobot

```

function multiply_matrix(matrix, weights):
    weighted_matrix = []

    for each row in matrix:
        weighted_row = []

        for each value, weight in zip(row, weights):
            weighted_value = value * weight
            add weighted_value to weighted_row
        end for

        add weighted_row to weighted_matrix
    end for

    return weighted_matrix
end function

```

Setelah dilakukan normalisasi pada matriks keputusan, tahap selanjutnya adalah melakukan normalisasi terbobot untuk matriks keputusan dengan metode TOPSIS. Matriks keputusan dikalikan dengan bobot kriteria. Bobot kriteria berguna untuk menentukan tingkat pentingnya masing-masing kriteria dalam pengambilan keputusan.

## 3. Solusi Ideal Positif dan Negatif

### *Pseudocode 4.3* Solusi Ideal Positif dan Negatif

```

function find_ideal_solutions(matrix, criterion_type):
    positive_ideal = []
    negative_ideal = []

    for each index in range(length(matrix[0])):
        if criterion_type[index] is 'max':
            positive_value = max(matrix, key=lambda row:
row[index])[index]
            negative_value = min(matrix, key=lambda row:
row[index])[index]
        else if criterion_type[index] is 'min':
            positive_value = min(matrix, key=lambda row:
row[index])[index]
            negative_value = max(matrix, key=lambda row:
row[index])[index]
        end if

        add positive_value to positive_ideal
        add negative_value to negative_ideal
    end for

    return positive_ideal, negative_ideal
end function

```

Kode tersebut digunakan untuk mencari solusi ideal positif (PIS) dan solusi ideal negatif (NIS). Dalam fungsi “DetermineIdealSolutions” nilai solusi ideal positif dan negatif ditentukan berdasarkan tipe kriteria. Jika kriteria bersifat "max," maka solusi ideal positif adalah nilai maksimum dalam kolom tersebut, dan solusi ideal negatif adalah nilai minimum. Sebaliknya, jika kriteria bersifat "min," maka solusi ideal positif adalah nilai minimum, dan solusi ideal negatif adalah nilai maksimum.

#### 4. Jarak alternatif terhadap Solusi Ideal Positif dan Negatif

##### *Pseudocode 4.4* Jarak alternatif terhadap Solusi Ideal Positif dan Negatif

```

function calculate_distances(matrix, ideal_solution):
    distances = []

    for each row in matrix:
        distance_squared = 0

        for each value, ideal in zip(row, ideal_solution):
            distance_squared += (value - ideal) ** 2
        end for
    end for

```

```

        distance = square_root(distance_squared)
        add distance to distances
    end for

    return distances
end function

```

Dalam fungsi “calculate\_distances” dihitung jarak antara setiap baris dalam matriks dengan solusi ideal positif (PIS) dan solusi ideal negatif (NIS) yang telah didapat sebelumnya. Pertama variabel “distances” diinisialisasi sebagai list yang akan menyimpan hasil perhitungan jarak antara setiap baris dalam matriks dengan solusi ideal. Selanjutnya terdapat dua tingkat perulangan, perulangan pertama iterasi melalui setiap baris dalam matriks. Perulangan kedua melakukan iterasi melalui setiap elemen dalam baris dan menghitung kuadrat jarak antara nilai aktual dan nilai ideal. Setelah selesai perulangan kedua, hasil kuadrat jarak diakarkan untuk mendapatkan nilai jarak sebenarnya. Nilai jarak ini kemudian ditambahkan ke dalam list “distances”.

### 5. Closeness Coefficient

#### *Pseudocode 4.5 Closeness Coefficient*

```

function calculate_closeness_coefficient(relative_closeness):
    sum_relative_closeness = sum(relative_closeness)

    closeness_coefficient = []

    for each closeness in relative_closeness:
        cc = closeness / sum_relative_closeness
        add cc to closeness_coefficient
    end for

    return closeness_coefficient
end function

```

Kode tersebut digunakan untuk menghitung *Closeness Coefficient* (CC), pertama variabel “sum\_relative\_closeness” dihitung sebagai jumlah dari semua

nilai *relative closeness*. Variabel *closeness\_coefficient* diinisialisasi sebagai list yang akan menyimpan nilai CC untuk setiap baris dalam matriks. Kemudian terdapat perulangan yang mengiterasi melalui setiap nilai *relative closeness* dan menghitung CC dengan membagi nilai *relative closeness* dengan jumlah total *relative closeness*.

## 6. Perankingan

### *Pseudocode 4.6* Perankingan

```
function rank_results(closeness_coefficient):
    ranking = sort_indexes_descending(closeness_coefficient)
    return ranking
end function
```

Perankingan dilakukan dengan mengurutkan nilai dari CC setiap alternatif, mulai dari yang tertinggi sampai yang terendah.

## 4.1.1.2 Implementasi Perhitungan TOPSIS-OWA

### 1. Normalisasi Matriks Keputusan

#### *Pseudocode 4.7* Normalisasi Matriks Keputusan

```
function NormalizeMatrix(matrix):
    m = getRowCount(matrix)
    n = getColumnCount(matrix)
    normalizedMatrix = createMatrix(m, n)

    for each column j in matrix:
        min = findMinValueInColumn(matrix, j)
        max = findMaxValueInColumn(matrix, j)

        for each row i in matrix:
            normalizedMatrix[i, j] = (matrix[i, j] - min) / (max
- min)
        end for
    end for

    return normalizedMatrix
end function
```

Dilakukan perhitungan normalisasi terhadap setiap elemen dalam matriks keputusan, tujuannya adalah untuk membawa nilai – nilai yang beragam ke skala yang seragam atau relatif terhadap rentang tertentu. Sesuai rumus tersebut tahap pertama adalah mendapatkan nilai variabel m dan n, dimana m adalah jumlah baris dan n adalah jumlah kolom dari matriks keputusan, kemudian membuat matriks baru dengan ukuran yang sama, lalu dihitung nilai minimum dan nilai maksimum dari setiap kolomnya, setelah itu dilakukan normalisasi. Dengan cara ini, setiap nilai dalam matriks input diubah sehingga berada dalam rentang 0 hingga 1.

2. Mengagregasikan matriks keputusan ke dalam vektor menggunakan OWA dengan bobot tertentu

*Pseudocode 4. 8 Agregasi Matriks Keputusan*

```
function owamatrix(A, w):
  n = getRowCount(A)
  h = createArray(n)

  for each row i in A:
    apu = getRow(A, i)
    sortedArray = sortDescending(apu)
    h[i] = sum(multiply(sortedArray, w))
  end for

  return h
end function
```

Diketahui bahwa dalam fungsi “owamatrix” dimasukkan matriks keputusan yang didefinisikan dengan variabel “A” dan bobot yang telah ditentukan yang didefinisikan dengan variabel “w”. Selanjutnya, dilakukan inisialisasi variabel n untuk menyimpan jumlah baris dalam matriks A, dan array h dibuat untuk menampung hasil perhitungan OWA.

