

**REKOMENDASI PEMILIHAN MOTOR PADA *GAME* “BLAR RISING”
MENGUNAKAN METODE TOPSIS BERBASIS BONFERRONI MEAN**

SKRIPSI

**Oleh:
WILDAH NOURINA ALFIANI
NIM. 19650091**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**REKOMENDASI PEMILIHAN MOTOR PADA *GAME* “BLAR RISING”
MENGUNAKAN METODE TOPSIS BERBASIS BONFERRONI MEAN**

SKRIPSI

**Oleh:
WILDAH NOURINA ALFIANI
NIM. 19650091**

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

**REKOMENDASI PEMILIHAN MOTOR PADA GAME “BLAR RISING”
MENGUNAKAN METODE TOPSIS BERBASIS BONFERRONI MEAN**

SKRIPSI

Oleh:
WILDAH NOURINA ALFIANI
NIM. 19650091

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 14 Juni 2023

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

REKOMENDASI PEMILIHAN MOTOR PADA *GAME* “BLAR RISING” MENGUNAKAN METODE TOPSIS BERBASIS BONFERRONI MEAN

SKRIPSI

Oleh:
WILDAH NOURINA ALFIANI
NIM. 19650091

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 18 Oktober 2023

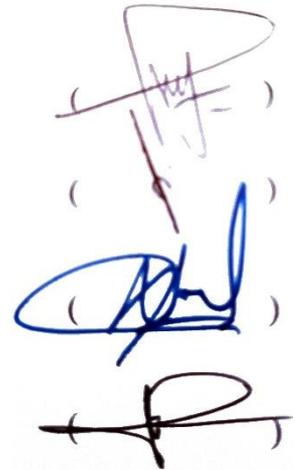
Susunan Dewan Penguji

Ketua Penguji : Dr. Yunifa Miftahul Arif
NIP. 19830616 201101 1 004

Anggota Penguji I : Dr. Muhammad Faisal, M.T
NIP. 19740510 200501 1 007

Anggota Penguji II : Dr. Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

Anggota Penguji III : Dr. Fachrul Kurniawan, M.MT, IPM
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

Saya yang bertanda tangan di bawah ini:

Nama : Wildah Nourina Alfiani

NIM : 19650091

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Rekomendasi Pemilihan Motor pada *Game* “Blar Rising”
Menggunakan Metode TOPSIS Berbasis Bonferroni Mean

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, 18 Oktober 2023
Yang membuat pernyataan,



Wildah Nourina Alfiani
NIM.19650091

MOTTO

“Apapun yang menjadi takdirmu, akan mencari jalannya menemukanmu.”

– Ali bin Abi Thalib

HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Penulis mengucapkan rasa syukur dan terima kasih kepada Allah Subhanahu Wa Ta'ala atas segala rahmat, hidayah, dan karunia-Nya yang telah melimpah dalam menyelesaikan penulisan skripsi ini.

Penulis pesembahkan skripsi ini kepada

Kedua orang tua Abah Moch. Anwar Musadad dan Ibu Roudotul Chusnia serta orang tua kedua penulis selama tinggal di Malang Abi Dr. KH Sutaman, MA dan Umi Qurroti A'yun, S.Ag. Kepada Keluarga tercinta, Seluruh Dosen, teman-teman, sahabat, serta seluruh orang yang telah terlibat dan membantu penulis dalam menyusun skripsi ini.

Terimakasih banyak saya ucapkan

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala, Tuhan Yang Maha Esa, yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul **“Rekomendasi Pemilihan Motor pada *Game* “Blar Rising” Menggunakan Metode TOPSIS Berbasis Bonferroni Mean”** dengan baik.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Informatika di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam penulisan skripsi ini, penulis menyadari bahwa tidaklah mudah. banyak hambatan dan tantangan yang dihadapi sepanjang perjalanan penelitian ini. Namun, berkat dukungan, bimbingan, dan dorongan dari berbagai pihak, penulis berhasil menyelesaikannya.

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT, IPM selaku Ketua Prodi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Dr. Fresy Nugroho, MT dan Dr. Fachrul Kurniawan, M.MT, IPM selaku Dosen Pembimbing skripsi serta Dewan Penguji yang telah memberikan arahan, bimbingan, dan masukan yang sangat berharga dalam penulisan skripsi ini.
5. Abah Moch. Anwar Musadad dan Ibu Roudotul Chusnia selaku orang tua penulis yang telah memberikan dukungan moral, materiil, doa, dan semangat tanpa henti sepanjang perjalanan penulisan skripsi ini. Keberhasilan penulis tidak akan tercapai tanpa cinta dan dukungan mereka.
6. Abi Dr. KH. Sutaman, MA dan Umi Qurroti A'yun, S.Ag selaku Pengasuh Pondok Pesantren Al-Wafa Dinoyo, Malang yang mana berperan sebagai orang tua kedua penulis yang juga memberikan dukungan doa yang tiada henti untuk penulis.
7. Teman seperjuangan Angkatan 2019 Jurusan Teknik Informatika, seluruh teman di Pondok Pesantren Al-Wafa, dan juga teman-teman SMA penulis atas kebersamaan, dukungan, dan persahabatan yang telah terjalin. Setiap momen bersama kalian menjadi sumber kekuatan dan inspirasi bagi penulis.
8. Seluruh Dosen dan Jajaran Staf Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang memberikan ilmu pengetahuan serta pengalaman.
9. Rekan Satriver Studio yang telah memberikan kesempatan untuk *internship* selama 6 bulan sehingga penulis memiliki ilmu pengetahuan tentang *game developing* yang mana sangat berjasa dalam perancangan *game* dalam penelitian ini.

10. Semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan kontribusi, saran, dan dukungan dalam perjalanan penulisan skripsi ini.

Penulis juga menyadari bahwa skripsi ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, penulis dengan rendah hati menerima kritik dan saran yang membangun untuk perbaikan di masa depan.

Malang, 18 Oktober 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN.....	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
المخلص	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Penelitian	5
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	6
BAB II LANDASAN TEORI	7
2.1 <i>Game</i>	7
2.2 <i>Decision Support System</i>	11
2.3 TOPSIS	14
2.4 Bonferroni Mean	17
2.5 TOPSIS Berbasis Bonferroni Mean.....	21
2.6 Pengujian Bonferroni Mean	23
BAB III METODE PENELITIAN	27
3.1 Analisis dan Perancangan <i>Game</i>	27
3.1.1 Analisis <i>Game</i>	27
3.1.2 Perancangan <i>Game</i>	27
3.1.2.1 Rancangan Antarmuka.....	28
3.2 Desain Sistem.....	32
3.3 Rancangan Perhitungan TOPSIS berbasis Bonferroni <i>mean</i>	34
3.3.1 Alternatif	34
3.3.2 Kriteria	34
3.3.3 Matriks Keputusan	36
3.3.4 Perhitungan TOPSIS	37
3.3.4.1 Normalisasi Matriks Keputusan.....	37
3.3.4.2 Normalisasi Terbobot	38
3.3.4.3 Solusi Ideal Positif dan Negatif	38
3.3.4.4 Jarak.....	39
3.3.4.5 Nilai Preferensi	39
3.3.5 Perhitungan TOPSIS Berbasis Bonferroni <i>Mean</i>	40
3.3.5.1 Normalisasi Matriks Keputusan.....	40

3.3.5.2	Normalisasi Terbobot	41
3.3.5.3	Solusi Ideal Positif dan Negatif	41
3.3.5.4	<i>Similarity</i>	42
3.3.5.5	<i>Closeness Coefficient</i>	42
3.4	Implementasi Perhitungan Bonferroni <i>Mean</i>	43
3.5	Desain Pengujian Sistem.....	44
BAB IV HASIL DAN PEMBAHASAN		47
4.1	Implementasi Sistem	47
4.1.1	Implementasi Perhitungan Metode	47
4.1.1.1	Implementasi Perhitungan TOPSIS	48
4.1.1.2	Implementasi Perhitungan TOPSIS-BM	53
4.2	Pengujian Sistem.....	61
4.2.1	Uji Coba <i>Game</i>	61
4.2.2	Hasil Uji Coba.....	64
4.2.2.1	Perbandingan Perhitungan Metode.....	69
4.3	Pengujian <i>Usability</i> pada Fitur Rekomendasi dalam <i>Game</i>	70
4.3.1	Analisa <i>Usability</i> pada Fitur Rekomendasi dalam <i>Game</i>	72
4.4	Integrasi dalam Islam	73
BAB V PENUTUP.....		77
5.1	Kesimpulan	77
5.2	Saran.....	78
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 3.1 Diagram menu	28
Gambar 3.2 Tampilan awal	29
Gambar 3.3 Menu <i>garage</i>	29
Gambar 3.4 Menu <i>tracks</i>	30
Gambar 3.5 Tampilan rekomendasi	30
Gambar 3.6 Menu <i>upgrade</i>	31
Gambar 3.7 <i>Gameplay</i>	31
Gambar 3.8 Blok diagram sistem.....	32
Gambar 3.9 Skenario <i>game</i> dalam FSM	32
Gambar 3.10 Blok diagram metode TOPSIS	37
Gambar 3.11 Blok diagram metode TOPSIS-BM	40
Gambar 3.12 Diagram hasil perhitungan metode	45
Gambar 4.1 Nilai tiap kriteria pada <i>scriptable object</i>	59
Gambar 4.2 Nilai hasil dari tiap alternatif pada <i>scriptable object</i>	60
Gambar 4.3 Simbol rekomendasi	61
Gambar 4.4 Simbol <i>recommended</i>	62
Gambar 4.5 Uji coba pertama	62
Gambar 4.6 Uji coba kedua.....	63
Gambar 4.7 Uji coba ketiga	63
Gambar 4.8 Uji coba keempat.....	64
Gambar 4.9 Diagram hasil dengan medan <i>Snow Hill</i>	68
Gambar 4.10 Diagram hasil dengan medan <i>Mountain</i>	68

DAFTAR TABEL

Tabel 3.1 Tabel kriteria	34
Tabel 3.2 Pembobotan kriteria menggunakan metode ROC.....	35
Tabel 3.3 Skala penilaian kriteria <i>speed, agility, dan handling</i>	35
Tabel 3.4 Skala penilaian kriteria <i>fuel usage</i>	36
Tabel 3.5 Skala penilaian kriteria <i>motor weight</i>	36
Tabel 3.6 Matriks keputusan	37
Tabel 3.7 Normalisasi matriks keputusan metode TOPSIS	38
Tabel 3.8 Normalisasi terbobot metode TOPSIS	38
Tabel 3.9 Solusi ideal positif dan negatif metode TOPSIS.....	38
Tabel 3.10 Jarak	39
Tabel 3.11 Nilai preferensi.....	39
Tabel 3.12 Normalisasi matriks keputusan metode TOPSIS-BM	40
Tabel 3.13 Normalisasi terbobot metode TOPSIS-BM	41
Tabel 3.14 Solusi ideal positif dan negative	42
Tabel 3.15 <i>Similarity</i>	42
Tabel 3.16 <i>Closeness Coefficient</i>	42
Tabel 4.1 <i>Weight List</i>	48
Tabel 4.2 Hasil implementasi perhitungan Normalisasi TOPSIS	49
Tabel 4.3 Hasil implementasi perhitungan Normalisasi Terbobot TOPSIS	50
Tabel 4.4 Hasil implementasi perhitungan PIS dan NIS TOPSIS	51
Tabel 4.5 Hasil implementasi perhitungan Jarak	52
Tabel 4.6 Hasil implementasi perhitungan Nilai Preferensi	53
Tabel 4.7 Hasil implementasi perhitungan Normalisasi TOPSIS-BM	54
Tabel 4.8 Hasil implementasi perhitungan Normalisasi Terbobot TOPSIS-BM..	54
Tabel 4.9 Hasil implementasi perhitungan PIS dan NIS TOPSIS-BM.....	55
Tabel 4.10 Hasil implementasi perhitungan <i>Similarity</i>	57
Tabel 4.11 Hasil implementasi perhitungan <i>Closeness Coefficient</i>	58
Tabel 4.12 Hasil uji coba	62
Tabel 4.13 Input nilai tiap kriteria.....	64
Tabel 4.14 Bobot kriteria ketika medan <i>snow hill</i>	65
Tabel 4.15 Nilai akhir perhitungan (medan <i>Snow Hill</i>)	66
Tabel 4.16 Bobot kriteria ketika medan <i>mountain</i>	66
Tabel 4.17 Nilai akhir perhitungan (medan <i>mountain</i>).....	67
Tabel 4.18 Pertanyaan untuk <i>player</i>	71
Tabel 4.19 Penilaian <i>Usability</i> berdasarkan skor	72
Tabel 4.20 Pengukuran <i>Usability</i>	72

ABSTRAK

Alfiani, Wildah Nourina. 2023. **Rekomendasi Pemilihan Motor pada Game “Blar Rising” Menggunakan Metode TOPSIS Berbasis Bonferroni Mean**. Skripsi. Jurusan 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: TOPSIS, Bonferroni mean, Similarity

Dalam penelitian ini menghadirkan sistem rekomendasi pada menu *motor selection* dalam game yang dirancang “Blar Rising”. Fitur rekomendasi ini menggantikan fitur *randomize* yang kebanyakan digunakan dalam *game* untuk memberi saran item pilihan pada *player* untuk digunakan bermain. Namun, pada dasarnya fitur *randomize* ini hanya mengacak indeks pilihan item sedangkan untuk fitur rekomendasi ini memperhitungkan nilai tiap kriteria yang dimiliki masing-masing pilihan dalam hal ini motor. Penentuan bobot dilakukan menggunakan metode *Rank Order Centroid* (ROC), yang mempertimbangkan tingkat pentingnya kriteria tersebut. Oleh karena itu, dalam game ini terdapat pemilihan *track* atau medan yang digunakan untuk membedakan tingkat kepentingan dalam penentuan bobot. Dalam implementasinya, proses perhitungan rekomendasi pemilihan motor menggunakan metode *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS) yang berbasis *Bonferroni mean* (BM) sebagai operator agregat digunakan untuk menggeneralisasi metode TOPSIS. Setelah dilakukan pada 30 alternatif motor, perhitungan menggunakan metode TOPSIS-BM menghasilkan nilai yang *similar* tiap alternatifnya. Dengan generalisasi menggunakan *Bonferroni mean* menjadikan hasil lebih stabil ketika data kriteria memiliki nilai yang ekstrem. Selain itu, dilakukan pengujian *usability* berdasarkan 5 aspek hingga menghasilkan nilai rata-rata keseluruhan 3.43, dengan rincian *learnability* (kemudahan) 2.95, *efficiency* (efisiensi) 3.87, *memorability* (kemudahan mengingat) 3.67, *error* (kesalahan) 3.20, dan *satisfaction* (kepuasan) 3.45. Ini menunjukkan penerimaan *usability* yang cukup baik oleh *player* untuk fitur rekomendasi.

ABSTRACT

Alfiani, Wildah Nourina. 2023. **Motor Selection Recommendation in the Game “Blar Rising” Using Bonferroni Mean-based TOPSIS Method.** Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Counselor: (I) Dr. Fresy Nughroho, M.T (II) Dr. Fachrul Kurniawan, M.MT, IPM.

This study presents a recommendation system in the motor selection menu of the game "Blar Rising". This recommendation feature replaces the randomize feature commonly used in games to provide players with suggested items for gameplay. However, the randomize feature simply randomizes the index of item choices, while the recommendation feature takes into account the value of each criterion for each option, in this case, the motor. The weighting is determined using the Rank Order Centroid (ROC) method, which considers the importance of each criterion. Therefore, in this game, there is a selection of tracks or terrains used to differentiate the level of importance in determining the weights. In its implementation, the calculation process for motor selection recommendations uses the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method, based on the Bonferroni mean (BM) as an aggregate operator used to generalize the TOPSIS method. After conducting the calculations on 30 alternative motors, the TOPSIS-BM method yields similar values for each alternative. By generalizing using the Bonferroni mean, the results become more stable when the data criteria have extreme values. Additionally, usability testing was conducted based on 5 aspects, resulting in an overall average value of 3.43, with the breakdown as follows: learnability (2.95), efficiency (3.87), memorability (3.67), error (3.20), and satisfaction (3.45). This indicates a reasonably good acceptance of usability by players for the recommendation feature.

Key words: TOPSIS, Bonferroni mean, Similarity

الملخص

ألفياني، ولدة نورينا. ٢٠٢٣. توصية اختيار المحرك في لعبة "Blar Rising" باستخدام طريقة TOPSIS على أساس متوسط بونفيرني. رسالة جامعية. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ. المشرفون: (١) الدكتور فريسي نوغروهو، ماجستير في التكنولوجيا (٢) الدكتور فاخروك كورنياوان، ماجستير في التكنولوجيا، IPM.