3. Menghasilkan bobot dengan RIM *quantifier*

*Pseudocode 4. 9 RIM quantifier*

```
function Rim1(n, m):
```

```

re = createArray(n)

for h = 1 to n:
    re[h - 1] = pow(h/n, m) - pow((h - 1)/n, m)

return re
end function

```

Kode tersebut digunakan untuk menghasilkan bobot dengan *Regularly Increasing Monotone (RIM) quantifier*. Nantinya bobot ini akan digunakan pada perhitungan metode OWA tepatnya untuk mencari *similarity*. Fungsi “Rim1” dimulai dengan mendefinisikan variabel “re” sebagai array yang akan menyimpan hasil perhitungan bobot. Selanjutnya, dilakukan iterasi dari “h = 1” hingga “h = n”, di mana setiap iterasi menghasilkan nilai bobot sesuai dengan rumus RIM. Bobot ini dihitung dengan mengambil pangkat masing-masing dari rasio “h/n” dan “h-1/n”. Hasil perhitungan tersebut kemudian disimpan dalam *array* “re”.

#### 4. *Similarity*

##### *Pseudocode 4.10 Similarity*

```

function simLPowa(center, data, p, alpha):
    m = getRowCount(data)
    simM = createMatrix(m, length(center))

    for each row i in data:
        for each column j in center:
            simM[i, j] = 1 - pow(abs(pow(data[i, j], p) -
pow(center[j], p)), 1 / p)
        end for
    end for

    w = Rim1(length(center), alpha)
    totsim = owamatrix(simM, w)

    return totsim
end function

```

Fungsi *simLPowa* nantinya digunakan untuk mencari *similarity* dari *Posistive Ideal Solution (PIS)* dan *Negative Ideal Solution (NIS)*. Tahapannya

adalah setiap elemen “simM” dihitung berdasarkan perbandingan antara elemen data dan *center* menggunakan rumus *similarity*. Selanjutnya, bobot untuk metode OWA dihasilkan dengan memanggil fungsi “Rim1” dengan parameter panjang *center* dan nilai alpha. Akhirnya, matriks *similarity* tersebut diagregasi menggunakan metode OWA dengan memanggil fungsi *owamatrix*. Hasil akhirnya, yaitu *Similarity to Positive Ideal Solution (SPIS)* dan *Similarity to Negative Ideal Solution (SNIS)*.

## 5. Perhitungan TOPSIS-OWA

### Pseudocode 4. 11 TOPSIS-OWA

```
function topsissimowa(data, crit, cc, SPIS, SNIS):
    m = getRowCount(data)
    n = getColumnCount(data)
    a = NormalizeMatrix(data)
    r = createMatrix(m, n)

    for each row i in data:
        for each column j in data:
            r[i, j] = w[j] * a[i, j]
        end for
    end for

    PIS = createArray(n)
    NIS = createArray(n)

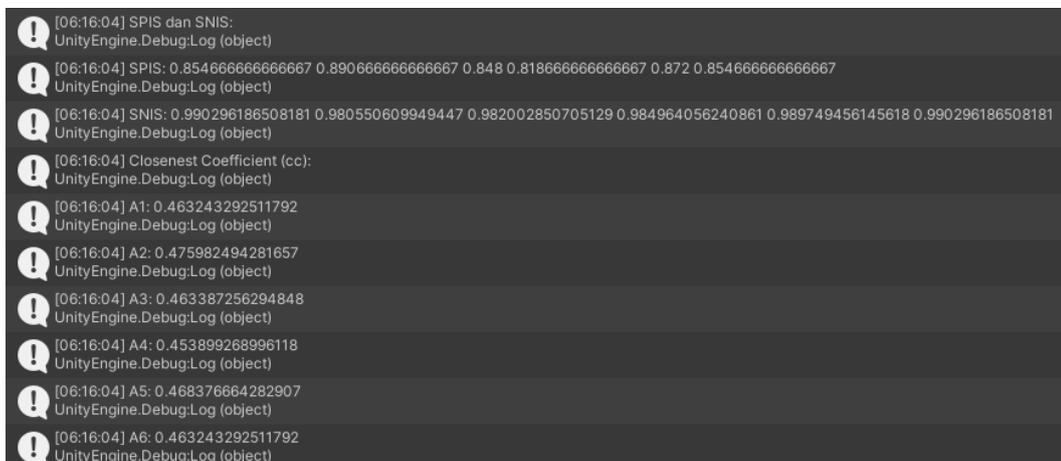
    for each column j in data:
        if crit[j] == 1:
            PIS[j] = maxColumn(r, j)
            NIS[j] = minColumn(r, j)
        else:
            PIS[j] = minColumn(r, j)
            NIS[j] = maxColumn(r, j)
        end if
    end for

    SPIS = simLPowa(PIS, r, 1, 2)
    SNIS = simLPowa(NIS, r, 1, 0.1)

    cc = createArray(m)
    for each row i in data:
        cc[i] = SPIS[i] / (SPIS[i] + SNIS[i])
    end for
end function
```

Kode tersebut digunakan untuk mencari *Similarity to Positive Ideal Solution* (SPIS), *Similarity to Negative Ideal Solution* (SNIS), dan *Closeness Coefficient* dari setiap alternatif dengan menggabungkan seluruh fungsi yang telah dijelaskan sebelumnya. Fungsi tersebut akan mengambil variabel “data” yang merupakan matriks keputusan dan variabel “crit” yang merupakan kriteria, keduanya telah di *input* secara manual sebelumnya, kemudian data tersebut akan dihitung sampai diketahui CC dari setiap alternatifnya.

Implementasi tersebut akan menghasilkan nilai yang ditampilkan pada menu *console* Unity, sebagai berikut



```

[06:16:04] SPIS dan SNIS:
UnityEngine.Debug.Log (object)
[06:16:04] SPIS: 0.854666666666667 0.890666666666667 0.848 0.818666666666667 0.872 0.854666666666667
UnityEngine.Debug.Log (object)
[06:16:04] SNIS: 0.990296186508181 0.980550609949447 0.982002850705129 0.984964056240861 0.989749456145618 0.990296186508181
UnityEngine.Debug.Log (object)
[06:16:04] Closeness Coefficient (cc):
UnityEngine.Debug.Log (object)
[06:16:04] A1: 0.463243292511792
UnityEngine.Debug.Log (object)
[06:16:04] A2: 0.475982494281657
UnityEngine.Debug.Log (object)
[06:16:04] A3: 0.463387256294848
UnityEngine.Debug.Log (object)
[06:16:04] A4: 0.453899268996118
UnityEngine.Debug.Log (object)
[06:16:04] A5: 0.468376664282907
UnityEngine.Debug.Log (object)
[06:16:04] A6: 0.463243292511792
UnityEngine.Debug.Log (object)

```

Gambar 4. 1 Hasil Perhitungan Metode dalam Unity

Setelah itu nilai akan diurutkan dari yang terbesar sampai yang terkecil untuk diambil peringkat alternatif terbaik.