في هذه الدراسة، يتم تقديم نظام توصيات في قائمة اختيار المحركات في لعبة "Blar Rising" تعوض ميزة التوصيات هذه ميزة العشوائية التي تستخدم بشكل شائع في الألعاب لتوفير اقتراحات عناصر لاعب للاستخدام في اللعب. ومع ذلك، تقوم ميزة العشوائية في الأساس بتعشيق فهرس الخيارات المحددة، بينما تأخذ ميزة التوصيات في الاعتبار قيمة كل معيار لكل خيار في هذه الحالة المحرك. يتم تحديد الأوزان باستخدام طريقة ترتيب مركز الأوامر (ROC) التي تأخذ في الاعتبار مستوى أهمية هذه المعايير. لذلك، في هذه اللعبة يتوفر اختيار المسارات أو البيئات المستخدمة للتمييز بين مستوى الأهمية في تحديد الأوزان. في التنفيذ، يتم استخدام طريقة Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) المعتمدة على متوسط بونفيرني (BM) كمشغل تجميع لتعميم طريقة TOPSIS في عملية حساب التوصيات لاختيار المحركات. بعد إجراء الحسابات على 30 بديلاً للمحركات، تعطي طريقة TOPSIS-BM قيمةً متشابهة لكل بديل. من خلال التعميم باستخدام متوسط بونفيرني، يصبح النتائج أكثر استقراراً عندما يكون للمعايير قيم متطرفة. بالإضافة إلى ذلك، تم إجراء اختبارات لقابلية الاستخدام استناداً إلى 5 جوانب، مما أدى إلى الحصول على قيمة متوسطة عامة تبلغ 3.43، مع التفصيلات التالية: سهولة التعلم (2.95)، الكفاءة (3.87)، سهولة التذكر (3.67)، الأخطاء (3.20)، والرضا (3.45). هذا يشير إلى قبول جيد بما فيه الكفاية لقابلية الاستخدام من قبل اللاعبين لميزة التوصيات.

كلمات رئيسية: متوسط بونفيرني، التشابه، TOPSIS

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia memiliki minat yang tinggi dalam permainan *game*, terutama dalam sektor *game mobile*. Meskipun pada tahun-tahun sebelumnya pasar *game* di Indonesia relatif kecil, namun saat ini minat tersebut telah meningkat secara signifikan. Penyebab utamanya adalah kemajuan teknologi yang semakin pesat di zaman sekarang, sehingga anak-anak lebih memilih untuk bermain *game* melalui *gadget* mereka daripada permainan tradisional. Perkembangan *game* pada saat ini diantaranya memiliki kualitas grafis yang tinggi, *game mobile* semakin populer, teknologi VR dan AR semakin berkembang, *e-sport* semakin populer, serta *game cross-platform* semakin banyak. Dengan peminat *game* yang tinggi ini dapat membuka peluang besar bagi para pengembang *game* untuk mengembangkan *game* yang bervariasi serta menyediakan pengalaman bermain yang menyenangkan dan interaktif pada pemain (Qarthazanny, 2022).

Untuk menciptakan pengalaman bermain yang menyenangkan, pengembang *game* perlu untuk menambahkan fitur-fitur pada *game* yang bertujuan menambahkan rasa ketagihan dalam bermain. Fitur dalam *game* sangat penting untuk menjaga minat pemain terhadap *game*. Fitur yang ditambahkan biasanya berupa mode permainan, menu *selection*, *achievement-mission*, level permainan, item-item dalam *game*, *customization* dan sebagainya sesuai kebutuhan *game* (Vaghasiya, 2019).

Dari sekian banyak genre *game*, *racing game* atau *game* balap merupakan salah satu yang digemari saat ini. *Game* balap merupakan jenis *game* yang mana menempatkan pemain dalam kontrol kendaraan untuk bersaing dalam perlombaan. Jenis kendaraan yang digunakan dalam *game* balap bisa berupa mobil, motor, atau kendaraan lainnya. *Game* balap memiliki beberapa jenis balapan sirkuit, balapan jalanan, balapan drag, balapan *off-road* dan sebagainya. Setiap jenis balapan memiliki karakteristik yang berbeda dalam hal medan dan rintangan serta kendaraan yang digunakan. Selain itu *game* balap memungkinkan pemain untuk menyesuaikan kendaraan mereka terhadap sirkuit pada *gameplay*, seperti memilih ban yang cocok atau menyesuaikan suspensi untuk meningkatkan kontrol dan kenyamanan dalam berkendara (Michael et al., 2020).

Maka dari itu dalam *game* balap biasanya terdapat menu untuk pemilihan kendaraan yang akan digunakan bermain. Di dalam menu *selection* pemain diperkenankan untuk memilih dan menyesuaikan kendaraan yang akan digunakan dalam permainan yang mana bertujuan untuk meningkatkan kinerja dan kepuasan saat bermain *game*. Dalam *game* balap, menu *selection* ini merupakan salah satu factor untuk memenangkan permainan. Ada beberapa faktor yang diperhatikan dalam memilih kendaraan seperti *speed*, *handling*, *agility*, *acceleration*, *break* dan sebagainya untuk melewati medan dan rintangan yang ada dalam permainan (Behler, 2020).

Dalam menu *selection* di beberapa *game* memiliki sistem *randomize* untuk pemilihan secara acak. Dalam menggunakan *randomize* memungkinkan pemain untuk memilih opsi secara acak untuk menambah variasi dalam bermain. Dalam

beberapa *game*, penggunaan fitur *randomize* juga membantu mempercepat waktu pengambilan keputusan ketika terlalu banyak opsi yang harus dipilih dalam menu *selection*. Namun fitur *randomize* pada umumnya hanya mengacak indeks opsi yang ada tanpa memperhitungkan faktor-faktor penunjang kendaraan.

Peneliti menganggap alih-alih menggunakan sistem *randomize* pada menu *selection*, seharusnya digunakan sistem rekomendasi pada menu *selection* yang bertujuan untuk merekomendasikan satu pilihan motor untuk digunakan *player* bermain. Dengan hal ini dapat memudahkan *player* untuk menentukan pilihannya. Hal ini selaras dengan firman Allah Subhanahu Wa Ta'ala tentang memberi manfaat kepada orang lain pada QS. Al-Qashas ayat 84 yang berbunyi:

مَنْ جَاءَ بِالْحَسَنَةِ فَلَهُ خَيْرٌ مِنْهَا وَمَنْ جَاءَ بِالسَّيِّئَةِ فَلَا يُجْزَى الَّذِينَ عَمِلُوا السَّيِّئَاتِ إِلَّا مَا كَانُوا يَعْمَلُونَ - ٨٤

“Barangsiapa datang dengan (membawa) kebaikan, maka dia akan mendapat (pahala) yang lebih baik daripada kebbaikannya itu; dan barang siapa datang dengan (membawa) kejahatan, maka orang-orang yang telah mengerjakan kejahatan itu hanya diberi balasan (seimbang) dengan apa yang dahulu mereka kerjakan).” QS. Al-Qashas:84

Berdasarkan ayat tersebut menyatakan bahwa setiap perbuatan baik atau buruk yang dilakukan oleh seseorang akan kembali padanya sendiri. Jika seseorang melakukan kebaikan, maka kebaikan tersebut akan menjadi kebaikan bagi dirinya sendiri. Namun, jika seseorang melakukan kejahatan, maka kejahatan tersebut akan menjadi kejahatan bagi dirinya sendiri. Pada akhirnya, setiap orang akan kembali kepada Tuhan mereka untuk mendapatkan balasan atas perbuatan yang mereka lakukan. Oleh karena itu, penting bagi setiap orang untuk melakukan perbuatan yang baik dan menghindari perbuatan yang buruk.

Dengan itu, dalam penelitian ini, sistem rekomendasi pada menu motor *selection* pada game Blar Rising dibuat dimana indeks yang muncul didapatkan melalui proses perhitungan pengambilan keputusan menggunakan metode TOPSIS yang berbasis Bonferroni *mean*. Penelitian sebelumnya tentang metode TOPSIS dilakukan menggunakan *mean* aritmatika standar dalam menghitung nilai agregat dari kesamaan antara vektor atribut dari setiap alternatif yang dianalisis dan solusi ideal (positif dan negatif). Namun, penelitian ini mengusulkan penggunaan operator agregasi yang lebih umum yaitu Bonferroni *mean* untuk meningkatkan agregasi yang dilakukan pada metode Topsis (Luukka & Collan, 2016).

Bonferroni *mean* dipilih sebagai operator agregasi karena merupakan "*mean* superset" dari banyak *mean* yang telah digunakan sebelumnya dan dengan demikian memungkinkan tingkat generalisasi yang lebih tinggi pada metode TOPSIS. Generalisasi TOPSIS berbasis Bonferroni *mean* adalah salah satu metode pengambilan keputusan multikriteria yang menggabungkan teknik TOPSIS dengan teknik generalisasi Bonferroni *mean*. Tujuan dari generalisasi adalah untuk meningkatkan keandalan dan kestabilan hasil pengambilan keputusan multikriteria (Bonferroni, 1950).

Maka pada penelitian ini, TOPSIS berbasis Bonferroni *mean* digunakan untuk menghitung proses pemilihan keputusan untuk sistem rekomendasi pada motor *selection* dalam game Blar Rising. Dengan menambahkan sistem rekomendasi yang mana menggantikan sistem *randomize* pada menu *selection* memiliki keuntungan bagi *player* untuk mengambil keputusan dalam memilih dari sekian banyak pilihan dengan keputusan yang mempertimbangkan kriteria yang ada

pada setiap motor. Namun sistem ini juga memiliki kelemahan dimana menjadikan *player* lebih bergantung pada rekomendasi sehingga kehilangan keterampilan dalam memilih motor secara mandiri.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, peneliti merumuskan masalah sebagai berikut. Bagaimana hasil analisa penentuan alternatif untuk *motor selection* pada game “Blar Rising” menggunakan metode TOPSIS berbasis Bonferroni *mean*?

1.3 Batasan Penelitian

Adapun pembatasan masalah dalam penelitian ini sebagai berikut.

1. Kriteria yang digunakan dalam pemilihan keputusan diantaranya *speed*, *agility*, dan *handling* sebagai nilai *benefit* serta *fuel usage* dan *motor weight* sebagai nilai *cost*.
2. Nilai – nilai kriteria pada setiap motor bersifat *dummy* digunakan untuk keperluan penentuan rekomendasi.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian yang dilakukan adalah menganalisa hasil penentuan alternatif untuk *motor selection* menggunakan perhitungan TOPSIS berbasis Bonferroni *mean*.

1.5 Manfaat Penelitian

Peneliti mengharapkan penelitian ini dapat menghasilkan manfaat sebagai berikut.

1. Memudahkan *player* untuk memilih motor untuk digunakan dalam bermain.
2. Berguna sebagai sumber informasi dalam menentukan rekomendasi pemilihan pada menu *selection* alih-alih menggunakan sistem *randomize*.

BAB II

LANDASAN TEORI

2.1 *Game*

Game merupakan sebuah program yang dibuat sebagai hiburan dimana seorang pemain dapat berinteraksi didalamnya. Sebuah permainan merupakan sebuah sistem dimana pemain akan turut dalam konflik buatan, yang mana pemain akan berinteraksi dengan sistem dan konflik dalam permainan merupakan buatan atau rekayasa semata (Annubaha, 2014).

Menurut (Kaurie et al., 2020) dalam (Jailani & Purwanto, 2019), *game* merujuk pada interaksi antara pengguna dan antarmuka pada perangkat *video* yang menghasilkan umpan balik visual. Perangkat *video* dalam *video game* tradisional biasanya menggunakan tampilan raster. Namun, istilah "permainan *video*" saat ini dapat merujuk pada permainan pada berbagai jenis perangkat layar. Sistem perangkat elektronik yang digunakan sebagai platform untuk bermain *video game* dapat mencakup berbagai tingkatan, mulai dari komputer mainframe hingga perangkat *mobile*.

Menurut (Napitupulu, 2016), *game* adalah sebuah permainan yang terdiri dari karakter dan objek yang dapat digerakkan menggunakan bahasa program untuk menciptakan pergerakan atau perubahan. Beberapa *game* memungkinkan pengguna untuk mengontrol pergerakan karakter dan objek dalam permainan, sementara *game* lainnya menggunakan kecerdasan buatan untuk merespons gerakan sendiri. Sebagai contoh, *game* catur menggunakan kecerdasan buatan dalam permainannya. Bermain game di komputer dapat sangat menyenangkan, tetapi membuat *game*

sendiri dan memungkinkan orang lain memainkannya dapat lebih menyenangkan. Namun, membuat *game* tidaklah mudah, terutama untuk *game* komersial yang dibuat oleh banyak programmer berbakat dengan biaya yang relatif tinggi. Meskipun pembuatan *game* dianggap rumit, namun pembuatan *game* sederhana dapat dilakukan dengan mudah menggunakan Unity 3D dan *action script*.

Menurut (Sari & Naim, 2022), secara mendasar, *game* adalah bentuk permainan yang dapat mengisi waktu luang dengan tujuan hiburan dan memberikan kesenangan bagi pemainnya. *Game* dapat dimainkan secara *offline* oleh satu pemain atau *online* bersama-sama melalui jaringan internet. Bermain *game* dapat dianggap sebagai sebuah aktivitas atau kontes yang memiliki aturan tertentu dan dilakukan oleh orang-orang untuk kesenangan secara fisik atau mental.

Adapun *environment* atau lingkungan serta situasi dan rintangan dalam *game* dibuat seolah-olah seperti kehidupan nyata dimana *player* diminta untuk berusaha mempertahankan permainan hingga tercapai tujuan dari *game* tersebut. *Game* juga bisa dimaknai sebagai arena keputusan dan aksi permainan (Ayu, 2019).

Menurut (Pratama, 2022) mengutip karya Neumann 1953, teori mengenai permainan pertama kali diajukan oleh John Von Neumann dan Oscar Morgenstern, yang menyatakan bahwa "Permainan terdiri dari kumpulan aturan yang menciptakan situasi kompetitif antara dua hingga beberapa orang dengan memilih strategi yang dirancang untuk mencapai kemenangan pribadi atau mengurangi kemenangan lawan. Aturan-aturan ini mengatur kemungkinan tindakan untuk setiap pemain, menerima sejumlah informasi sebagai perkembangan permainan,

dan mengukur sejumlah kemenangan atau kekalahan dalam berbagai situasi yang timbul."

Menurut (Novayani, 2019) *game* berdasarkan genrenya dibedakan menjadi beberapa jenis sebagai berikut.

1. *Shooting* merupakan *game* tembak-menembak yang memerlukan kejelian mata, kecepatan refleks dalam menembak musuh. Jika memiliki tingkat fokus yang rendah atau mudah teralihkn maka bisa tertembak oleh lawan main.
2. *Fighting* merupakan *game* pertarungan dimana memerlukan keahlian dalam menghafalkan tombol untuk mengeluarkan setiap jurus yang ada. Selain itu juga melatih dalam mengatur fokus karena lawan main yang menyerang. Akhir dari *game* jenis ini adalah ketika ada salah satu yang mati.
3. *Adventure* merupakan *game* petualangan dengan tujuan tertentu. Biasanya *player* diminta untuk mengumpulkan koin atau apapun yang bersifat *collectable* yang nantinya akan terakumulasi sebagai skor di akhir *game*. Dalam petualangannya *player* harus melewati berbagai rintangan yang ada termasuk mengalahkan *enemy* yang ada. Akhir *game* jenis ini ketika *player* berhasil menuju *stage* yang telah ditentukan biasanya berupa rumah, bendera, pintu dan lain-lain.
4. *Role Playing Game* atau RPG merupakan *game* yang dimana permainnya bebas menjelajah dunia *game*. RPG memungkinkan pemain untuk berinteraksi terhadap karakter lain bahkan dapat bertarung satu sama lain. Biasanya pemain bisa menentukan akhir dari jalan cerita di beberapa *game* RPG.