## 4.2 Pengujian Sistem

Setelah metode diimplementasikan selanjutnya sistem akan masuk ke tahap pengujian, pengujian dilakukan dengan cara memainkan *game* yang telah dibuat

dan memastikan seluruh fitur dari sistem yang dibangun berjalan sesuai dengan harapan.

#### 4.2.1 Uji Coba *Game*

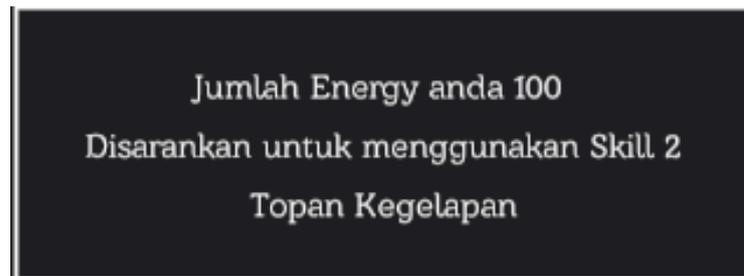
*Player* dipersilahkan untuk memainkan *game* sampai masuk kedalam *battle scene* dikarenakan fitur rekomendasi pemilihan *skill* berada didalam *scene* tersebut. *Player* dipersilahkan memilih berbagai *skill* yang berbeda untuk mengalahkan musuh, sebab setiap *skill* memiliki kriteria yang berbeda. Maka dari itu setiap *battle scene* memiliki bobot yang berbeda terhadap setiap kriteria *skill*. *Game* ini memiliki 3 *battle scene*, yang artinya *player* akan melawan 3 musuh yaitu prajurit hijau, prajurit merah, dan *boss* yang harus dikalahkan Jendral de Kock.

Untuk memasuki *battle scene*, *player* diharuskan mengelilingi *scene* petualangan sampai bertemu antara 3 musuh, *player* harus bertabrakan dengan salah satu musuh yang ditemuinya agar *player* dapat memasuki *battle scene*. Di dalam *battle scene*, *player* diberikan tombol untuk memunculkan sistem rekomendasi TOPSIS – OWA, tampilan tombolnya adalah sebagai berikut.



Gambar 4. 2 Tombol Hint

Apabila *player* menekan tombol tersebut maka akan keluar tampilan sistem rekomendasi seperti berikut ini.



Gambar 4. 3 Sistem Rekomendasi

Sistem tersebut adalah sistem rekomendasi berbasis TOPSIS-OWA, bila dibandingkan dengan TOPSIS, maka hasilnya sebagai berikut.

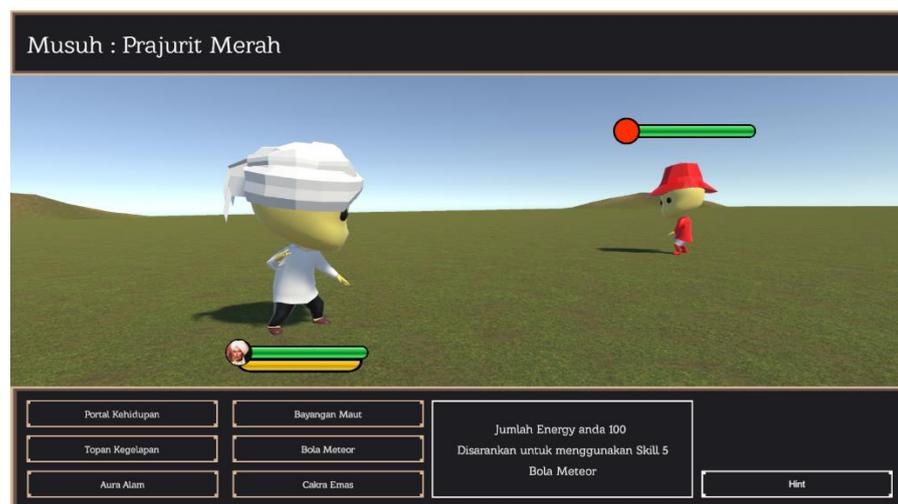
Tabel 4. 1 Hasil Sistem Rekomendasi Setiap *Scene*

Musuh	TOPSIS-OWA	TOPSIS
Prajurit Hijau	<i>Skill 1</i>	<i>Skill 1</i>
Prajurit Merah	<i>Skill 5</i>	<i>Skill 5</i>
Jendral de Kock	<i>Skill 2</i>	<i>Skill 2</i>

1. Uji coba pertama dilakukan pada *scene* melawan Prajurit Hijau, digunakan metode TOPSIS-OWA untuk perhitungan sistem rekomendasi, hasilnya sistem memberikan saran untuk menggunakan *Skill 1* (Portal Kehidupan) dan jika sisa *energy* dibawah 60 maka disarankan menggunakan *Skill 3* (Aura Alam). Seperti pada gambar berikut ini.

Gambar 4. 4 *Battlescene* Prajurit Hijau

2. Uji coba kedua dilakukan pada *scene* melawan Prajurit Merah, digunakan metode TOPSIS-OWA untuk perhitungan sistem rekomendasi, hasilnya sistem memberikan saran untuk menggunakan *Skill 5* (Bola Meteor) dan jika sisa energy dibawah 60 maka disarankan menggunakan *Skill 6* (Cakra Emas). Seperti pada gambar berikut ini.



Gambar 4. 5 *Battlescene* Prajurit Merah

3. Uji coba ketiga dilakukan pada *scene* melawan Jendral de Kock, digunakan metode TOPSIS-OWA untuk perhitungan sistem rekomendasi, hasilnya sistem memberikan saran untuk menggunakan *Skill 2* (Topan Kegelapan) dan jika sisa energy dibawah 60 maka disarankan menggunakan *Skill 4* (Bayangan Maut). Seperti pada gambar berikut ini.

Gambar 4. 6 *Battlescene* Jendral de Kock

#### 4.2.2 Hasil Uji Coba

Implementasi sistem TOPSIS-OWA pada *game* ini dibuat langsung dengan *script* C#, dimana seluruh input sekaligus perhitungannya dilakukan dalam *script* tersebut. Hasil dari perhitungan disimpan dalam variabel pada *script* C# dan ditampilkan pada teks Unity.

Dikarenakan terdapat kriteria yang memiliki nilai berbeda di setiap *scene* yakni K2 yang merupakan *Skill Effectiveness* dan K6 yang merupakan *Energy Cost* maka dibuat matriks keputusan yang berbeda untuk setiap *scene*. Tidak hanya matriks keputusan yang dibuat berbeda, pemberian bobot juga dibuat berbeda untuk setiap *battlescene* menyesuaikan tingkat kepentingan pada setiap *battlescene*.