5. *Simulation* merupakan *game* yang menggambarkan lingkungan kehidupan seolah-olah nyata. Biasanya berupa *game* yang *player*-nya menjalani kehidupan misalnya berkebun, bekerja, serta aktivitas manusia lainnya. Pada saat sekarang *game* simulasi ini telah berkembang menggunakan teknologi metaverse yaitu dunia virtual 3D yang mewakili kehidupan nyata.
6. *Strategy* merupakan *game* yang lebih menekankan dalam pengaturan strategi para pemain. Diperlukan juga kemampuan memimpin yang baik untuk sebuah pasukan. Mayoritas *game* genre ini adalah *game* peperangan.
7. *Sport* merupakan *game* yang memiliki *gameplay* berbagai jenis olahraga. Bisa dikatakan dalam *game* sport para pemain melakukan pertandingan olahraga secara virtual. *Game* ini dihadirkan pada orang-orang yang gemar berolahraga atau sekedar gemar menonton pertandingan olahraga. Dibutuhkan kelincahan serta mengatur strategi dalam bermain layaknya bertanding sungguhan.
8. *Racing* merupakan *game* membalap atau kemampuan menyetir motor maupun mobil. Orang yang memainkan *game* jenis ini dapat terpacu adrenalinnya karena berusaha menjadi yang tercepat di sirkuit. Akhir dari *game* ini adalah ketika *player* telah mencapai garis *finish*

Dalam (Fathurrahman, 2020) menjabarkan dampak positif dan negatif dari sebuah *game* bagi para pemainnya. Dampak positifnya meliputi fungsi sebagai penenang stres setelah hari yang melelahkan, terutama setelah beraktivitas keras sepanjang hari. Selain itu, permainan juga dapat meningkatkan kecerdasan dan respon otak, terutama pada anak-anak. Namun, di sisi lain, ada dampak negatifnya. Misalnya, terlalu sering bermain *game* dapat mengakibatkan pengabaian terhadap

tugas-tugas lain yang akhirnya tertunda, dan bermain game dalam durasi yang berlebihan dapat merusak kesehatan mata.

2.2 Decision Support System

Menurut (Turban et al., 2017) *Decision Support System* atau yang dikenal dengan Sistem Pendukung Keputusan (SPK) merupakan sebuah sistem yang membantu pengambilan keputusan dengan menghasilkan alternatif yang diinginkan, kemudian membantu pengambil keputusan untuk mengevaluasi dan memilih alternatif yang paling sesuai. SPK tidak hanya berfokus pada penggunaan data, tetapi juga dapat menggunakan kecerdasan buatan, seperti machine learning dan neural network, untuk meningkatkan akurasi dan kecepatan pengambilan keputusan. Dengan menggabungkan teknologi informasi dan metode analisis data, SPK dapat memberikan informasi yang tepat waktu dan akurat yang akan membantu seseorang dalam memilih opsi terbaik. Dalam bisnis, SPK dapat digunakan untuk memprediksi hasil keuangan atau membuat keputusan investasi, sedangkan dalam industri kesehatan, SPK dapat digunakan untuk membantu dokter dalam diagnosis penyakit dan meresepkan obat yang tepat. Dalam semua situasi, SPK membantu seseorang dalam membuat keputusan yang terinformasi dengan baik dan memperkecil risiko kesalahan.

Menurut (Hou et al., 2023) DSS merupakan sebuah alat ilmiah yang digunakan untuk membantu pengambilan keputusan dalam bentuk tertentu. DSS memberikan lingkungan kerja bagi para pengambil keputusan yang menggabungkan pengetahuan, inisiatif, kreativitas, dan kemampuan pengolahan informasi, serta menggabungkan metode kualitatif dan kuantitatif melalui dialog

manusia-komputer. DSS membantu para pengambil keputusan untuk menganalisis masalah, mengeksplorasi metode pengambilan keputusan, dan melakukan evaluasi, prediksi, dan optimasi.

SPK memiliki beberapa tipe yang dibedakan menjadi 2 macam diantaranya sistem yang berorientasi pada data yang merupakan sistem yang memberikan fungsi untuk pemanggilan, analisa, dan presentasi data. Sedangkan tipe lain yaitu sistem yang beorientasi pada model yang merupakan sistem yang memberikan fungsi pada model seperti model simulasi, akutansi, optimasi yang dapat membantu manajemen dan membuat sebuah keputusan (Pratiwi, 2020).

Di dalam bukunya (Rahmansyah & Lusinia, 2016) menjabarkan beberapa metode perhitungan SPK diantaranya.

1. Metode Logika Fuzzy

Metode ini digunakan untuk mengatasi ketidakpastian dan ketidakjelasan dalam pengambilan keputusan dengan memodelkan konsep-konsep seperti "tinggi", "sedang", dan "rendah" dalam skala keputusan yang fuzzy. Dalam DSS, logika fuzzy sering digunakan untuk memetakan data numerik ke dalam kategori-kategori ini dan kemudian menggunakan aturan fuzzy untuk memperkirakan hasil

2. Metode TOPSIS (*Technique for Order of Preference by Similarity to Ideal Solution*)

Metode ini digunakan untuk memilih alternatif terbaik dari sekelompok alternatif dengan membandingkan jarak antara setiap alternatif dan titik ideal positif dan negatif. TOPSIS menghitung nilai kedekatan relatif antara setiap

alternatif dan kriteria, dan kemudian memilih alternatif yang paling dekat dengan titik ideal positif dan paling jauh dari titik ideal negatif.

3. Metode AHP (*Analytical Hierarchy Process*)

Metode ini digunakan untuk memilih alternatif terbaik dengan mempertimbangkan beberapa kriteria dan sub-kriteria yang berbeda. Dalam AHP, kriteria dan sub-kriteria diberi bobot berdasarkan kepentingannya, dan kemudian setiap alternatif dinilai terhadap setiap kriteria dan sub-kriteria. Nilai akhir kemudian dihitung berdasarkan bobot relatif dari setiap kriteria dan sub-kriteria.

4. Metode SAW (*Simple Additive Weighting*)

Metode ini digunakan untuk memilih alternatif terbaik dalam situasi yang kompleks dengan menentukan bobot untuk setiap kriteria dan sub-kriteria. Setiap alternatif kemudian dinilai terhadap setiap kriteria dan sub-kriteria, dan nilai akhir kemudian dihitung dengan menambahkan setiap nilai kriteria yang telah dinormalisasi dan dikalikan dengan bobotnya.

5. Metode Moving Average

Metode ini digunakan untuk memprediksi data masa depan berdasarkan data masa lalu dengan menggunakan rata-rata pergerakan dari sejumlah titik data sebelumnya. Moving average sering digunakan dalam DSS untuk memperkirakan permintaan atau penjualan masa depan, yang merupakan faktor penting dalam pengambilan keputusan bisnis.

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

TOPSIS merupakan salah satu metode untuk mengambil keputusan multikriteria yang digunakan untuk menentukan alternatif yang terbaik dari beberapa pilihan yang ada. Metode ini pertama diperkenalkan oleh Hwang dan Yoon pada tahun 1981 dan sekarang telah digunakan di berbagai bidang seperti industri, manajemen, kesehatan. Menurut (Abror, 2017) metode TOPSIS berdasarkan konsep bahwa alternatif terbaik tidak hanya ditentukan oleh jarak terdekat dari solusi ideal positif, tetapi juga oleh jarak terjauh dari solusi ideal negatif.

Menurut (Wen et al., 2023), TOPSIS adalah sebuah metode yang digunakan untuk menentukan solusi optimal dari suatu fungsi dengan menemukan titik terdekat dari titik ideal positif dan terjauh dari titik ideal negatif. Metode ini berfokus pada perhitungan solusi ideal positif dan solusi ideal negatif dari suatu masalah evaluasi. Kemudian, dengan menghitung kedekatan relatif dari setiap skema dengan skema ideal, skema tersebut diurutkan, dan skema yang paling cocok dipilih.

Dalam penggunaan TOPSIS, penting untuk menentukan kriteria-kriteria yang relevan dan mengumpulkan data yang akurat untuk digunakan dalam analisis. Selain itu, penggunaan bobot yang tepat untuk setiap kriteria juga sangat penting untuk mendapatkan hasil yang akurat dan bermanfaat. Metode TOPSIS memiliki beberapa kelebihan, di antaranya adalah konsep yang mudah dipahami dan jelas, akurasi dalam perhitungan, dan kemampuan untuk mengevaluasi kinerja relatif dari alternatif keputusan dalam bentuk matematis yang sederhana (Gajah, 2019).

Dalam penelitian yang dilakukan oleh (Novriantama, 2022) menggunakan TOPSIS berbobot dinamis untuk memilih skenario game dalam suatu *game*. Bobot kepentingan kriteria digunakan sebagai input dalam metode TOPSIS berbobot dinamis untuk menentukan pilihan skenario game dari berbagai destinasi wisata. Pilihan skenario ini kemudian menjadi alternatif tempat wisata yang dapat dijelajahi oleh pengguna, yang pada akhirnya mempermudah pengguna dalam menentukan destinasi wisata yang sesuai dengan preferensi mereka. Sehingga penelitiannya bertujuan untuk meningkatkan pengalaman pengguna dalam menentukan tujuan wisata yang sesuai dengan harapan mereka melalui penggunaan metode TOPSIS berbobot dinamis dalam permainan.

Berikut merupakan langkah perhitungan metode TOPSIS menurut (Arif et al., 2019) dalam (Roszkowska, 2011) dan (Kaur et al., 2017)

- a. Menghitung Normalisasi r_{ij} dari matriks keputusan x_{ij}

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.1)$$

Dimana $i = 1, 2, 3, \dots, m$ dan $j = 1, 2, 3, \dots, n$

- b. Menentukan solusi ideal positif dan negatif (A^+ dan A^-) didapatkan berdasarkan rating bobot ternormalisasi

$$y_{ij} = w_i \cdot r_{ij} \quad (2.2)$$

Dimana $i = 1, 2, 3, \dots, m$ dan $j = 1, 2, 3, \dots, n$

y_{ij} = Normalisasi terbobot, w_i = bobot kriteria, r_{ij} = normalisasi

- c. Menentukan nilai maksimum dan minimum dari matriks normalisasi terbobot (A^+ dan A^-)

$$A^+ = (y_1^+, y_2^+, y_3^+, \dots, y_n^+) \quad (2.3)$$

$$A^- = (y_1^-, y_2^-, y_3^-, \dots, y_n^-)$$

dengan

$$y_j^+ = \begin{cases} \max_i y_{ij}; \text{ jika } j \text{ atribut benefit} \\ \min_i y_{ij}; \text{ jika } j \text{ adalah atribut cost} \end{cases} \quad (2.4)$$

$$y_j^- = \begin{cases} \max_i y_{ij}; \text{ jika } j \text{ atribut cost} \\ \min_i y_{ij}; \text{ jika } j \text{ adalah atribut benefit} \end{cases}$$

$j = 1, 2, \dots, n$

y^+ = nilai maksimal dari normalisasi terbobot setiap alternatif dari masing-masing kriteria

y^- = nilai minimal dari normalisasi terbobot setiap alternatif dari masing-masing kriteria

- d. Menghitung jarak antara nilai setiap alternatif dengan matriks solusi ideal positif maupun negatif (D^+ dan D^-)

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2} \quad i = 1, 2, 3, \dots, m \quad (2.5)$$

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2} \quad i = 1, 2, 3, \dots, m \quad (2.6)$$

- e. Menghitung nilai preferensi untuk setiap alternatif

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (2.7)$$

Nilai preferensi alternatif yang paling besar membuktikan bahwa alternatif tersebut terpilih dan merupakan yang terbaik dari alternatif-alternatif lain.

2.4 Bonferroni Mean

Bonferroni *mean* (BM) adalah salah satu metode agregasi data dalam pengambilan keputusan multi-kriteria yang banyak digunakan. Dalam penelitian yang dilakukan oleh (Yager, 2009) menggunakan Bonferroni *mean* untuk tujuan meningkatkan kemampuan pemodelan dalam konteks masalah multi-kriteria. Masalah yang melibatkan multi-kriteria sangat umum dalam banyak bidang teknologi modern, seperti pengambilan keputusan, pengenalan pola, pengambilan informasi, penalaran berbasis kasus, dan kueri basis data, di antara lainnya. Penulis berkontribusi dalam teori agregasi dengan mengkaji Bonferroni *mean* operator dan mengusulkan beberapa generalisasi yang meningkatkan kemampuan pemodelannya. Penulis memberikan interpretasi operator ini sebagai hasil perkalian antara setiap argumen dengan rata-rata dari argumen lainnya, yang merupakan kombinasi dari operator rata-rata dan operator 'anding'. Hal ini memungkinkan penulis untuk mengusulkan generalisasi dari operator ini dengan mengganti rata-rata sederhana dengan jenis operator rata-rata lainnya seperti operator OWA dan integral Choquet serta mengaitkan bobot yang berbeda pada argumen-argumen tersebut. Penulis menyimpulkan bahwa berbagai ekstensi dari Bonferroni *mean* dapat memodelkan tingkat konjungsi parsial yang keras dan lembut yang berbeda-beda.

Selanjutnya pada penelitian yang dilakukan oleh (Shaheen & Toor, 2022) menggunakan Bonferroni *mean* dalam pengambilan keputusan multiatribut (MADM) berdasarkan Fuzzy berbasis square root fuzzy set (SR-FS). SR-FS merupakan salah satu generalisasi efisien dari himpunan fuzzy intuisi yang

digunakan untuk mengatasi ketidakpastian dan kabur dalam informasi. Dalam penelitian ini, metode baru untuk MADM berbasis informasi SR-Fuzzy diselidiki. Operator agregasi memegang peranan penting dalam proses pengambilan keputusan. Oleh karena itu, penelitian ini mengusulkan berbagai operator Bonferroni Mean dan Weighted Bonferroni Mean yang baru untuk menggabungkan nilai SR-Fuzzy sesuai dengan preferensi pengambil keputusan yang berbeda-beda. Operator SR-Fuzzy Bonferroni *Mean* (SRFBM) dan operator *weighted* SR-Fuzzy Bonferroni Mean (WSRFBM) didirikan dan properti-propertinya dijelaskan. Selanjutnya, pendekatan MADM menggunakan operator-operator yang diusulkan untuk informasi SR-Fuzzy dikembangkan dan diuji dengan contoh matematika. Pada akhirnya, studi perbandingan antara pendekatan yang dikembangkan dan pendekatan yang sudah ada dibahas untuk mengevaluasi kelayakan dan kepraktisan dari teknik pengambilan keputusan yang diusulkan.

Menurut (Luukka & Collan, 2016) dalam (Yager, 2009) dan (Beliakov et al., 2010) Bonferroni *mean* dalam penelitian belum banyak digunakan oleh komunitas peneliti sampai akhir-akhir ini. Metode ini kemudian mulai dikembangkan dengan melakukan generalisasi terhadap metode yang menggunakan rata-rata (*mean*) sebagai acuan. Hal ini menunjukkan bahwa para peneliti mulai melihat potensi metode tersebut dan mulai mempelajarinya secara lebih mendalam.

Bonferroni *mean* operator digunakan untuk menggambarkan interaksi antara kriteria dalam pengambilan keputusan multi-kriteria. Namun, metode-metode yang ada saat ini cenderung bergantung pada pengaturan subjektif dari

pembuat keputusan untuk menentukan interaksi atau independensi antara kriteria. Contohnya, dalam menentukan derajat interaksi antara kriteria C_i dan C_j , serta apakah kriteria C_i dan C_j independen, keputusan tersebut seringkali diputuskan oleh para ahli atau pembuat keputusan berdasarkan pandangan subjektif mereka. Hal ini dapat menyebabkan perbedaan hasil yang signifikan dan memperumit proses pengambilan keputusan. Dalam konteks ini, perlu dilakukan pengembangan metode yang dapat mengurangi tingkat subjektivitas dan meningkatkan akurasi dalam menentukan interaksi antara kriteria (Yang et al., 2022).

Menurut (Luukka & Collan, 2016), Bonferroni *Mean* didefinisikan sebagai berikut.

Definisi 1. Misalkan $p, q \leq 0$ dan $x_i \geq 0, i = 1, \dots, n$. Bonferroni *mean* merupakan sebuah fungsi.

$$B^{p,q}(x) = \left(\frac{1}{n(n-1)} \sum_{i,j=1, i \neq j}^n x_i^p x_j^q \right)^{1/(p+q)} \quad (2.8)$$

Dari rumus 2.8, jika $p=0$, maka kita akan mendapatkan *mean* yang biasa disebut sebagai generalisasi rata-rata, atau disebut juga sebagai *mean* kuadratik atau *mean* aritmatik-geometrik. Konsep ini digunakan untuk menghitung rata-rata dari sebuah rangkaian bilangan dengan memberikan bobot pada masing-masing bilangan. Jika $n=2$ dan $p=q$, maka Bonferroni *mean* akan sama dengan *geometric mean*. *Geometric mean* adalah jenis *mean* yang digunakan untuk menghitung rata-rata dari sekelompok bilangan positif dengan menggunakan operasi perkalian. Bonferroni *mean* terdiri dari dua rata-rata. Konsep ini merupakan generalisasi dari rata-rata aritmatik dan geometrik. Dalam literatur, terdapat beberapa definisi yang berbeda

untuk "*similarity*" atau kesamaan. Ini disebabkan oleh fakta bahwa subjek ini dapat dianggap dari beberapa sudut pandang yang berbeda, seperti sudut pandang teori himpunan, atau berdasarkan pada logika *multi-value*. Konsep kesamaan ini banyak digunakan dalam matematika dan ilmu komputer untuk membandingkan objek-objek yang berbeda atau mengklasifikasikan data.

Menurut (Zadeh, 1970) dalam (Luukka & Collan, 2016) memberikan definisi untuk *similarity* tahun 1971:

Definisi 2. *Similarity relation* S dalam himpunan X adalah sebuah *fuzzy relation* yang digunakan untuk membandingkan tingkat *similarity* antara elemen-elemen dalam himpunan X , yang berarti:

- a. *Reflexive* yaitu $S(x, x) = 1, \forall x \in X$
- b. *Symmetric* yaitu $S(x, y) = S(y, x), \forall x, y \in X$
- c. *Transitive* yaitu $S(x, z) \geq \sup_y (S(x, y) * S(y, z)), \forall x, y \in X$

Transitivity dalam definisi ke 2 merupakan bentuk transivitas ‘max-star’ yang merupakan salah satu dari bentuk transivitas yang ada dalam literatur. Ukuran *similarity* yang umum digunakan sebagai berikut.

$$S(x, y) = 1 - |x - y| \quad (2.9)$$

Berdasarkan rumus ukuran *similarity* 2.9, jika diteliti *similarity* $S_i, i = 1, \dots, n$ dalam himpunan X , dapat mendefinisikan *binary relation* dengan menetapkan $S(x, y) = \frac{1}{n} \sum_{i=1}^n S_i(x_i, y_i), f or all x, y \in X$ (Turunen, 1999). Kemudian jika

menggunakan Bonferroni *mean* daripada Arithmetic *mean*, didapatkan *similarity* untuk $S(x, y)$ sebagai berikut.

$$S(x, y) = \left(\frac{1}{n(n-1)} \sum_{i,j=1, i \neq j}^n s_i^p s_j^q \right)^{1/(p+q)} \quad (2.10)$$

dimana $s_i = s(x_i, y_i)$, $s_j = s(x_j, y_j)$, $\mathbf{x} = x(x_1, \dots, x_n)$, $\mathbf{y} = y(y_1, \dots, y_n)$. Dan bisa juga dituliskan dengan rumus berikut.

$$S(x, y) = \left(\frac{1}{n} \sum_{i=1}^n s_i^p \left(\frac{1}{n-1} \sum_{j=1}^n s_j^q \right) \right)^{1/(p+q)} \quad (2.11)$$

2.5 TOPSIS Berbasis Bonferroni Mean

Bonferroni *mean* operator digunakan untuk menggambarkan interaksi antara kriteria dalam pengambilan keputusan multi-kriteria. Namun, metode-metode yang ada saat ini cenderung bergantung.

Di dalam penelitian ini, penentuan keputusan menggunakan varian baru dari TOPSIS yaitu perhitungan TOPSIS yang didasarkan pada Bonferroni *mean*. Dalam metode TOPSIS yang asli, dihitung jarak dari alternatif yang diteliti ke solusi ideal positif dan negatif, dan dipilih alternatif terbaik yang memiliki kedekatan relatif paling tinggi dengan solusi ideal positif dan jarak relatif terpanjang dari solusi ideal negatif.

Pada perhitungan varian baru TOPSIS ini menggunakan kesamaan alternatif dengan solusi ideal positif dan negatif sebagai ukuran keberhasilan alternatif. Metode ini didasarkan pada konsep kesamaan Bonferroni, yang

mempertimbangkan semua pasangan kriteria secara bersamaan. Dalam metode ini, alternatif terbaik dipilih berdasarkan kesamaan Bonferroni relatif terbesar dengan solusi ideal positif dan kesamaan Bonferroni relatif terkecil dengan solusi ideal negatif (Luukka & Collan, 2016).

Dalam kedua metode, solusi ideal positif dan solusi ideal negatif merepresentasikan kondisi ideal dan kondisi yang harus dihindari atau tidak diinginkan. Namun, varian baru ini mengubah ukuran keberhasilan alternatif dari jarak menjadi kesamaan, sehingga dapat memberikan hasil yang berbeda dengan metode asli.

Perhitungan menggunakan TOPSIS berbasis Bonferroni *mean* mulai dari penentuan matrix $X = [x_{ij}]$. Dimana x_{ij} merupakan nilai matriks keputusan, $i =$ alternatif dan $j =$ kriteria. Langkah-langkah perhitungan sebagai berikut (Luukka & Collan, 2016).

Langkah 1. Normalisasi matriks keputusan $R = [r_{ij}]$

$$r_{ij} = \frac{x_{ij} + |\min(x_{ij})_i|}{\max(x_{ij})_i + \min(x_{ij})_i}, i = 1, \dots, m; j = 1, \dots, n \quad (2.12)$$

Langkah 2. Menghitung normalisasi terbobot matriks $V = [v_{ij}]$

$$v_{ij} = r_{ij} \cdot w_i, j = 1, \dots, m; i = 1, \dots, n \quad (2.13)$$

Langkah 3. Penentuan solusi ideal positif dan negative

$$\begin{aligned} A^+ &= \{v_1^+, \dots, v_m^+\} = \{(max_i v_{ij} | j \in B), (min_i v_{ij} | j \in C)\} \\ A^- &= \{v_1^-, \dots, v_m^-\} = \{(min_i v_{ij} | j \in B), (max_i v_{ij} | j \in C)\} \end{aligned} \quad (2.14)$$

Dimana B untuk kriteria *benefit* dan C untuk kriteria *cost*

Langkah 4. Menghitung *similarity* dari setiap alternatif solusi ideal positif dan negatif

$$s_i^+ = \left(\frac{1}{m(m-1)} \sum_{k,l=1, k \neq l}^m s_k^{+p} s_l^{+q} \right)^{1/(p+q)}, i = 1, \dots, n \quad (2.15)$$

dimana $s_k^+ = 1 - |v_{ik} - v_k^+|$ (s_l *similarity*)

$$s_i^- = \left(\frac{1}{m(m-1)} \sum_{k,l=1, k \neq l}^m s_k^{-p} s_l^{-q} \right)^{1/(p+q)}, i = 1, \dots, n \quad (2.16)$$

dimana $s_k^- = 1 - |v_{ik} - v_k^-|$ (s_l *similarity*)

Langkah 5. Menghitung nilai *closeness coefficient* tiap alternatif

$$CC_i = \frac{s_i^+}{s_i^+ + s_i^-}, i = 1, \dots, n \quad (2.17)$$

Langkah 6. Ranking hasil nilai *closeness coefficient* tiap alternatif, dimana alternatif dengan nilai *closeness coefficient* tertinggi merupakan keputusan alternatif yang didapatkan.

2.6 Pengujian Bonferroni Mean

Penelitian yang dilakukan oleh (Luukka & Collan, 2016), perankingan Bonferroni *mean* dilakukan menggunakan *Histogram Ranking method*. Menurutny dalam (Collan & Luukka, 2014), metode ini bekerja dengan cara membuat pengurutan semua alternatif dengan rentang nilai parameter diskrit yang bervariasi dari nilai minimum hingga maksimum, atau dalam kasus beberapa parameter dengan kombinasi nilai parameter diskrit dari minimum hingga maksimum. Kemudian dengan membuat histogram terpisah dari peringkat yang diterima setiap alternatif dan dengan menghitung pusat gravitasi untuk setiap

"ranking histogram". Akhirnya, pengurutan alternatif dilakukan berdasarkan pusat gravitasi yang dihitung sebelumnya. Dengan metode ini, pembentukan histogram dapat dilakukan tanpa memperhatikan nilai parameter tertentu, sehingga menghasilkan pengurutan alternatif yang tidak tergantung pada nilai parameter yang dipilih.

Dalam perhitungan Bonferroni *mean* yang dilakukan (Luukka & Collan, 2016) berikut langkah-langkahnya.

Langkah 1. Normalisasi

Pseudocode 2.1 Fungsi Normalisasi

```
Function normalization(data)
mins_mk = min(data);
maks_mk = max(data);
norm = (data+(abs(mins_mk)))*((mins_mk + maks_mk)^(-1));
return norm;
```

Pseudocode 2.1 ini berfungsi untuk menghitung normalisasi terhadap data matriks keputusan yang diberikan

Langkah 2. Perhitungan Bonferroni *Mean*

Pseudocode 2.2 Fungsi Bonferroni Mean

```
function Bonmean(x, ml, q)
[n,m]=x(size)
for j=1:m
for i=1:n
xn=x(j,[1:i-1,i+1:end]);
tmp1(i)=sum(xn.^q,2)/(n-1);
end
y(j) = (sum((x(j,:).^ml).*tmp1,2)/(n)).^(1/(ml+q));
end
return y;
```

Pseudocode 2.2 untuk melakukan perhitungan Bonferroni *mean*.

Langkah 3. Perhitungan *Similarity*

Pseudocode 2.3 Fungsi *similarity*

```
function simLPBon(center, data, p, m1, q)
[n, m]=size(data);
for i=1:n
    for j=1:m
        simM(i, j)=1-abs(data(i, j)^p - center^p);
    end
end
totsim = Bonmean(simM, m1, q);
return totsime;
```

Pseudocode 2.3 untuk menghitung nilai *similarity* dimana nilainya akan memproses fungsi Bonmean().

Langkah 4. Perhitungan solusi ideal positif dan negatif (PIS dan NIS)

Pseudocode 2.4 Fungsi PIS NIS

```
function [cc, PISB, NISB]=topsisBM(data, p, m1, q, crit, we)
[m, n]=size(data);
nr = normalization(data);
a= we.*nr;
for j=1:n
    if crit(j)==1
        PIS(j)=max(a(:, j));
        NIS(j)=min(a(:, j));
    else
        PIS(j)=min(a(:, j));
        NIS(j)=max(a(:, j));
    end
end
end
```

Pseudocode 2.4 untuk menentukan nilai solusi ideal positif dan negatif dari matriks yang ternormalisasi

Langkah 5. Perhitungan *similarities* dari solusi ideal positif dan negative

Pseudocode 2.5 Fungsi PISB NISB

```
PISB=simLPBon(PIS, a, p, m1, q)';
NISB=simLPBon(NIS, a, p, m1, q)';
```

Dengan menggunakan fungsi simLPBon() dapat menghitung *similarities* solusi ideal positif dan negatif (PISN dan NISB)

Langkah 4. Perhitungan *Closeness Coefficient**Pseudocode 2.6 Fungsi Closeness Coefficient*

```
cc=zeros(1,m);  
for i=1:m  
    cc(i)=PISB(i)/(PISB(i)+NISB(i));  
end
```

Pseudocode 2.6 ini akan melakukan perhitungan untuk nilai *closeness coefficient* pada masing-masing alternatif menggunakan nilai PISB dan NISB yang telah didapatkan sebelumnya.

BAB III

METODE PENELITIAN

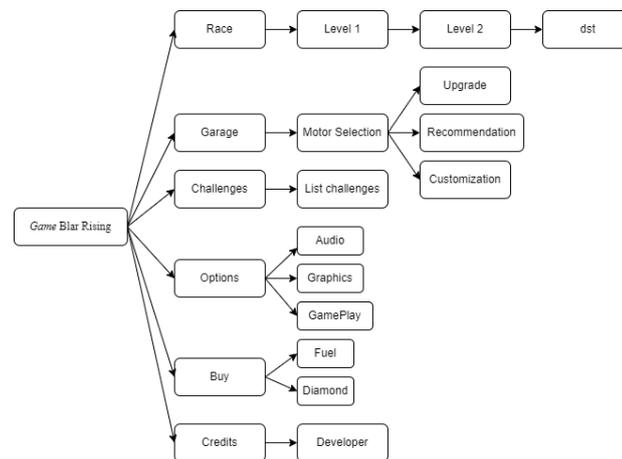
3.1 Analisis dan Perancangan Game

3.1.1 Analisis Game

Game “Blar Rising” merupakan *game* 2.5 dimensi dengan genre *racing* yang dimainkan *single player*. *Player* pada *game* ini diharuskan untuk mengendarai motor hingga akhir level dengan melalui rintangan landasan yang ada. Di dalam *game* ini menggunakan 4 kontrol yaitu *acceleration*, *brake*, *right*, dan *left* . Pada saat motor berada di udara, menekan *right* akan menyebabkan motor untuk berputar ke depan melakukan *front flip* dan sebaliknya jika menekan *left* saat berada di udara maka motor akan berputar ke belakang melakukan *back flip*. Ketika motor berhasil untuk *front flip* atau *back flip* tanpa terjatuh maka pemain akan mendapatkan poin. Pemain diberikan 5 kali *retry* untuk melanjutkan permainan ketika terjatuh, jika *retry count* habis maka permainan berakhir atau *gameover*. Akhir dari *game* ini adalah garis *finish* di ujung *track* dan akan menakumulasi poin yang didapatkan selama bermain serta juga mempertimbangkan waktu paling sedikit dalam menyelesaikan level untuk menjadikannya *high score*.

3.1.2 Perancangan Game

Pada *game* “Blar Rising” memiliki beberapa menu utama pada tampilan awal diantaranya *race*, *garage*, *challenges*, *options*, dan *credits*. Dapat digambarkan dalam gambar 3.1.



Gambar 3.1 Diagram menu

Berdasarkan diagram menu pada gambar 3.1, penerapan perhitungan TOPSIS berbasis Bonferroni *mean* digunakan dalam pemilihan motor pada menu motor *selection* yang mana dalam menu tersebut terdapat pilihan motor dengan nilai kriteria yang berbeda-beda. Dengan sekali menekan *button* rekomendasi maka akan ditampilkan motor yang menjadi rekomendasi melalui perhitungan TOPSIS berbasis Bonferroni *mean*.

3.1.2.1 Rancangan Antarmuka

a. Tampilan awal *game*

Pada tampilan awal *game* (gambar 3.2) menampilkan *main menu* yang terdiri dari *garage*, *challenges*, *race*, *tracks*, dan *options*.



Gambar 3.2 Tampilan awal

b. Menu *Garage*

Pada menu *garage* (gambar 3.3) menampilkan pilihan motor yang memiliki spesifikasi yang berbeda-beda. Dalam menu *garage* memiliki fitur untuk *customization* motor, *upgrade* spesifikasi motor, serta fitur untuk merekomendasikan motor dari sekian variasi motor sebagai pilihan alternatif untuk digunakan dalam bermain.

Gambar 3.3 Menu *garage*

c. Menu *Tracks*

Menu *tracks* (gambar 3.4) menampilkan pilihan *track* atau medan yang akan digunakan dalam bermain. Dalam pemilihan medan ini mempengaruhi

rekomendasi yang diberikan. Hal ini dikarenakan setiap medan memiliki tingkat kepentingan kriteria/spesifikasi motor yang berbeda.



Gambar 3.4 Menu tracks

d. Tampilan Rekomendasi

Fitur rekomendasi akan merekomendasikan sebuah pilihan motor menggunakan perhitungan metode TOPSIS berbasis Bonferroni *mean* yang akan menghasilkan sebuah alternatif untuk digunakan dalam bermain. Ketika menggunakan fitur ini maka motor rekomendasi akan ditampilkan seperti pada gambar 3.5.



Gambar 3.5 Tampilan rekomendasi

f. *Menu Upgrade*

Pada menu *upgrade* (gambar 3.6) bertujuan untuk meng-*upgrade* kriteria dari sebuah motor dengan cara membelinya. Dengan dilakukan *upgrade* dapat meningkatkan performa dari motor tersebut.



Gambar 3.6 Menu *upgrade*

g. *Gameplay*

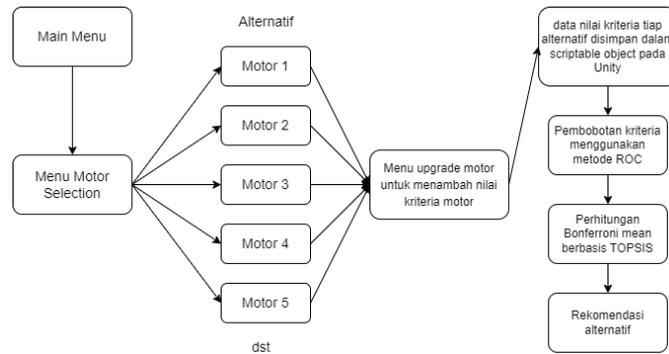
Tampilan *gameplay* (gambar 3.7) memiliki 4 kontrol diantaranya *brake*, *acceleration*, *right*, dan *left*. Dalam *gameplay* ini kriteria tiap motor bekerja sesuai nilainya.