##### a. Prajurit Hijau

Tabel 4. 2 Matriks Keputusan *Battlescene* Prajurit Hijau

Alternatif	K1	K2	K3	K4	K5	K6
<i>Skill 1</i>	100	25	7	6	7	60
<i>Skill 2</i>	100	11	8	7	6	70
<i>Skill 3</i>	100	18	5	7	7	20
<i>Skill 4</i>	100	12	6	7	6	10
<i>Skill 5</i>	100	10	7	8	6	13
<i>Skill 6</i>	100	12	7	7	6	10

Tabel 4. 3 Bobot *Battlescene* Prajurit Hijau

<b>Kriteria</b>	<b>Bobot</b>
<i>Enemy Health Point</i>	0.08333333
<i>Skill Effectiveness</i>	0.30555556
<i>Blue Attack</i>	0.13888889
<i>Red Attack</i>	0.19444444
<i>Green Attack</i>	0.25
<i>Skill Energy Cost</i>	0.02777778

Sesuai dengan matriks keputusan dan bobot yang diberikan untuk melawan Prajurit Hijau, dibandingkan hasil dari kedua metode yaitu TOPSIS dan TOPSIS-OWA, berikut merupakan hasil dari kedua metode.

Tabel 4. 4 Perbandingan Metode *Battlescene* Prajurit Hijau

<b>Alternatif</b>	<b>TOPSIS</b>	<b>Ranking</b>	<b>TOPSIS-OWA</b>	<b>Ranking</b>
<i>Skill 1</i>	0.83518693	1	0.48462473	1
<i>Skill 2</i>	0.22810159	3	0.45473471	4
<i>Skill 3</i>	0.51107066	2	0.47403201	2
<i>Skill 4</i>	0.16417082	6	0.45112305	6
<i>Skill 5</i>	0.1871298	5	0.45720776	3
<i>Skill 6</i>	0.19363441	4	0.4526935	5
<b>Total</b>	<b>2.11929421</b>		<b>2.77441576</b>	

#### b. Prajurit Merah

Tabel 4. 5 Matriks Keputusan *Battlescene* Prajurit Merah

<b>Alternatif</b>	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	<b>K5</b>	<b>K6</b>
<i>Skill 1</i>	100	13	7	6	7	10
<i>Skill 2</i>	100	11	8	7	6	70
<i>Skill 3</i>	100	10	5	7	7	13
<i>Skill 4</i>	100	12	6	7	6	10
<i>Skill 5</i>	100	25	7	8	6	60
<i>Skill 6</i>	100	18	7	7	6	20

Tabel 4. 6 Bobot *Battlescene* Prajurit Merah

<b>Kriteria</b>	<b>Bobot</b>
<i>Enemy Health Point</i>	0.08333333
<i>Skill Effectiveness</i>	0.30555556
<i>Blue Attack</i>	0.13888889
<i>Red Attack</i>	0.25
<i>Green Attack</i>	0.19444444
<i>Skill Energy Cost</i>	0.02777778

Sesuai dengan matriks keputusan dan bobot yang diberikan untuk melawan Prajurit Merah, dibandingkan hasil dari kedua metode yaitu TOPSIS dan TOPSIS-OWA, berikut merupakan hasil dari kedua metode.

Tabel 4. 7 Perbandingan Metode *Battlescene* Prajurit Merah

<b>Alternatif</b>	<b>TOPSIS</b>	<b>Ranking</b>	<b>TOPSIS-OWA</b>	<b>Ranking</b>
<i>Skill 1</i>	0.23832208	3	0.46341004	3
<i>Skill 2</i>	0.23596851	4	0.45741641	4
<i>Skill 3</i>	0.13436047	6	0.45704115	5
<i>Skill 4</i>	0.17671543	5	0.45380574	6
<i>Skill 5</i>	0.89025028	1	0.48462473	1
<i>Skill 6</i>	0.52429539	2	0.4660849	2
<b>Total</b>	<b>2.19991216</b>		<b>2.78238297</b>	

### c. Jendral de Kock

Tabel 4. 8 Matriks Keputusan *Battlescene* Jendral de Kock

<b>Alternatif</b>	<b>K1</b>	<b>K2</b>	<b>K3</b>	<b>K4</b>	<b>K5</b>	<b>K6</b>
<i>Skill 1</i>	100	13	7	6	7	70
<i>Skill 2</i>	100	25	8	7	6	60
<i>Skill 3</i>	100	12	5	7	7	10
<i>Skill 4</i>	100	18	6	7	6	20
<i>Skill 5</i>	100	10	7	8	6	13
<i>Skill 6</i>	100	11	7	7	6	10

Tabel 4. 9 Bobot *Battlescene* Jendral de Kock

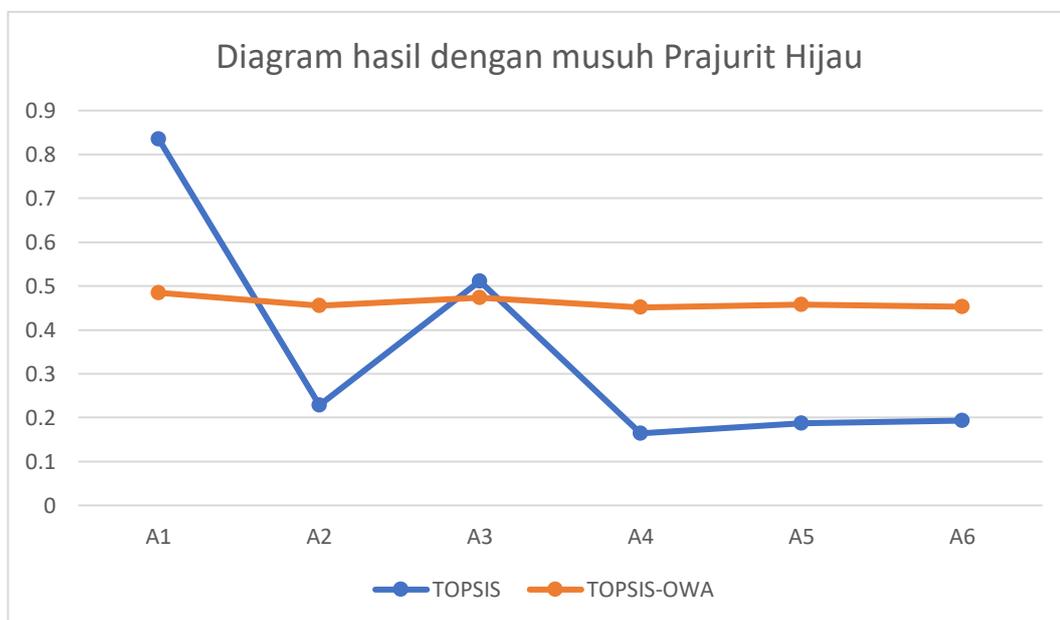
<b>Kriteria</b>	<b>Bobot</b>
<i>Enemy Health Point</i>	0.08333333
<i>Skill Effectiveness</i>	0.30555556
<i>Blue Attack</i>	0.25
<i>Red Attack</i>	0.19444444
<i>Green Attack</i>	0.13888889
<i>Skill Energy Cost</i>	0.02777778

Sesuai dengan matriks keputusan dan bobot yang diberikan untuk melawan Jendral de Kock, dibandingkan hasil dari kedua metode yaitu TOPSIS dan TOPSIS-OWA, berikut merupakan hasil dari kedua metode.