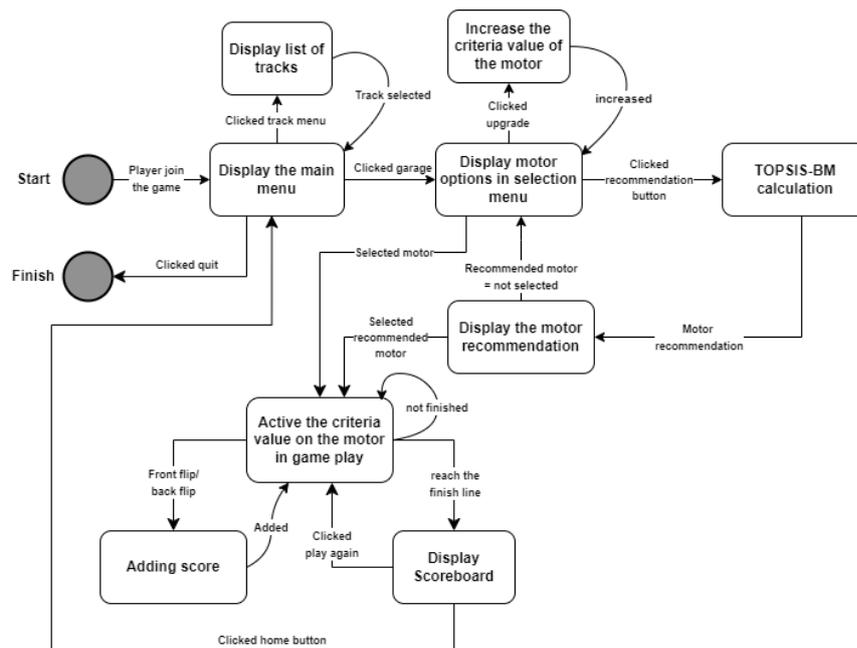
Gambar 3.7 *Gameplay*

3.2 Desain Sistem

Desain sistem merupakan proses perancangan sistem atau teknologi yang akan digunakan dalam penelitian. Desain sistem ini mencakup semua aspek dari pengembangan sistem, mulai dari awal sistem sampai sistem menghasilkan hasil yang diteliti. Desain sistem bisa berupa diagram alur cara kerja sistem. Berikut blok diagram sistem yang diusulkan oleh peneliti mengenai penerapan sistem rekomendasi pada *game* Blar Rising menggunakan metode TOPSIS berbasis Bonferroni mean disajikan pada gambar 3.8.



Gambar 3.8 Blok diagram sistem



Gambar 3.9 Skenario *game* dalam FSM

Gambar 3.9 menunjukkan desain aturan skenario *game* dalam bentuk *Finite State Machine* atau FSM untuk menjelaskan aktivitas sistem *game* (Arif et al., 2022). Pertama ketika *game* di-*play* maka akan menampilkan *main menu*, menu awal dari tampilan dalam *game*. Dalam *main menu* akan menampilkan beberapa menu utama dalam *game*. Selanjutnya, *player* diharuskan untuk memilih medan yang akan digunakan dalam *track menu*. Ini dikarenakan pemilihan medan akan memberikan bobot pada kriteria motor. Selanjutnya, *player* dapat memilih motor dalam menu *garage* yang menampilkan pilihan-pilihan motor dengan spesifikasi yang berbeda-beda. Nilai spesifikasi motor yang mana juga merupakan kriteria dapat di *upgrade* atau nilainya dapat bertambah dikarenakan proses pembelian. Selanjutnya *player* juga dapat menggunakan fitur rekomendasi untuk memperoleh rekomendasi motor. Rekomendasi yang diberikan dalam prosesnya menggunakan perhitungan TOPSIS yang berbasis Bonferroni *Mean*. Dengan bertambahnya nilai kriteria pada suatu motor ataupun bertambahnya pilihan motor yang mana adalah alternatif, maka rekomendasi yang muncul bisa jadi berubah. Maka dari itu dalam *game* ini sistem rekomendasinya secara *realtime* dapat meng-*update* hasil perhitungan keputusan rekomendasi yang muncul. Hasil rekomendasi dapat digunakan oleh *player* ataupun tidak. Pilihan motor yang digunakan *player*, spesifikasinya akan aktif dalam motor yang digunakan bermain dalam *gameplay*.

3.3 Rancangan Perhitungan TOPSIS berbasis Bonferroni *mean*

Dalam perhitungan pemilihan alternatif terbaik menggunakan TOPSIS menggunakan Bonferroni *mean* maupun tidak, diperlukan data alternatif dan kriteria yang dijadikan acuan dalam pemilihan.

3.3.1 Alternatif

Data alternatif merupakan solusi atau opsi yang dapat dipilih dalam memecahkan suatu masalah, dalam hal ini adalah data motor yang tersedia pada *game*. Data alternatif pada *game* ini bersifat *realtime* dimana data bisa bertambah ketika *player* membeli motor dalam *game*.

3.3.2 Kriteria

Data kriteria merupakan faktor – faktor yang dimiliki oleh setiap alternatif yang dipertimbangkan saat pengambilan keputusan. Motor yang digunakan dalam *game* ini memiliki beberapa kriteria yang bisa dijadikan penentu kualitas sebuah motor dalam *game*. Kriteria pada setiap motor pada *game* ini diantaranya seperti pada tabel 3.1.

Tabel 3.1 Tabel kriteria

Simbol	Kriteria	Jenis Kriteria
C1	<i>Speed</i>	<i>Benefit</i>
C2	<i>Agility</i>	<i>Benefit</i>
C3	<i>Handling</i>	<i>Benefit</i>
C4	<i>Fuel Usage</i>	<i>Cost</i>
C5	<i>Motor Weight</i>	<i>Cost</i>

Setelah data kriteria dikumpulkan, maka pembobotan dilakukan menggunakan metode ROC (*Rank Order Centroid*). Perhitungan pembobotan

kriteria menggunakan metode ROC dapat dilakukan untuk menentukan bobot relatif dari kriteria dalam sebuah sistem pengambilan keputusan. Sehingga diperoleh data bobot pada tabel 3.2.

Tabel 3.2 Pembobotan kriteria menggunakan metode ROC

Simbol	Kriteria	Proses Metode ROC	Hasil
C1	<i>Speed</i>	$(1+1/2+1/3+1/4+1/5)/5$	0.45667
C2	<i>Agility</i>	$(1/2+1/3+1/4+1/5)/5$	0.25667
C3	<i>Handling</i>	$(1/3+1/4+1/5)/5$	0.15667
C4	<i>Fuel Usage</i>	$(1/4+1/5)/5$	0.09
C5	<i>Motor Weight</i>	$(1/5)/5$	0.04

Adapun skala penilaian tiap kriteria sebagai berikut.

1. *Speed, Agility, Handling*

Speed, agility, dan handling merupakan kriteria yang bernilai *benefit*.

Semakin tinggi nilai kriteria sebuah motor semakin baik. Berikut skala penilaian untuk kriteria *speed, agility, dan handling* pada tabel 3.3.

Tabel 3.3 Skala penilaian kriteria *speed, agility, dan handling*

Subkriteria (float)		Skala Nilai
≤ 0	Kurang	1
0 – 0.25	Cukup	2
0.25 – 0.5	Cukup Baik	3
0.5 – 0.75	Baik	4
0.75 – 1	Sangat Baik	5

2. *Fuel Usage*

Fuel usage merupakan kriteria yang bernilai *cost*. Semakin tinggi nilai kriterianya semakin kurang performa motornya. Berikut merupakan skala penilaian kriteria *fuel usage* pada tabel 3.4.

Tabel 3.4 Skala penilaian kriteria *fuel usage*

Subkriteria (float)		Skala Nilai
≤ 0	Tidak Boros	1
0 – 0.25	Cukup	2
0.25 – 0.5	Cukup Boros	3
0.5 – 0.75	Boros	4
0.75 – 1	Sangat Boros	5

3. Motor Weight

Motor weight merupakan kriteria yang juga bernilai *cost*. Semakin berat motornya maka semakin kurang performa motornya. Berikut merupakan skala penilaian kriteria *motor weight* pada tabel 3.5.

Tabel 3.5 Skala penilaian kriteria *motor weight*

Subkriteria (kg)		Skala Nilai
≤ 70	Kurang Berat	1
70 – 100	Cukup	2
100 – 130	Cukup Berat	3
130 – 160	Berat	4
≥ 160	Sangat Berat	5

3.3.3 Matriks Keputusan

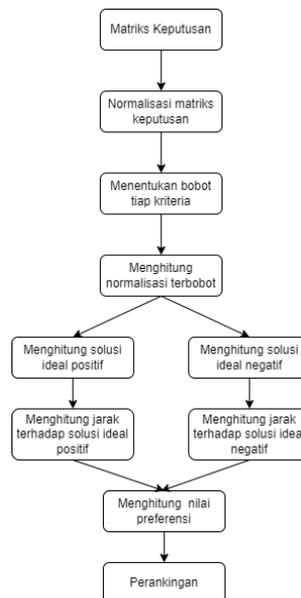
Berdasarkan skala penilaian tingkat kepentingan tiap kriteria, maka dapat ditentukan matriks keputusan untuk setiap alternatif. Dalam matriks keputusan, setiap alternatif dinilai berdasarkan kriteria-kriteria yang telah ditetapkan, kemudian dinilai dengan menggunakan skala penilaian tertentu. Berikut merupakan data kriteria tiap alternatif motor dalam *game* pada tabel 3.6.

Tabel 3.6 Matriks keputusan

	C1	C2	C3	C4	C5
A1	1	3	2	1	2
A2	5	1	3	4	3
A3	4	2	4	2	3
A4	2	5	2	5	1
A5	2	1	3	3	4

3.3.4 Perhitungan TOPSIS

Adapun blok diagram perhitungan TOPSIS asli disajikan seperti pada gambar 3.10.



Gambar 3.10 Blok diagram metode TOPSIS

3.3.4.1 Normalisasi Matriks Keputusan

Setelah didapatkan nilai matriks keputusan kemudian dihitung nilai normalisasi matriks keputusan menggunakan persamaan 2.1 sehingga diperoleh nilai normalisasi pada tabel 3.7.

Tabel 3.7 Normalisasi matriks keputusan metode TOPSIS

	C1	C2	C3	C4	C5
A1	0.141	0.474	0.308	0.134	0.320
A2	0.707	0.158	0.462	0.539	0.480
A3	0.565	0.316	0.617	0.269	0.480
A4	0.282	0.790	0.308	0.674	0.160
A5	0.282	0.158	0.462	0.404	0.640

3.3.4.2 Normalisasi Terbobot

Menghitung rating bobot ternormalisasi dengan mengalikan hasil normalisasi dengan bobot yang telah ditentukan atau dapat dilihat pada rumus 2.2 sehingga diperoleh hasil pada tabel 3.8.

Tabel 3.8 Normalisasi terbobot metode TOPSIS

	C1	C2	C3	C4	C5
A1	0.064	0.121	0.048	0.012	0.012
A2	0.322	0.040	0.072	0.048	0.019
A3	0.258	0.081	0.096	0.024	0.019
A4	0.129	0.020	0.048	0.060	0.006
A5	0.129	0.040	0.072	0.036	0.025

3.3.4.3 Solusi Ideal Positif dan Negatif

Nilai solusi ideal positif dan negatif diperoleh dari nilai maksimum dan minimum dari normalisasi terbobot tiap kriteria. Dimana jika kriteria *benefit* maka solusi ideal positifnya adalah nilai maksimum sedangkan solusi ideal negatif adalah nilai minimum, dan sebaliknya untuk kriteria *cost*. Sehingga diperoleh hasil pada tabel 3.9.

Tabel 3.9 Solusi ideal positif dan negatif metode TOPSIS

	C1	C2	C3	C4	C5
A⁺	0.322	0.121	0.096	0.012	0.006
A⁻	0.129	0.020	0.048	0.060	0.025

3.3.4.4 Jarak

Untuk menghitung jarak tiap alternatif terhadap nilai solusi ideal positif dan negatif menggunakan persamaan 2.5 dan 2.6 sehingga diperoleh hasil jarak pada tabel 3.10.

Tabel 3.10 Jarak

	D_i^+	D_i^-
A1	0.2751	0.0954
A2	0.1685	0.2598
A3	0.1389	0.2070
A4	0.2055	0.1757
A5	0.2557	0.0731

3.3.4.5 Nilai Preferensi

Setelah nilai jarak tiap alternatif terhadap solusi ideal positif dan negatif didapatkan, menghitung nilai preferensi menggunakan persamaan 2.7 sehingga diperoleh hasil nilai preferensi tiap alternatif pada tabel 3.11.

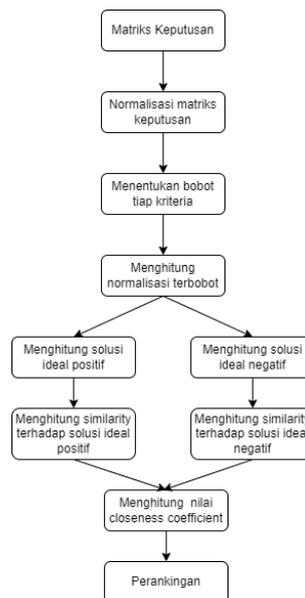
Tabel 3.11 Nilai preferensi

Alternatif	Nilai Preferensi
A1	0.2575
A2	0.6064
A3	0.5984
A4	0.4609
A5	0.2222

Dari tabel 3.11 didapatkan alternatif 2 memperoleh nilai preferensi tertinggi yang mana merupakan hasil keputusan yang didapatkan.

3.3.5 Perhitungan TOPSIS Berbasis Bonferroni Mean

Adapun blok diagram perhitungan TOPSIS menggunakan Bonferroni *mean* disajikan pada gambar 3.11.



Gambar 3.11 Blok diagram metode TOPSIS-BM

3.3.5.1 Normalisasi Matriks Keputusan

Setelah mendapatkan matriks keputusan, pada tahap ini menghitung nilai normalisasi setiap kriteria pada setiap alternatif. Normalisasi dapat dihitung (menggunakan rumus 2.12) dengan menjumlahkan matriks dengan *absolute* nilai minimal dari kriteria dari setiap alternatif kemudian dibagi dengan jumlah nilai maksimal dan minimal dari kriteria dari setiap alternatif. Dengan ini didapatkan data normalisasi pada tabel 3.12.

Tabel 3.12 Normalisasi matriks keputusan metode TOPSIS-BM

	C1	C2	C3	C4	C5
A1	0.333	0.667	0.667	0.333	0.6

A2	1	0.333	0.833	0.833	0.8
A3	0.833	0.5	1	0.5	0.8
A4	0.5	1	0.667	1	0.4
A5	0.5	0.333	0.833	0.667	1

3.3.5.2 Normalisasi Terbobot

Untuk menghitung normalisasi terbobot (menggunakan rumus 2.13) yaitu dengan mengalikan nilai normalisasi matriks keputusan yang didapatkan sebelumnya dengan bobot kriterianya yang telah didapatkan dari perhitungan metode ROC sebelumnya. Sehingga didapatkan data normalisasi terbobot pada tabel 3.13.

Tabel 3.13 Normalisasi terbobot metode TOPSIS-BM

	C1	C2	C3	C4	C5
A1	0.152	0.171	0.104	0.03	0.024
A2	0.457	0.085	0.130	0.075	0.032
A3	0.38	0.128	0.157	0.045	0.032
A4	0.228	0.256	0.104	0.09	0.016
A5	0.228	0.085	0.130	0.06	0.04

3.3.5.3 Solusi Ideal Positif dan Negatif

Nilai solusi ideal positif dan negatif didapatkan dari nilai maksimal dan nilai minimal dari nilai normalisasi terbobot kriteria dari setiap alternatif. Untuk kriteria yang bernilai *benefit*, nilai maksimal merupakan solusi ideal positif dan nilai minimal merupakan solusi ideal negatif. Sedangkan kriteria yang bernilai *cost*, nilai maksimal merupakan solusi ideal negatif dan nilai minimal merupakan solusi ideal positif. Lihat rumus 2.14. Sehingga didapatkan data solusi ideal positif dan negatif pada tabel 3.14.

Tabel 3.14 Solusi ideal positif dan negative

	C1	C2	C3	C4	C5
A⁺	0.457	0.256	0.157	0.03	0.016
A⁻	0.152	0.085	0.104	0.09	0.04

3.3.5.4 Similarity

Pada tahap ini menghitung nilai *similarity* masing-masing alternatif terhadap solusi ideal positif dan terhadap solusi ideal negatif menggunakan rumus Bonferroni *mean* (seperti pada rumus 2.15 dan 2.16). Sehingga didapatkan data *similarity* masing-masing alternatif pada tabel 3.15.

Tabel 3.15 Similarity

	s_i^+	s_i^-
A1	0.9478	0.9948
A2	0.9825	0.9609
A3	0.9701	0.9704
A4	0.9583	0.9843
A5	0.9471	0.9950

3.3.5.5 Closeness Coefficient

Setelah mendapatkan nilai *similarity* maka menghitung nilai *closeness coefficient* untuk masing-masing alternatif dengan rumus 2.17. Sehingga diperoleh nilai *closeness coefficient* sebagai pada tabel 3.16.

Tabel 3.16 Closeness Coefficient

Alternatif	Closeness Coefficient
A1	0.4879
A2	0.5055
A3	0.4999
A4	0.4933
A5	0.4877

Dari tabel 3.16 didapatkan alternatif 2 memperoleh nilai *closeness coefficient* tertinggi yang mana merupakan hasil keputusan yang didapatkan.

3.4 Implementasi Perhitungan Bonferroni Mean

Pada tahap ini perhitungan dapat diimplementasikan seperti pada *pseudocode* 3.1.

Pseudocode 3.1 Main

```
data = [1,3,2,1,2; 5,1,3,4,3; 4,2,4,2,3; 2,5,2,5,1; 2,1,3,3,4];
p=2; m=3; q=3;
we=[0.4566667, 0.2566667, 0.1566667, 0.09, 0.04];
crit=[1 1 1 2 2];
[cc,PISB,NISB]=topsissimBM(data,p,m,q,crit, we)
[Y,I]=sort(cc,'descend');
Order=I
```

Berdasarkan *pseudocode* 3.1 tersebut dituliskan bahwa variable data merupakan matriks keputusan. Kemudian dimasukkan nilai untuk p, m, dan q. Selanjutnya untuk variable crit merupakan jenis kriteria yang digunakan pada masing-masing kriteria, 1 untuk *benefit* dan 2 untuk *cost*. Kemudian untuk memperoleh nilai akhir yaitu *similarities* solusi ideal positif dan negatif serta nilai *closeness coefficient* menggunakan *function* topsissimBM() dengan parameter data yang telah ditentukan sebelumnya. Sehingga diperoleh output sebagai berikut.

Pseudocode 3.2 Output

cc =	0.4879	0.5055	0.4999	0.4933	0.4877
PISB =	0.9478	0.9825	0.9701	0.9583	0.9471

Dari *output* yang diberikan didapatkan alternatif 2 memperoleh peringkat pertama dengan nilai *closeness coefficient* tertinggi dan merupakan hasil rekomendasi yang didapatkan

3.5 Desain Pengujian Sistem

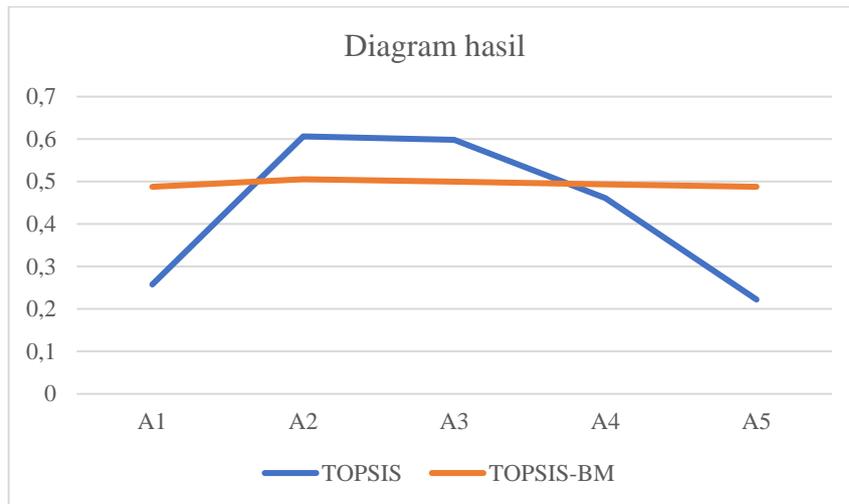
Rencana pengujian dilakukan pada penerapan metode TOPSIS berbasis Bonferroni *mean* dalam game Blar Rising. Uji coba dilakukan pada sejumlah *user* untuk memainkan *game* yang kemudian diambil datanya. Berikut tahapan yang dilakukan dalam pengujian metode pada penelitian ini.

1. Uji coba pemilihan motor pada *game* Blar Rising

Dalam rencana pemilihan motor *game* Blar Rising, disediakan 30 pilihan motor dengan spesifikasi/kriteria yang berbeda-beda dimana tiap kriteria telah memiliki bobot yang berbeda-beda sesuai tingkat kepentingannya. Nilai kriteria dari tiap alternatif motor sendiri dapat dinaikkan nilainya oleh *user* dalam menu *upgrade*. Di dalam *game* terdapat fitur rekomendasi yang dapat memberikan pilihan alternatif dengan menggunakan perhitungan TOPSIS asli maupun TOPSIS berbasis Bonferroni *mean*. Sehingga dengan alternatif motor yang direkomendasikan, pemain dapat menggunakannya untuk bermain.

2. Uji coba perbandingan pilihan motor menggunakan perhitungan TOPSIS asli dan TOPSIS berbasis Bonferroni *mean*

Dalam perhitungan yang telah dilakukan pada 5 alternatif menggunakan metode TOPSIS asli dan TOPSIS yang berbasis Bonferroni *mean* dihasilkan data sebagai berikut.



Gambar 3.12 Diagram hasil perhitungan metode

Dari hasil yang didapatkan pada gambar 3.12, pengujian menggunakan Topsis berbasis Bonferroni *mean* menghasilkan nilai yang similar yang mana nilainya berdekatan, namun sejauh ini tiap alternatif memiliki peringkat yang sama. Selanjut akan dilakukan pada 30 alternatif yang akan ditambahkan pada pilihan motor dalam *game*. Dari jumlah alternatif yang lebih banyak ini akan diketahui apakah akan mempengaruhi peringkat dari hasil akhir perhitungan kedua metode.

3. Uji coba persentase perbandingan metode

Perhitungan persentase menggunakan tingkat kesesuaian pada masing-masing metode. Jika dari masing-masing metode telah didapatkan nilai hasil tiap alternatif maka dilakukan perhitungan dengan menjumlahkan seluruh nilai hasil tiap alternatif dan dibagi dengan banyaknya jumlah alternatif seperti terlihat pada persamaan 3.1 berikut.

$$\text{Metode MADM} = \frac{\text{jumlah nilai hasil}}{\text{jumlah alternatif}} \quad (3.1)$$

Selanjutnya dihitung tingkat kesesuaian untuk membandingkan hasil dari masing-masing metode menggunakan rumus 3.2 berikut.

$$Tk_i = 100 - \frac{X_i}{Data\ MADM(100\%)} \quad (3.2)$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Uji coba dilakukan dalam menu *selection* pada *game* Blar Rising untuk pemilihan motor menggunakan metode TOPSIS yang berbasis Bonferroni *mean*. Pada tahap ini, perhitungan yang telah diperoleh dari bab sebelumnya diimplementasikan dengan membangunnya sesuai dengan kebutuhan yang telah ditetapkan, dan kemudian dilanjutkan dengan melakukan pemrograman sesuai dengan desain yang telah dirancang.

4.1.1 Implementasi Perhitungan Metode

Perhitungan metode TOPSIS asli maupun yang berbasis Bonferroni *mean* dihitung melalui kode program C# yang terintegrasi dengan Unity. Hasil perhitungan tiap tahap disimpan dalam *scriptable object* agar data yang disimpan dapat diakses dan diperbarui oleh beberapa *script*. Tahap awal yang dilakukan yaitu memasukkan nilai kriteria (MK) yang masih berupa *slider* diubah menjadi skala penilaian kemudian diproses sebagai inputan. Kemudian tiap kriteria dihitung bobot menggunakan metode Rank Order Centroid (ROC) seperti *pseudocode 4.1*.

Pseudocode 4.1 Bobot Kriteria

```
FUNCTION CriteriaWeight(TotalCriteria)
  weightList = empty list of floats
  FOR i = 1 TO TotalCriteria
    a = 0
    FOR j = i TO TotalCriteria
      a += (1 / j)
    END FOR
    b = a / TotalCriteria
    weightList.Add(b)
  END FOR
  RETURN weightList
END FUNCTION
```

Pseudocode 4.1 merupakan sebuah fungsi yang melakukan pembobotan kriteria menggunakan metode ROC. Fungsi ini menghasilkan daftar bobot kriteria berdasarkan jumlah total kriteria yang diberikan. Dalam perhitungan bobot, fungsi ini menggunakan nested loop untuk mengakumulasi nilai a, yang merupakan penjumlahan pembalikan dari setiap bilangan bulat dalam range i hingga TotalCriteria. Setelah itu, bobot b dihitung dengan membagi nilai a dengan TotalCriteria, dan ditambahkan ke dalam daftar bobot. Fungsi ini mengembalikan daftar bobot sebagai *output*. Tabel 4.1 merupakan hasil implementasi *pseudocode* 4.1.

Tabel 4.1 *Weight List*

Weight List	0.456667	0.256667	0,156667	0.09	0.04
--------------------	----------	----------	----------	------	------

4.1.1.1 Implementasi Perhitungan TOPSIS

a. Normalisasi Matriks Keputusan

Pseudocode 4.2 Normalisasi matriks keputusan TOPSIS

<pre> FUNCTION SetMK2(index) FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1 Criteria[index].alternativeCriteriaValues[i].MK2 = (Criteria[index].alternativeCriteriaValues[i].MK)^2 END FOR END FUNCTION </pre>
<pre> FUNCTION SetTotal(index) value = 0 FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1 value += Criteria[index].alternativeCriteriaValues[i].MK2 END FOR Criteria[index].TotalMK2 = value END FUNCTION </pre>
<pre> FUNCTION SetNormalisasi(index) FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1 Criteria[index].alternativeCriteriaValues[i].Normalisasi = Criteria[index].alternativeCriteriaValues[i].MK / squareRoot(Criteria[index].TotalMK2) END FOR END FUNCTION </pre>

Pseudocode 4.2 menggambarkan perhitungan menggunakan rumus 2.1 untuk menentukan normalisasi yang dilakukan dengan membagi matriks keputusan dengan akar dari jumlah kuadrat matriks keputusan. Tahap pertama adalah mengkuadratkan nilai MK untuk setiap alternatif dalam fungsi SetMK2. Tahap kedua, dalam fungsi SetTotal, dihitung total nilai MK2 dari seluruh alternatif. Tahap terakhir, dalam fungsi SetNormalisasi, nilai MK setiap alternatif dinormalisasi dengan membaginya dengan akar kuadrat dari total MK2. Hasil normalisasi disimpan dalam atribut Normalisasi. Dengan menjalankan ketiga fungsi ini, matriks keputusan akan dinormalisasi sesuai dengan metode TOPSIS. Hasil dari implementasi *pseudocode* 4.2 disajikan dalam tabel 4.1.

Tabel 4.2 Hasil implementasi perhitungan Normalisasi TOPSIS

	C1	C2	C3	C4	C5
Motor 1	0,0659	0,2142	0,1114	0,0656	0,1212
Motor 2	0,3296	0,0714	0,1671	0,2626	0,1819
...
Motor 29	0,1978	0,1428	0,2229	0,1313	0,1819
Motor 30	0,2637	0,1428	0,2229	0,0656	0,1819

b. Normalisasi Terbobot

Pseudocode 4.3 Normalisasi Terbobot TOPSIS

```

FUNCTION SetNormalisasiTerbobot(index)
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot =
Criteria[index].alternativeCriteriaValues[i].Normalisasi *
Criteria[index].Weight
  END FOR
END FUNCTION

```

Pseudocode 4.3 menggambarkan perhitungan menggunakan rumus 2.2 merupakan sebuah fungsi untuk melakukan normalisasi terbobot pada matriks keputusan menggunakan metode TOPSIS. Fungsi ini melakukan perulangan untuk setiap

alternatif dalam kriteria yang diberikan. Pada setiap iterasi, normalisasi MK alternatif dikalikan dengan bobot kriteria terkait dan hasilnya disimpan dalam atribut `NormalisasiTerbobot` pada struktur data. Hasil dari implementasi *pseudocode* 4.3 disajikan dalam tabel 4.2.

Tabel 4.3 Hasil implementasi perhitungan Normalisasi Terbobot TOPSIS

	C1	C2	C3	C4	C5
Motor 1	0,0301	0,0550	0,0174	0,0059	0,0048
Motor 2	0,1505	0,0183	0,0261	0,0236	0,0072
...
Motor 29	0,0903	0,0367	0,0349	0,0118	0,0072
Motor 30	0,1204	0,0367	0,0349	0,0059	0,0072

c. Solusi Ideal Positif dan Negatif

Pseudocode 4.4 Solusi Ideal Positif dan Negatif TOPSIS

```

FUNCTION SetMaxMinNormalisasiTerbobot(index)
  NormalisasiTerbobotList = empty list of floats
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    NormalisasiTerbobotList.Add(Criteria[index].alternativeCriteriaValues[i]
    ].NormalisasiTerbobot)
  END FOR
  IF Criteria[index].Type = benefit THEN
    Criteria[index].PIS = maximum value in NormalisasiTerbobotList
    Criteria[index].NIS = minimum value in NormalisasiTerbobotList
  END IF
  IF Criteria[index].Type = cost THEN
    Criteria[index].PIS = minimum value in NormalisasiTerbobotList
    Criteria[index].NIS = maximum value in NormalisasiTerbobotList
  END IF
END FUNCTION

```

Berdasarkan rumus 2.3 dan 2.4, *pseudocode* 4.4 menghitung solusi ideal positif (PIS) dan solusi ideal negatif (NIS) dalam metode TOPSIS. Fungsi tersebut membuat sebuah list untuk menyimpan nilai normalisasi terbobot dari alternatif-alternatif dalam kriteria yang diberikan. Kemudian, terdapat dua kondisi IF yang memeriksa tipe kriteria (benefit atau cost). Jika kriteria adalah benefit, PIS dihitung sebagai nilai maksimum dalam list, sedangkan NIS dihitung sebagai nilai

minimum. Jika kriteria adalah cost, PIS dihitung sebagai nilai minimum, dan NIS dihitung sebagai nilai maksimum. Dengan menjalankan *pseudocode* 4.4, PIS dan NIS akan dihitung berdasarkan normalisasi terbobot dari alternatif-alternatif. Hasil dari implementasi *pseudocode* 4.4 disajikan dalam tabel 4.3.

Tabel 4.4 Hasil implementasi perhitungan PIS dan NIS TOPSIS

	C1	C2	C3	C4	C5
PIS	0,15055	0,0916	0,0436	0,0059	0,0024
NIS	0,03011	0,0183	0,0087	0,0295	0,0121

d. Jarak

Pseudocode 4.5 Jarak

```

FUNCTION SetJarakSolusiPositifNegatif(index)
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    Criteria[index].alternativeCriteriaValues[i].JSP =
square(Criteria[index].PIS -
Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot)

    Criteria[index].alternativeCriteriaValues[i].JSN =
square(Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot
- Criteria[index].NIS)
  END FOR
END FUNCTION

FUNCTION SetTotalJS(indexAlt)
  _totalJSP = 0
  _totalJSN = 0
  FOR i = 0 TO CriteriaCount - 1
    _totalJSP += Criteria[i].alternativeCriteriaValues[indexAlt].JSP
    _totalJSN += Criteria[i].alternativeCriteriaValues[indexAlt].JSN
  END FOR

  sqrtJsp = squareRoot(_totalJSP)
  sqrtJsn = squareRoot(_totalJSN)

  AlternativesResult[indexAlternatif].TotalJSP = sqrtJsp
  AlternativesResult[indexAlternatif].TotalJSN = sqrtJsn
END FUNCTION

```

Berdasarkan rumus 2.5 dan 2.6, *pseudocode* 4.5 menghitung jarak tiap alternatif terhadap solusi ideal positif (PIS) dan solusi ideal negatif (NIS) dalam metode TOPSIS. Fungsi *SetJarakSolusiPositifNegatif* melakukan perulangan untuk menghitung jarak alternatif terhadap PIS dan NIS dengan mengkuadratkan selisih

nilai NormalisasiTerbobot. Fungsi SetTotalJS mengakumulasi total jarak JSP dan JSN dari setiap kriteria untuk suatu alternatif, kemudian melakukan pengakaran kuadrat dari total jarak tersebut dan menyimpan hasilnya. Dengan menjalankan kedua fungsi ini, jarak tiap alternatif terhadap PIS dan NIS akan dihitung dan disimpan. Hasil dari implementasi *pseudocode* 4.5 disajikan dalam tabel 4.4.