Tabel 4. 10 Perbandingan Metode *Battlescene* Jendral de Kock

Alternatif	TOPSIS	Ranking	TOPSIS-OWA	Ranking
<i>Skill 1</i>	0.31735611	3	0.46402958	4
<i>Skill 2</i>	0.89697741	1	0.49301939	1
<i>Skill 3</i>	0.18903542	6	0.45406636	6
<i>Skill 4</i>	0.50155385	2	0.47006981	2
<i>Skill 5</i>	0.24266724	4	0.46923905	3
<i>Skill 6</i>	0.22393331	5	0.45838376	5
<b>Total</b>	<b>2.37152334</b>		<b>2.80880795</b>	

Berikut adalah bentuk grafik perbedaan dari kedua metode TOPSIS dan TOPSIS-OWA, dimana pada grafik ini akan memperlihatkan perubahan posisi dan bentuk dari datanya.

Diagram 4. 1 Perbandingan Metode pada *Scene* Prajurit Hijau

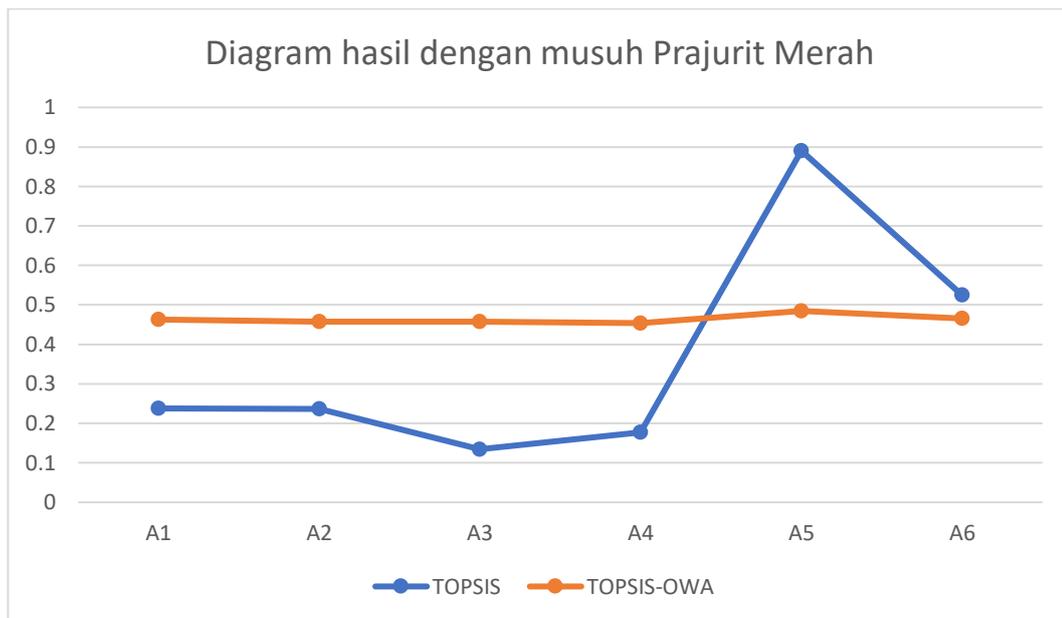


Diagram 4. 2 Perbandingan Metode pada Scene Prajurit Merah

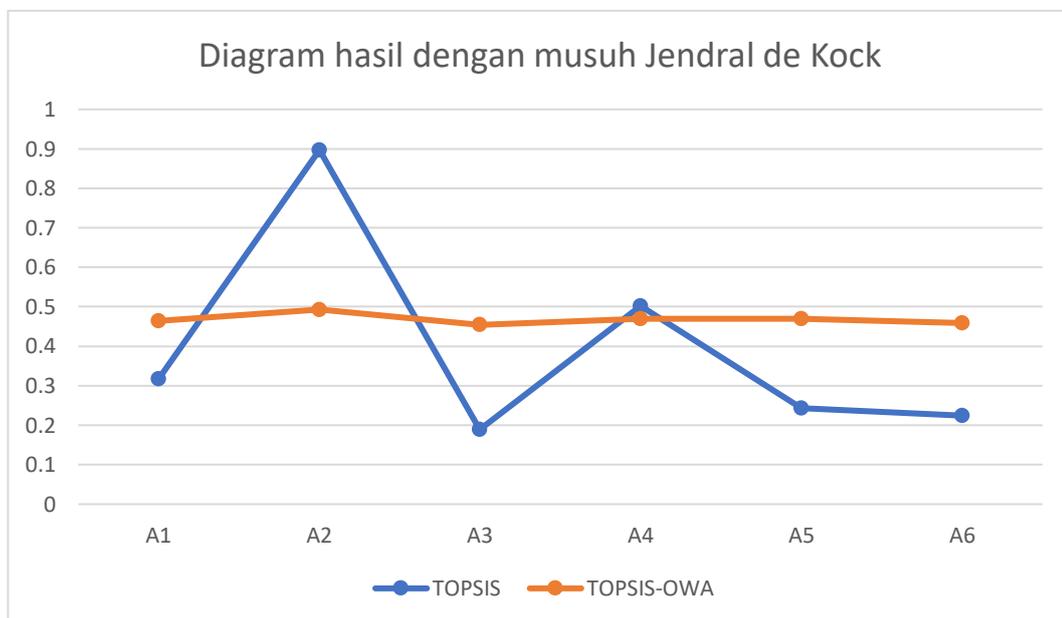


Diagram 4. 3 Perbandingan Metode pada Scene Jendral de Kock

Dari ketiga diagram tersebut dapat disimpulkan bahwa metode Topsis menghasilkan nilai yang berbeda secara signifikan pada setiap alternatif, dan menghasilkan perankingan alternatif yang sedikit berbeda namun untuk ranking 1 dan 2 hasilnya sama dengan metode Topsis-OWA sedangkan metode Topsis-

OWA menghasilkan nilai yang hampir sama pada setiap alternatif. Jika dilihat dari kebutuhan sistem metode TOPSIS menunjukkan nilai yang kurang sesuai pada ranking 3 dari *battlescene* melawan Prajurit Hijau dan Jendral De Kock, sedangkan metode TOPSIS-OWA dapat menunjukkan perankingan yang sesuai dengan kebutuhan sistem, mulai dari ranking 1 sampai ranking 6.