Tabel 4.5 Hasil implementasi perhitungan Jarak

	D⁺	D⁻
Motor 1	0,1286	0,0450
Motor 2	0,0775	0,1219
...
Motor 29	0,0823	0,0706
Motor 30	0,0634	0,0988

e. Nilai Preferensi

Pseudocode 4.6 Nilai Preferensi

```

FUNCTION SetPreferensiAlternatif(index)
  AlternativesResult[index].Preferensi =
  AlternativesResult[index].TotalJSN /
  (AlternativesResult[index].TotalJSP +
  AlternativesResult[index].TotalJSN)
END FUNCTION

```

Berdasarkan rumus 2.7, *pseudocode* 4.6 menghitung nilai preferensi pada metode TOPSIS. Fungsi SetPreferensiAlternatif menghitung nilai preferensi alternatif dengan membagi total jarak JSN dengan jumlah total jarak JSP dan JSN. Hasil perhitungan nilai preferensi tersebut disimpan dalam atribut Preferensi pada struktur data AlternativesResult. Dengan menjalankan fungsi ini, nilai preferensi

untuk setiap alternatif akan dihitung. Hasil dari implementasi *pseudocode* 4.6 tersebut disajikan dalam tabel 4.5.

Tabel 4.6 Hasil implementasi perhitungan Nilai Preferensi

	Nilai Preferensi
Motor 1	0.2478515
Motor 2	0.6148065
...	...
Motor 29	0.456366
Motor 30	0.6016182

4.1.1.2 Implementasi Perhitungan TOPSIS-BM

a. Normalisasi Matriks Keputusan

Pseudocode 4.7 Normalisasi Matriks Keputusan TOPSIS-BM

```

FUNCTION SetMaxMinMK(index)
  MKList = empty list of floats
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    MKList.Add(Criteria[index].alternativeCriteriaValues[i].MK)
  END FOR
  Criteria[index].MaxMK = maximum value in MKList
  Criteria[index].MinMK = minimum value in MKList
END FUNCTION

FUNCTION SetNormalisasiBM(index)
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    Criteria[index].alternativeCriteriaValues[i].Normalisasi =
(Criteria[index].alternativeCriteriaValues[i].MK +
absoluteValue(Criteria[index].MinMK)) / (Criteria[index].MaxMK +
Criteria[index].MinMK)
  END FOR
END FUNCTION

```

Berdasarkan rumus 2.12, *pseudocode* 4.7 tersebut merupakan tahapan untuk menghitung normalisasi matriks keputusan menggunakan metode TOPSIS berbasis Bonferroni mean. Fungsi SetMaxMinMK menentukan nilai maksimum dan minimum dari matriks keputusan (MK), sedangkan fungsi SetNormalisasiBM menghitung nilai MK alternatif ditambahkan dengan nilai absolut dari MinMK, kemudian dibagi dengan penjumlahan MaxMK dan MinMK. Hasil perhitungan

normalisasi tersebut disimpan dalam atribut Normalisasi. Hasil dari implementasi *pseudocode 4.7* disajikan dalam tabel 4.6.

Tabel 4.7 Hasil implementasi perhitungan Normalisasi TOPSIS-BM

	C1	C2	C3	C4	C5
Motor 1	0,3333	0,6667	0,5	1	1,1667
Motor 2	1	0,3333	0,6667	1,5	1,3333
...
Motor 29	0,6667	0,5	0,8333	1,1667	1,3333
Motor 30	0,8333	0,5	0,8333	1	1,3333

b. Normalisasi Terbobot

Pseudocode 4.8 Normalisasi Terbobot TOPSIS-BM

```

FUNCTION SetNormalisasiTerbobot(index)
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot =
Criteria[index].alternativeCriteriaValues[i].Normalisasi *
Criteria[index].Weight
  END FOR
END FUNCTION

```

Berdasarkan rumus 2.14, *pseudocode 4.8* merupakan tahapan untuk menghitung normalisasi terbobot menggunakan metode TOPSIS-BM. Fungsi SetNormalisasiTerbobot mengalikan nilai normalisasi MK alternatif dengan bobot kriteria terkait. Hasil perhitungan ini disimpan dalam atribut NormalisasiTerbobot. Hasil dari implementasi *pseudocode 4.8* disajikan dalam tabel 4.7 berikut.

Tabel 4.8 Hasil implementasi perhitungan Normalisasi Terbobot TOPSIS-BM

	C1	C2	C3	C4	C5
Motor 1	0,1523	0,1713	0,0783	0,09	0,0467
Motor 2	0,4567	0,0857	0,1044	0,135	0,0533
...
Motor 29	0,3044	0,1283	0,1305	0,105	0,0533
Motor 30	0,3805	0,1283	0,1305	0,09	0,0533

c. Solusi Ideal Positif dan Negatif

Pseudocode 4.9 Solusi Ideal Postif dan Negatif TOPSIS-BM

```

FUNCTION SetMaxMinNormalisasiTerbobot(index)
  NormalisasiTerbobotList = empty list of floats
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
  NormalisasiTerbobotList.Add(Criteria[index].alternativeCriteriaValues[i]
  ].NormalisasiTerbobot)
  END FOR

  IF Criteria[index].Type = benefit THEN
    Criteria[index].PIS = maximum value in NormalisasiTerbobotList
    Criteria[index].NIS = minimum value in NormalisasiTerbobotList
  END IF
  IF Criteria[index].Type = cost THEN
    Criteria[index].PIS = minimum value in NormalisasiTerbobotList
    Criteria[index].NIS = maximum value in NormalisasiTerbobotList
  END IF
END FUNCTION

```

Berdasarkan rumus 2.14, *pseudocode 4.9* merupakan tahapan untuk menghitung solusi ideal positif (PIS) dan solusi ideal negatif (NIS) dalam metode TOPSIS. Fungsi SetMaxMinNormalisasiTerbobot menghasilkan PIS dan NIS berdasarkan nilai maksimum dan minimum dari NormalisasiTerbobotList, tergantung pada tipe kriteria (benefit atau cost). Dengan menjalankan fungsi ini, solusi ideal positif dan solusi ideal negatif akan dihitung dalam metode TOPSIS. Hasil dari implementasi *pseudocode 4.9* disajikan dalam tabel 4.8.

Tabel 4.9 Hasil implementasi perhitungan PIS dan NIS TOPSIS-BM

	C1	C2	C3	C4	C5
PIS	0,4567	0,2567	0,1305	0,09	0,04
NIS	0,1523	0,0855	0,0783	0,15	0,06

d. Similarity

Pseudocode 4.10 Similarity

```

FUNCTION SetSimN_SimP(index, p)
  FOR i = 0 TO length of Criteria[index].alternativeCriteriaValues - 1
    Criteria[index].alternativeCriteriaValues[i].JSP_SimP = 1 -
absoluteValue
(Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot)^p -
(Criteria[index].PIS)^p

    Criteria[index].alternativeCriteriaValues[i].JSN_SimN = 1 -
absoluteValue
(Criteria[index].alternativeCriteriaValues[i].NormalisasiTerbobot)^p -
(Criteria[index].NIS)^p
  END FOR

```

```

FUNCTION tmpPIS(indexAlternatif, q)
  tmpPIS = empty list of floats
  FOR i = 0 TO CriteriaCount - 1
    xnPIS = empty list of floats
    sumPIS = 0
    FOR a = 0 TO CriteriaCount - 1
      xnPIS.Add(Criteria[a].alternativeCriteriaValues[indexAlternatif].SimP)
    END FOR
    xnPIS.RemoveAt(i)
    FOR k = 0 TO length of xnPIS - 1
      sumPIS += (xnPIS[k])^q
    END FOR
    tmpPIS.Add(sumPIS / (CriteriaCount - 1))
    xnPIS.Clear()
  END FOR
  RETURN tmpPIS
END FUNCTION

```

```

FUNCTION tmpNIS(indexAlternatif, q)
  tmpNIS = empty list of floats
  FOR i = 0 TO CriteriaCount - 1
    xnNIS = empty list of floats
    sumNIS = 0
    FOR a = 0 TO CriteriaCount - 1
      xnNIS.Add(Criteria[a].alternativeCriteriaValues[indexAlternatif].SimN)
    END FOR
    xnNIS.RemoveAt(i)
    FOR k = 0 TO length of xnNIS - 1
      sumNIS += (xnNIS[k])^q
    END FOR
    tmpNIS.Add(sumNIS / (CriteriaCount - 1))
    xnNIS.Clear()
  END FOR
  RETURN tmpNIS
END FUNCTION

```

```

FUNCTION SetPISB_NISB(indexAlternatif, m, q)
  tmpPISList = tmpPIS(indexAlternatif, q)
  tmpNISList = tmpNIS(indexAlternatif, q)
  pangkat = 1 / (m + q)
  PIS = empty list of floats
  NIS = empty list of floats
  FOR i = 0 TO CriteriaCount - 1
PIS.Add((Criteria[i].alternativeCriteriaValues[indexAlternatif].SimP)^m
)
NIS.Add((Criteria[i].alternativeCriteriaValues[indexAlternatif].SimN)^m
)
  END FOR
  totalPIS = 0
  totalNIS = 0
  FOR i = 0 TO length of PIS - 1
    totalPIS = totalPIS + PIS[i] * tmpPISList[i]
  END FOR
  FOR i = 0 TO length of NIS - 1
    totalNIS = totalNIS + NIS[i] * tmpNISList[i]
  END FOR
  AlternativesResult[indexAlternatif].TotalPISB = (totalPIS /
CriteriaCount)^pangkat
  AlternativesResult[indexAlternatif].TotalNISB = (totalNIS /
CriteriaCount)^pangkat
END FUNCTION

```

Berdasarkan rumus 2.15 dan 2.16, *pseudocode* 4.10 untuk menghitung nilai *similarity* terhadap solusi ideal positif dan negatif. Dalam baris kode tersebut, dibagi beberapa tahapan yaitu pada fungsi SetSimN_SimP digunakan untuk menghitung nilai ukuran *similarity* positif dan negative yang digunakan dalam rumus Bonferroni *mean*. Kemudian pada fungsi tmpPIS dan tmpNIS digunakan untuk menyimpan nilai *temporary* dari perhitungan Bonferroni *mean* yang kemudian dieksekusi pada fungsi SetPISB_NISB sehingga menghasilkan nilai PISB dan NISB atau nilai *similarity* tiap alternatif terhadap solusi ideal positif dan negatif. Hasil dari implementasi *pseudocode* 4.10 disajikan dalam tabel 4.9.

Tabel 4.10 Hasil implementasi perhitungan *Similarity*

	<i>Similarity+</i>	<i>Similarity-</i>
Motor 1	0,1287	0,0424
Motor 2	0,0765	0,1222
...
Motor 29	0,0827	0,0694
Motor 30	0,0642	0,0970

e. *Closeness Coefficient*

Pseudocode 4.11 Closeness Coefficient

```

FUNCTION SetCC(index)
  AlternativesResult[index].Preferensi =
  AlternativesResult[index].TotalPISB /
  (AlternativesResult[index].TotalPISB +
  AlternativesResult[index].TotalNISB)
END FUNCTION

```

Berdasarkan rumus 2.17, *pseudocode 4.11* digunakan untuk perhitungan *closeness coefficient* pada perhitungan TOPSIS-BM. Dimana fungsi SetCC menghitung nilai PISB dibagi dengan jumlah PISB ditambah NISB. Hasil dari implementasi *pseudocode 4.11* disajikan dalam tabel 4.10 berikut.

Tabel 4.11 Hasil implementasi perhitungan *Closeness Coefficient*

	<i>Closeness Coefficient</i>
Motor 1	0,2478
Motor 2	0,6148
...	...
Motor 29	0,4563
Motor 30	0,6016

Selanjutnya fungsi tiap tahap diproses pada *pseudocode 4.12*.

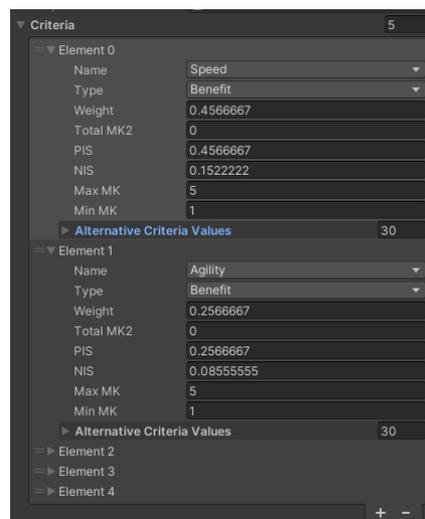
Pseudocode 4.12 Main class

```

FUNCTION Topsis_Method()
  CriteriaWeight(CriteriaData.CriteriaCount)
  CriteriaData.SetCriteriaName()
  for i = 0 to CriteriaData.CriteriaCount do
    CriteriaData.SetWeight(i, weightList[i])
    for i = 0 to MotorData.MotorCount do
      for j = 0 to CriteriaData.CriteriaCount do
        CriteriaData.SetMatrixKep(MotorData.GetMotor(i), i, j)
      for i = 0 to CriteriaData.CriteriaCount do
        if CalculateType == TOPSIS then
          CriteriaData.SetMK2(i)
          CriteriaData.SetTotal(i)
          CriteriaData.SetNormalisasi(i)
          CriteriaData.SetNormalisasiTerbobot(i)
          CriteriaData.SetMaxMinNormalisasiTerbobot(i)
          CriteriaData.SetJarakSolusiPositifNegatif(i)
        else if CalculateType == BonferroniMean then
          CriteriaData.SetMaxMinMK(i)
          CriteriaData.SetNormalisasiBM(i)
          CriteriaData.SetNormalisasiTerbobot(i)
          CriteriaData.SetMaxMinNormalisasiTerbobot(i)
          CriteriaData.SetSimN_SimP(i, p)
        for j = 0 to MotorData.MotorCount do
          if CalculateType == CalculateTypes.TOPSIS then
            CriteriaData.SetTotalJS(j)
            CriteriaData.SetPreferensiAlternatif(j)
          else if CalculateType == BonferroniMean then
            CriteriaData.SetPISB_NISB(j, m, q)
            CriteriaData.SetPreferensiBM(j)
          end FUNCTION

```

Implementasi *pseudocode* 4.12 menghasilkan nilai yang disimpan dalam *scriptable object* seperti pada gambar 4.1.



Element	Name	Type	Weight	Total MK2	PIS	NIS	Max MK	Min MK	Alternative Criteria Values
Element 0	Speed	Benefit	0.4566667	0	0.4566667	0.1522222	5	1	30
Element 1	Agility	Benefit	0.2566667	0	0.2566667	0.0855555	5	1	30
Element 2									
Element 3									
Element 4									

Gambar 4.1 Nilai tiap kriteria pada *scriptable object*

Dalam pengembangannya, penulis membuat *class* khusus *scriptable object* dengan nama *CriteriaData* yang berisikan atribut-atribut untuk menyimpan data kriteria seperti nama kriteria, jenis kriteria, bobot kriteria dan sebagainya serta nilai tiap alternatif dalam sebuah kriteria. Setelah membuat instance dari *scriptable object*, data kriteria dapat diisi dan diakses oleh berbagai script, sehingga dengan mudah mengelola data tersebut. Pada *scriptable object* tersebut juga dibuat untuk menampilkan nilai akhir tiap alternatif sehingga menghasilkan nilai seperti pada gambar 4.2.

Element	Total JSP_PISB	Total JSN_NISB	Preferensi
Element 0	0.9550833	0.993467	0.4901507
Element 1	0.9851545	0.96397	0.5054343
Element 2	0.9757436	0.9712546	0.5011528
Element 3			
Element 4			
Element 5			
Element 6			
Element 7			
Element 8			
Element 9			
Element 10			
Element 11			
Element 12			
Element 13			
Element 14			
Element 15			
Element 16			
Element 17			
Element 18			
Element 19			

Gambar 4.2 Nilai hasil dari tiap alternatif pada *scriptable object*

Kemudian akan diambil nilai terbesar yang akan ditampilkan sebagai rekomendasi motor. Apabila terdapat lebih dari 1 nilai terbesar, maka akan dipilih secara acak dari nilai terbesar itu untuk ditampilkan.

4.2 Pengujian Sistem

Setelah diimplementasikan, tahap selanjutnya pengujian sistem yang dilakukan dengan cara diujikan kepada *user* untuk memainkan *game* yang dibangun dengan tujuan memastikan sistem yang dirancang berjalan sesuai harapan.

4.2.1 Uji Coba *Game*

Untuk menguji coba fitur rekomendasi pilihan motor, *user* menjalankan *game* sesuai arahan. *User* diharuskan memilih medan untuk dimainkan karena akan mempengaruhi hasil pilihan rekomendasi motor. Hal ini dikarenakan setiap medan memiliki kriteria motor yang diperlukan agar *user* dapat bermain dengan maksimal. Maka dari itu tiap medan memiliki bobot yang berbeda yang diberikan kepada kriteria-kriteria motor. Tampilan untuk pemilihan medan bisa dilihat pada Gambar 3.4. Dalam menunya, medan yang tersedia baru hanya ada 2 medan yaitu *Snow Hill* dan *Mountain*.

Setelah memilih medan, selanjutnya *user* menuju menu *garage* untuk memilih motor yang akan digunakan dalam bermain. Dalam menu *garage* (seperti pada Gambar 3.3) terdapat fitur rekomendasi yang ditandai oleh simbol pada gambar 4.3.