### 4.3 Integrasi dengan Islam

#### 1. *Muamalah Ma'a Allah Subhanahu wa ta'ala*

Di akhir *game* jika *player* berhasil mengalahkan Jendral de Kock maka akan muncul *panel win* menandakan *player* memenangkan *game*, di dalam *panel win* tersebut terdapat kalimat *thayyibah hamdalah* yang bermakna “segala puji bagi Allah”. Hal ini terinspirasi dari Al – Qur’an surah Al-Fatihah ayat 2 yang berbunyi.

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

“Segala puji bagi Allah, Tuhan seluruh alam” (QS. Al-Fatihah:2)

Menurut tafsir kemenag RI atas ayat diatas menunjukkan bahwa segala puji dipersembahkan hanya untuk Allah semata, Tuhan Pencipta dan Pemelihara seluruh alam. Dialah Yang Maha Pengasih, Pemilik dan sumber sifat kasih. Dialah yang menganugerahkan segala macam karunia, baik besar maupun kecil (Kementrian Agama RI, 2024b). Sistem rekomendasi berbasis TOPSIS-OWA yang telah dibangun mempunyai tujuan untuk memudahkan *player* dalam menyelesaikan *game* Pedjoeang, sistem ini semata mata adalah karunia dari Allah, untuk hambanya. Oleh karena itu pada saat berhasil memenangkan *game* terdapat kalimat

untuk mensyukuri karunia dari Allah, yakni kalimat hamdalah. Dan juga supaya sebagai hamba-Nya, *player* selalu mengingat Allah saat mendapat nikmat sebagai ungkapan bersyukur atau berterima kasih, seperti dalam firman-Nya Al-Qur'an surat Ibrahim ayat 7

وَإِذْ تَأَذَّنَ رَبُّكُمْ لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ وَلَئِن كَفَرْتُمْ إِنَّ عَذَابِي لَشَدِيدٌ

“Dan (ingatlah) ketika Tuhanmu memaklumkan, “Sesungguhnya jika kamu bersyukur, niscaya Aku akan menambah (nikmat) kepadamu, tetapi jika kamu mengingkari (nikmat-Ku), maka pasti azab-Ku sangat berat.”(QS. Ibrahim:7)

Ibn Katsir dalam kitab tafsirnya menjelaskan bahwa orang yang bersyukur adalah orang yang tahu berterima kasih. Bukan sekedar banyak atau sedikitnya rejeki yang diperoleh, disamping itu bersyukur juga dapat menambah keimanan seseorang. Karena hal itu dalam sistem ini diterapkan kalimat hamdalah sebagai ungkapan bersyukur kepada Allah (Andyan Fitryansyah & Hadiyyin, 2023).

Selain terdapat kalimat hamdalah saat berhasil memenangkan *game*, terdapat juga kalimat tarji' saat *player* kalah dalam *game* ini. Hal ini terinspirasi dari Al-Quran surat Al-Baqarah ayat 156, yang bunyinya.

الَّذِينَ إِذَا أَصَابَتْهُمُ مُصِيبَةٌ قَالُوا إِنَّا لِلَّهِ وَإِنَّا إِلَيْهِ رَاجِعُونَ

“(yaitu) orang-orang yang apabila ditimpa musibah, mereka mengucapkan “Innā lillāhi wa innā ilaihi rāji‘ūn” (sesungguhnya kami adalah milik Allah dan sesungguhnya hanya kepada-Nya kami akan kembali).”

Sesuai dengan tafsir kemenag RI kehidupan manusia memang penuh cobaan. Dan kami pasti akan menguji kamu untuk mengetahui kualitas keimanan seseorang dengan sedikit ketakutan, kelaparan, kekurangan harta, jiwa, dan buah -

buah. Bersabarlah dalam menghadapi semua itu. Dan sampaikanlah kabar gembira, wahai nabi Muhammad, kepada orang-orang yang sabar dan tangguh dalam menghadapi cobaan hidup, yakni orang-orang yang apabila ditimpa musibah, apa pun bentuknya, besar maupun kecil, mereka berkata, *inna' lilla'hi wa inna' ilaihi ra'ji'un* (sesungguhnya kami milik Allah dan kepada-Nyalah kami kembali) (Kementrian Agama RI, 2024a). Kekalahan dalam *game* ini merupakan suatu musibah, dengan adanya kalimat tarji' saat kalah dalam *game*, *player* diajak untuk mengingat Allah saat menerima musibah dengan mengucapkan kalimat tarji'. Dan juga agar *player* dapat bersabar dengan musibah yang dihadapinya.

Adanya kalimat thayyibah dalam *game* ini, dimaksudkan untuk *player* agar selalu mengingat Allah dalam segala situasi dan kondisi.

## 2. Muamalah Ma'a An-Nas

Telah diketahui apabila *player* tidak menggunakan *skill* yang direkomendasikan oleh sistem, maka *player* akan mengalami kekalahan dan sebaliknya jika *player* mengikuti *skill* yang direkomendasikan oleh sistem maka *player* dapat meraih kemenangan dalam *game* ini. hal tersebut terdapat kaitannya dengan sistem yang menolong *player* untuk mencapai kemenangan dengan cara membantu memilih *skill* yang digunakan, sebagaimana dalam agama Islam setiap manusia diajarkan untuk selalu menerapkan perilaku tolong-menolong dan saling membantu terhadap sesama seperti dalam Al – Qur'an surat Al-Maidah ayat 2 yang berbunyi

يٰۤاَيُّهَا الَّذِيْنَ ءَامَنُوْا لَا تُجْلُوْا شَعْبَرَ اللّٰهِ وَلَا الشَّهْرَ الْحَرَامَ وَلَا الْهٰدِيَ وَلَا الْقَلْبِدَ وَلَا ءَاتِمِيْنَ الْبَيْتِ الْحَرَامِ يَبْتَغُوْنَ فَضْلًا  
مِّن رَّبِّهِمْ وِرْضُوْنَ ۗ وَاِذَا حَلَلْتُمْ فَاصْطَادُوْا ۗ وَلَا يَجْرِمَنَّكُمْ شَنَاٰنُ فَوْمٍ اَنْ صَدُوْكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ اَنْ تَعْتَدُوْا . وَتَعَاوَنُوْا  
عَلَى الْبِرِّ وَالتَّقْوٰى ۗ وَلَا تَعَاوَنُوْا عَلَى الْاِثْمِ وَالْعُدُوْنِ ۗ وَاتَّقُوا اللّٰهَ ۗ اِنَّ اللّٰهَ شَدِيْدُ الْعِقَابِ

*“Hai orang-orang yang beriman, janganlah kamu melanggar syi'ar-syi'ar Allah, dan jangan melanggar kehormatan bulan-bulan haram, jangan (mengganggu) binatang-binatang had-ya, dan binatang-binatang qalaa-id, dan jangan (pula) mengganggu orang-orang yang mengunjungi Baitullah sedang mereka mencari kurnia dan keridhaan dari Tuhannya dan apabila kamu telah menyelesaikan ibadah haji, maka bolehlah berburu. Dan janganlah sekali-kali kebencian(mu) kepada sesuatu kaum karena mereka menghalang-halangi kamu dari Masjidil haram, mendorongmu berbuat aniaya (kepada mereka). Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya (QS. Al-Maidah:2)*

Berdasarkan tafsir Kemenag RI tentang ayat diatas manusia diperintahkan untuk saling tolong-menolong, dan tidak diperbolehkan untuk saling membenci (Lajnah Pentashihan Mushaf Al-Qur'an, 2022b). Sistem yang dibangun bertujuan untuk menolong *player* dalam mengalahkan musuh dan mencapai kemenangan, diharapkan *player* juga dapat menerapkan dalam kehidupan sehari – hari untuk saling tolong-menolong sesama manusia dan tidak saling membenci.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian mengimplementasikan metode *Technique for Order Performance by Similarity to Ideal Solution* (TOPSIS) berbasis *Ordered Weighted Averaging* (OWA) untuk melakukan perhitungan rekomendasi pemilihan *skill* pada *game* “PEDJOEANG”. Perankingan alternatif berdasarkan matriks keputusan dan bobot yang telah ditentukan, adapun matriks keputusan dan bobot pada setiap *battlescene* dibuat berbeda karena menyesuaikan dengan kebutuhan dalam *scene* tersebut.

Setelah sistem dibuat dan metode diimplementasikan, hasil dari metode TOPSIS-OWA dibandingkan dengan TOPSIS dan telah diketahui bahwa hasil perankingan dari metode TOPSIS kurang sesuai dengan sistem yang dibuat yakni pada ranking 3 *scene* melawan Prajurit Hijau dan Jendral De Kock, namun tetap sesuai dengan sistem pada ranking 1 dan 2, sedangkan hasil dari TOPSIS-OWA menunjukkan perankingan yang sesuai dengan sistem mulai dari ranking 1 sampai ranking 6 di seluruh *scene*.

Setelah dilakukan pengujian dengan membandingkan metode TOPSIS-OWA dan metode TOPSIS, dapat diambil kesimpulan bahwa metode TOPSIS-OWA layak untuk digunakan dalam sistem.

#### 5.2 Saran

Meskipun telah dilakukan proses pengujian dan mendapatkan kesimpulan, peneliti menyadari bahwa masih terdapat kekurangan pada penelitian ini dan

peneliti ingin memberikan saran yang ditujukan untuk penelitian selanjutnya, berikut beberapa saran yang diberikan.

1. Penambahan alternatif dan kriteria yang lebih beragam. Dengan tujuan membuat *player* membutuhkan sistem rekomendasi yang dibuat.
2. Penggunaan metode lain dalam sistem rekomendasi, baik metode yang dapat dikombinasikan dengan metode TOPSIS ataupun metode selain TOPSIS. Dengan harapan memperoleh hasil yang lebih baik.
3. Penambahan animasi, mekanik *game*, cerita sehingga membuat *game* lebih menarik untuk dimainkan.

## DAFTAR PUSTAKA

- Abootalebi, S., Hadi-Vencheh, A., & Jamshidi, A. (2022). Ranking the Alternatives With a Modified TOPSIS Method in Multiple Attribute Decision Making Problems. *IEEE Transactions on Engineering Management*, 69(5), 1800–1805. <https://doi.org/10.1109/TEM.2019.2933593>
- Amalia, E. L., RDA, R. A., & Pratama, A. N. (2019). Sistem Pendukung Keputusan Menentukan Lovebird Unggul dalam Perlombaan Menggunakan Metode AHP-Topsis. *MATICS: Jurnal Ilmu Komputer Dan Teknologi Informasi (Journal of Computer Science and Information Technology)*, 11(1), 21. <https://doi.org/10.18860/mat.v11i1.7690>
- Andyan Fitriyansyah, M., & Hadiyyin, I. (2023). Syukur Sebagai Fondasi Nilai-Nilai Pendidikan Islam (Studi Al-Qur'an Surat Al-Fatihah, Surat Luqman Ayat 12, Dan Surat Ibrahim Ayat 7). 17(2).
- Arif, Y. M., & Yulianto, A. (2014). PERGANTIAN SKENARIO OTOMATIS PADA GAME TAJWID MENGGUNAKAN FUZZY SUGENO. *MATICS: Jurnal Ilmu Komputer Dan Teknologi Informasi (Journal of Computer Science and Information Technology)*, 6(2), 101–105. <https://doi.org/10.18860/mat.v6i2.2604>
- Ashlock, D., Brown, J. A., Gregor, C., & Makhmutov, M. (2021). A Family of Turn Based Strategy Games with Moose. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–8. <https://doi.org/10.1109/SSCI50451.2021.9660049>
- Bahtiar, A., Muhima, R. R., & Rachman, A. (2019). Application of the Spiral Model in Platformer Game Design. *National Seminar on Applied Science and Technology VII*, 2(09), 601–606.
- Da Silva Oliveira, F. R., & De Lima Neto, F. B. (2023). Method to Produce More Reasonable Candidate Solutions With Explanations in Intelligent Decision Support Systems. *IEEE Access*, 11, 20861–20876. <https://doi.org/10.1109/ACCESS.2023.3250262>
- Ghasemkhani, N., Vayghan, S. S., Abdollahi, A., Pradhan, B., & Alamri, A. (2020). Urban development modeling using integrated fuzzy systems, ordered weighted averaging (OWA), and geospatial techniques. *Sustainability (Switzerland)*, 12(3). <https://doi.org/10.3390/su12030809>
- H. Ilyas, S. H., St. Hajrah Mansyur, & Harlinda. (2022). Analysis of Learning Web Application Testing With Weighting Product and Ordered Weighting Averaging on Group Decision Maker. *Jurnal Teknik*

Informatika (Jutif), 3(6), 1513–1522.  
<https://doi.org/10.20884/1.jutif.2022.3.6.333>

Juvrud, J., Ansgariusson, G., Selleby, P., & Johansson, M. (2022). Game or Watch: The Effect of Interactivity on Arousal and Engagement in Video Game Media. *IEEE Transactions on Games*, 14(2), 308–317.  
<https://doi.org/10.1109/TG.2021.3073084>

Kaban, R., Syahputra, F., & Fajrillah, F. (2021). Perancangan Game RPG (Role Playing Game) “Nusantara Darkness Rises.” *Journal of Information System Research (JOSH)*, 2(4), 235–246.  
<https://doi.org/10.47065/josh.v2i4.780>

Kementrian Agama RI. (2024a). TafsirWeb Surat Al-Baqarah Ayat 156.  
<https://tafsirweb.com/626-surat-al-baqarah-ayat-156.html>

Kementrian Agama RI. (2024b). TafsirWeb Surat Al-Fatihah Ayat 2.  
<https://tafsirweb.com/46-surat-al-fatihah-ayat-2.html>

Klang, C. E., Enhörning, V., Alvarez, A., & Font, J. (2021). Assessing Simultaneous Action Selection and Complete Information in TAG with Sushi Go! 2021 IEEE Conference on Games (CoG), 1–4.  
<https://doi.org/10.1109/CoG52621.2021.9618987>

Lajnah Pentashihan Mushaf Al-Qur’an. (2022a). Qur’an Kemenag Tafsir Surah Al-Baqarah. <https://quran.kemenag.go.id/quran/per-ayat/surah/2?from=1&to=286>

Lajnah Pentashihan Mushaf Al-Qur’an. (2022b). Qur’an Kemenag Tafsir Surah Al-Maidah. <https://quran.kemenag.go.id/quran/per-ayat/surah/5?from=1&to=120>

Latifah, A., Satria, E., & Iswahyudi Nugraha, A. (2022). Rancang Bangun Role Playing Game Cerita Rakyat Asal Usul Pulomas Berbasis Android. *Jurnal Algoritma*, 19(2), 790–797.  
<https://doi.org/10.33364/algoritma/v.19-2.1234>

Liu, X., & Wang, L. (2020). An extension approach of TOPSIS method with OWAD operator for multiple criteria decision-making. *Granular Computing*, 5(1), 135–148. <https://doi.org/10.1007/s41066-018-0131-4>

Luukka, P., & Stoklasa, J. (2022). Similarity based TOPSIS with linguistic-quantifier based aggregation using OWA. *Recent Advances in Business Analytics. Selected Papers of the 2021 KNOWCON-NSAIS Workshop on Business Analytics*, 29(313396), 45–51.  
<https://doi.org/10.15439/2021b6>

Mutmainah, I., & Yunita, Y. (2021). Penerapan Metode Topsis Dalam Pemilihan Jasa Ekspedisi. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 10(1), 86–92. <https://doi.org/10.32736/sisfokom.v10i1.1028>

- Muttakin, M. Y. A., Wibowo, S. A., & Primaswara P., R. (2020). Game Turn-Based Role Playing Game (Turn-Based Rpg) “Grand Line” Dengan Unity Game Engine Berbasis Android Menggunakan Metode Hierarchical Dynamic Scripting. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 4(2), 254–261. <https://doi.org/10.36040/jati.v4i2.2705>
- Prasetyo, A. A., Suprapedi, S., Narulita, S., & Praharsena, B. (2022). Metode K-Means Berbasis Ordered Weighted Averaging (OWA) pada Data Potensi Desa untuk Penentuan Status Desa. *Jurnal Bingkai Ekonomi (JBE)*, 7(2), 76–91. <https://doi.org/10.54066/jbe.v7i2.224>
- Qomariah, S., Rangan, A. Y., & Amelia Yusnita. (2020). Peningkatan Pengetahuan Pembuatan Game dalam Rangka Pengenalan Industri Kreatif pada Siswa di Madrasah Aliyah An-Nur Samarinda. *Bantenese : Jurnal Pengabdian Masyarakat*, 2(2), 110–119. <https://doi.org/10.30656/ps2pm.v2i2.2872>
- Rinaldi, A., Rahmadani, N., Papilo, P., Silvia, & Rizki, M. (2021). Analisa Pengambilan Keputusan Pemilihan Bahan Dalam. *Jurnal Sains, Teknologi Dan Industri*, 18(2), 163–172.
- Romadhona, F. T., & Yundra, E. (2018). Pengembangan Edugame sebagai Media Pembelajaran Berbasis Role Play Game (RPG) pada Mata Pelajaran Simulasi Digital Kelas X TAV di SMKN 3 Surabaya. *Pendidikan Teknik Elektro*, 07(2), 101–107.
- Setiawansyah, S. (2022). Sistem Pendukung Keputusan Rekomendasi Tempat Wisata Menggunakan Metode TOPSIS. *Jurnal Ilmiah Informatika Dan Ilmu Komputer (JIMA-ILKOM)*, 1(2), 54–62. <https://doi.org/10.58602/jima-ilkom.v1i2.8>
- Shu, Z. (2022). Analysis on ordered weighted averaging operators in different types and applications for decision making. *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 353–359. <https://doi.org/10.1109/ICSP54964.2022.9778323>
- Simangunsong, A., & Hamdani, R. (2020). Rekomendasi Penentuan Desa Terbaik dengan Kombinasi Metode OWA (Ordered Weighted Avarage) dan Metode SAW (Simple Additive Weighting) berbasis Web. *MEANS (Media Informasi Analisa Dan Sistem)*, 5(2), 99–104. <http://dx.doi.org/10.54367/means.v5i2.918>
- Surahaman, A., & Nursadi. (2019). Sistem Pendukung Keputusan Kenaikan Gaji Karyawan Dengan Metode Topsis Berbasis Web. *Jtksi*, 2(3), 82–87. <https://jurnal.ftikomibn.ac.id/index.php/jtksi/article/view/763>
- Sutyawati, Y. S., & Daniawan, B. (2024). Enhancing Employee Motivation: A TOPSIS-Based Decision Support System for Incentive Allocation through Performance Evaluation. *MATICS: Jurnal Ilmu Komputer Dan*

Teknologi Informasi (Journal of Computer Science and Information Technology), 16(1), 13–18. <https://doi.org/10.18860/mat.v16i1.23921>

Tom, M. (2021). Customized functional foods ranking: Application of importance quantifier guided ordered weighted averaging operator. 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2500–2505. <https://doi.org/10.1109/SMC52423.2021.9658951>

Trise Putra, D. W., Santi, S. N., Swara, G. Y., & Yulianti, E. (2020). Metode Topsis Dalam Sistem Pendukung Keputusan Pemilihan Objek Wisata. Jurnal Teknoif Teknik Informatika Institut Teknologi Padang, 8(1), 1–6. <https://doi.org/10.21063/jtif.2020.v8.1.1-6>

Wayan Alit Rangkan Nuaja, I., Cahyawan Wiranatha, Aak., & Suar Wibawa, K. (2022). Rancang Bangun Game Patih Kebo Iwa Bergenre Turn Based Role Playing Game Berbasis Android. JITTER-Jurnal Ilmiah Teknologi Dan Komputer, 3(1).

Wibowo, D. O., & Thyo Priandika, A. (2021). Sistem Pendukung Keputusan Pemilihan Gedung Pernikahan Pada Wilayah Bandar Lampung Menggunakan Metode Topsis. Jurnal Informatika Dan Rekayasa Perangkat Lunak (JATIKA), 2(1), page-page. xx~xx. <http://jim.teknokrat.ac.id/index.php/informatika>

Zhao, S., Dong, Y., Martíne, L., & Pedrycz, W. (2022). Analysis of Ranking Consistency in Linguistic Multiple Attribute Decision Making: The Roles of Granularity and Decision Rules. IEEE Transactions on Fuzzy Systems, 30(7), 2266–2278. <https://doi.org/10.1109/TFUZZ.2021.3078817>