Gambar 4.3 Simbol rekomendasi

Setelah menekan *button* rekomendasi akan muncul motor rekomendasi (seperti pada Gambar 3.5) yang bertuliskan “Recommended!”.

RECOMMENDED!

Gambar 4.4 Simbol *recommended*

Dalam fitur rekomendasi yang dibangun, sistem dapat melakukan 2 metode perhitungan yaitu metode TOPSIS asli dan metode TOPSIS yang berbasis Bonferroni *mean*. Tabel 4.11 merupakan hasil dari beberapa uji coba yang dilakukan pada 30 alternatif pilihan motor.

Tabel 4.12 Hasil uji coba

<i>Track</i>	TOPSIS	TOPSIS-BM
<i>Snow Hill</i>	Motor 24	Motor 10
<i>Mountain</i>	Motor 28	Motor 28

1. Uji coba pertama menggunakan medan *Snow Hill* dengan perhitungan TOPSIS asli. Pada percobaan ini menghasilkan motor 24 sebagai pilihan rekomendasi seperti yang terlihat pada gambar 4.5.



Gambar 4.5 Uji coba pertama

- Uji coba kedua menggunakan medan *Snow Hill* dengan perhitungan TOPSIS berbasis Bonferroni *mean*. Pada percobaan ini menghasilkan motor 10 sebagai pilihan rekomendasi seperti yang terlihat pada gambar 4.6.



Gambar 4.6 Uji coba kedua

- Uji coba ketiga menggunakan medan *Mountain* dengan perhitungan TOPSIS asli. Pada percobaan ini menghasilkan motor 28 sebagai pilihan rekomendasi seperti yang terlihat pada gambar 4.7.



Gambar 4.7 Uji coba ketiga

- Uji coba keempat menggunakan medan *Mountain* dengan perhitungan TOPSIS berbasis Bonferroni *mean*. Pada percobaan ini menghasilkan motor 28 sebagai pilihan rekomendasi seperti yang terlihat pada gambar 4.8.



Gambar 4.8 Uji coba keempat

4.2.2 Hasil Uji Coba

Implementasi metode TOPSIS berbasis Bonferroni *mean* pada *game* dibuat dengan perhitungan yang dilakukan langsung dari *C# script* dimana hasil perhitungan tiap tahap disimpan dalam *scriptable object* pada Unity. Adapun input yang diberikan merupakan nilai spesifikasi motor dalam *game* yang telah diubah/di-*upgrade* nilainya oleh player seperti pada tabel 4.12.

Tabel 4.13 Input nilai tiap kriteria

	C1	C2	C3	C4	C5
Motor 1	1	3	2	1	2
Motor 2	5	1	3	4	3
Motor 3	4	2	4	2	3
Motor 4	2	5	2	5	1
Motor 5	2	1	3	3	4
Motor 6	1	2	3	3	2
Motor 7	2	2	3	3	3
Motor 8	1	2	1	4	2
Motor 9	1	1	5	2	2
Motor 10	2	1	5	2	2
Motor 11	2	1	1	3	3
Motor 12	1	4	2	2	1
Motor 13	1	5	2	1	3
Motor 14	4	4	1	1	4
Motor 15	2	1	3	4	4
Motor 16	2	1	3	3	3
Motor 17	1	1	4	2	3

	C1	C2	C3	C4	C5
Motor 18	2	1	4	3	2
Motor 19	2	3	2	1	4
Motor 20	3	3	3	3	3
Motor 21	4	2	3	3	2
Motor 22	5	2	1	2	5
Motor 23	2	2	4	2	4
Motor 24	2	4	4	1	3
Motor 25	2	3	4	4	2
Motor 26	3	3	4	4	4
Motor 27	3	2	4	4	3
Motor 28	5	2	4	1	3
Motor 29	3	2	4	2	3
Motor 30	4	2	4	1	3

Selanjutnya memberikan bobot yang pada kriteria dengan urutan tingkat kepentingan pembobotan metode ROC yang berbeda tiap medannya sebagai berikut.

a. Medan *Snow Hill*

Tabel 4.14 Bobot kriteria ketika medan *snow hill*

Kriteria	Bobot (Metode ROC)
<i>Handling</i>	0.45667
<i>Agility</i>	0.25667
<i>Fuel usage</i>	0.15667
<i>Speed</i>	0.09
<i>Motor weight</i>	0.04

Pada tabel 4.13 merupakan bobot yang diberikan pada kriteria berdasarkan tingkat kepentingan kriteria medan *Snow Hill*. Perhitungan TOPSIS asli dan TOPSIS berbasis Bonferroni *mean* yang dilakukan sistem rekomendasi dengan bobot kriteria yang diberikan medan *Snow Hill* menghasilkan nilai hasil seperti pada tabel 4.14.

Tabel 4.15 Nilai akhir perhitungan (medan *Snow Hill*)

Alternatif	TOPSIS	Peringkat	TOPSIS-BM	Peringkat
Motor 1	0.4099599	22	0.4941072	26
Motor 2	0.3772931	24	0.4948708	22
Motor 3	0.5823063	7	0.5015728	6
Motor 4	0.4698083	16	0.4957124	19
Motor 5	0.3712614	26	0.4948291	23
Motor 6	0.4177915	20	0.4957232	18
Motor 7	0.4222926	19	0.4958093	17
Motor 8	0.1541084	29	0.4885998	29
Motor 9	0.5781321	8	0.5061064	2
Motor 10	0.5835702	6	0.5062254	1
Motor 11	0.1459432	30	0.4883924	30
Motor 12	0.4538359	17	0.4954618	20
Motor 13	0.5234629	11	0.4978255	14
Motor 14	0.4060305	23	0.4935285	27
Motor 15	0.3518382	27	0.494185	25
Motor 16	0.3723267	25	0.4948716	21
Motor 17	0.5015961	13	0.5001351	12
Motor 18	0.4917756	15	0.4997864	13
Motor 19	0.4127356	21	0.4941507	24
Motor 20	0.5	14	0.4972895	15
Motor 21	0.4388924	18	0.4962235	16
Motor 22	0.2696289	28	0.4903796	28
Motor 23	0.5677748	10	0.5011492	10
Motor 24	0.7388994	1	0.5045518	3
Motor 25	0.6002662	4	0.5013933	8
Motor 26	0.6060365	2	0.5014845	7
Motor 27	0.53686	12	0.5002137	11
Motor 28	0.6018359	3	0.5021926	4
Motor 29	0.5760845	9	0.5013584	9
Motor 30	0.5971407	5	0.5019305	5
Total	14.05949		14.93006	

b. Medan *Mountain*Tabel 4.16 Bobot kriteria ketika medan *mountain*

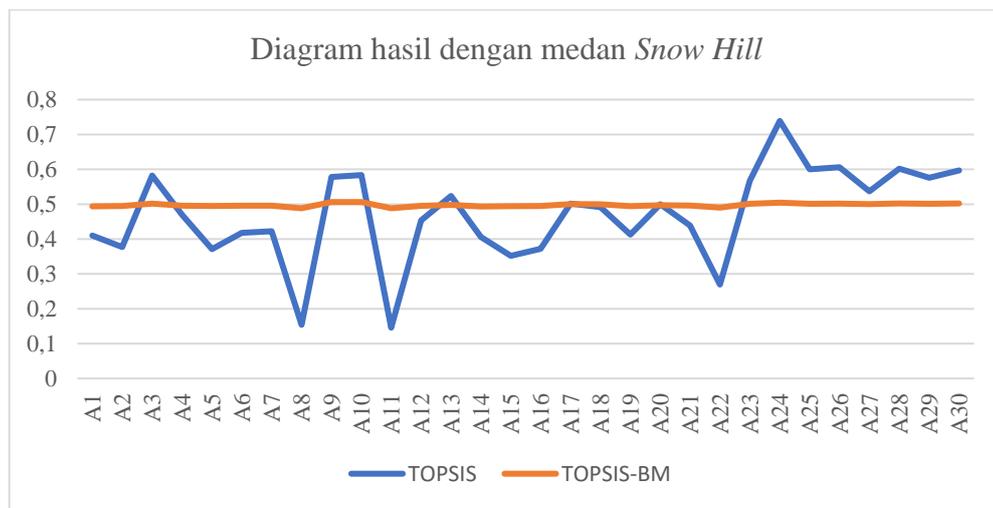
Kriteria	Bobot (Metode ROC)
<i>Speed</i>	0.45667
<i>Agility</i>	0.25667
<i>Handling</i>	0.15667
<i>Motor weight</i>	0.09
<i>Fuel usage</i>	0.04

Pada tabel 4.13 merupakan bobot yang diberikan pada kriteria berdasarkan tingkat kepentingan kriteria medan *Mountain*. Perhitungan TOPSIS asli dan TOPSIS berbasis Bonferroni *mean* yang dilakukan sistem rekomendasi dengan bobot kriteria yang diberikan medan *Mountain* menghasilkan nilai akhir seperti pada tabel 4.16.

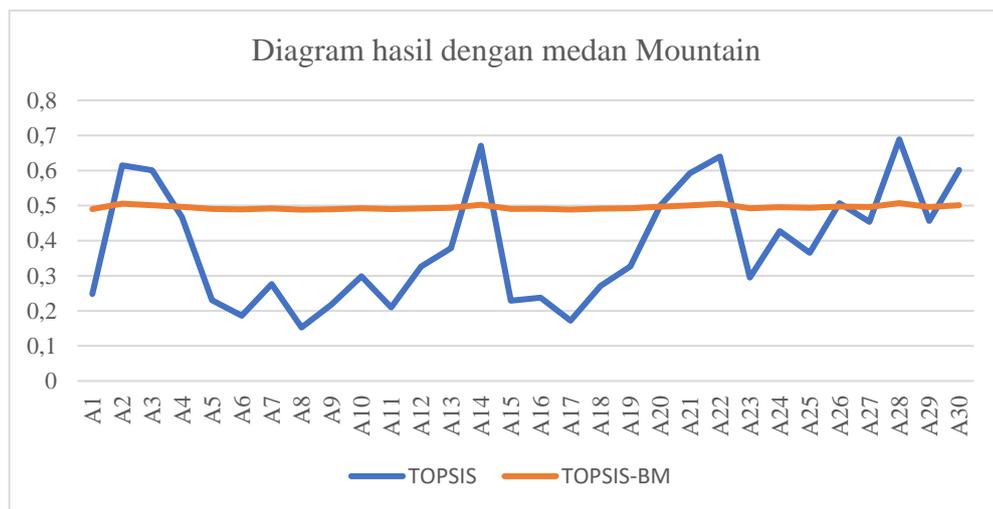
Tabel 4.17 Nilai akhir perhitungan (medan *mountain*)

Alternatif	TOPSIS	Peringkat	TOPSIS-BM	Peringkat
Motor 1	0.2478515	22	0.4901507	25
Motor 2	0.6148065	4	0.5054343	2
Motor 3	0.6008016	6	0.5011528	6
Motor 4	0.4662966	10	0.496562	10
Motor 5	0.2305246	23	0.490654	23
Motor 6	0.1861845	28	0.4892816	28
Motor 7	0.2761933	20	0.4918062	20
Motor 8	0.1525932	30	0.4883804	30
Motor 9	0.218081	26	0.4898083	27
Motor 10	0.2984347	18	0.4924996	16
Motor 11	0.2100189	27	0.4900096	26
Motor 12	0.3261532	17	0.4919349	19
Motor 13	0.3785854	14	0.4937353	15
Motor 14	0.6706883	2	0.5024617	4
Motor 15	0.2288548	25	0.4906115	24
Motor 16	0.2375713	24	0.4908685	22
Motor 17	0.1720334	29	0.4888544	29
Motor 18	0.2709479	21	0.4916794	21
Motor 19	0.3269256	16	0.492461	17
Motor 20	0.5	9	0.4969076	9
Motor 21	0.5927554	7	0.5006425	7
Motor 22	0.6397137	3	0.5051128	3
Motor 23	0.2950381	19	0.4922689	18
Motor 24	0.4275157	13	0.4955089	13
Motor 25	0.3655849	15	0.4938878	14
Motor 26	0.5071457	8	0.4972947	8
Motor 27	0.4539474	12	0.4961957	12
Motor 28	0.6886817	1	0.5071144	1
Motor 29	0.456366	11	0.4962713	11
Motor 30	0.6016182	5	0.5011764	5
Total	11.64191		14.85073	

Untuk melihat perbedaan dari kedua metode dapat disajikan dalam sebuah grafik yang mana akan memperlihatkan perubahan posisi dan bentuk dari datanya. Berikut pada gambar 4.9 dan 4.10 merupakan sajian grafik garis dari hasil perhitungan metode TOPSIS dan TOPSIS-BM.



Gambar 4.9 Diagram hasil dengan medan *Snow Hill*



Gambar 4.10 Diagram hasil dengan medan *Mountain*

Dari diagram 4.9 dan 4.10 didapatkan bahwa metode TOPSIS berbasis Bonferroni *mean* memiliki nilai hasil yang *similar* dibandingkan metode TOPSIS yang menghasilkan nilai tiap alternatifnya berbeda secara signifikan.

4.2.2.1 Perbandingan Perhitungan Metode

Perbandingan metode menggunakan rumus 3.1 yang hasilnya akan digunakan untuk perhitungan tingkat kesesuaian.

a. Menggunakan medan *Snow Hill*

$$TOPSIS = \frac{14.05949}{30} = 0,46864967$$

$$TOPSIS \text{ berbasis } BM = \frac{14.93006}{30} = 0,4976687$$

Kemudian menghitung persentase kesesuaian menggunakan rumus 3.2 seperti berikut

$$\%TOPSIS = 100 - \frac{0.46864967}{100} = 99.9953\%$$

$$\%TOPSIS \text{ berbasis } BM = 100 - \frac{0.46864967}{100} = 99.995\%$$

b. Menggunakan medan *Mountain*

$$TOPSIS = \frac{11.64191}{30} = 0,46864967 = 0,388$$

$$TOPSIS \text{ berbasis } BM = \frac{14.85073}{30} = 0,495$$

Kemudian menghitung persentase kesesuaian menggunakan rumus 3.2 seperti berikut

$$\%TOPSIS = 100 - \frac{0,388}{100} = 99,9998\%$$

$$\%TOPSIS \text{ berbasis } BM = 100 - \frac{0,495}{100} = 99,995\%$$

Dari hasil yang didapatkan diketahui bahwa kedua metode memiliki kemampuan yang sama untuk menyelesaikan permasalahan ini. Penggunaan Bonferroni *mean* sebagai operator agregat dalam perhitungan TOPSIS digunakan sebagai generalisasi. Metode ini menggeneralisasi metode *similarity* based TOPSIS dengan menggantikan pengukuran *similarity* yang digunakan dengan Bonferroni *mean*. Hal ini memungkinkan adanya pendekatan yang lebih fleksibel dalam menghitung tingkat kecocokan antara alternatif dengan solusi ideal positif dan negatif. Dengan menggunakan Bonferroni *mean*, metode ini dapat memberikan penilaian yang lebih konsisten dan akurat dalam mengevaluasi tingkat kecocokan antara alternatif. Bonferroni *mean* mencerminkan hubungan antara kriteria dengan cara yang lebih fleksibel, sehingga dapat menghasilkan penilaian yang lebih akurat dalam memperlihatkan nilai antar alternatif yang berdekatan atau similar.

4.3 Pengujian *Usability* pada Fitur Rekomendasi dalam *Game*

Pengujian *usability* pada fitur rekomendasi untuk mengukur fitur rekomendasi yang diberikan dalam menu *selection* pada *game* dapat digunakan dengan efektif oleh *player* dengan tujuan mengidentifikasi masalah dan kesulitan yang mungkin dialami *player* serta meningkatkan pengalaman *player* agar lebih baik. Pengujian didasarkan pada beberapa komponen sebagai berikut.

- a. *Learnability* (kemudahan), pengujian dapat dilakukan untuk mengukur seberapa cepat *player* dapat memahami dan menguasai fitur rekomendasi.

- b. *Efficiency* (efisiensi), pengujian dapat dilakukan untuk mengukur seberapa cepat dan efisien *player* dapat menyelesaikan tugas-tugas yang terkait dengan fitur rekomendasi.
- c. *Memorability* (kemampuan mengingat), pengujian dapat dilakukan untuk mengukur seberapa mudah *player* dapat mengingat cara menggunakan kembali fitur rekomendasi setelah periode waktu tertentu.
- d. *Error* (kesalahan), pengujian dapat dilakukan untuk mengidentifikasi jenis-jenis kesalahan yang sering terjadi dalam menggunakan fitur rekomendasi dan sejauh mana kesalahan tersebut dapat berdampak pada pengalaman *player*.
- e. *Satisfaction* (kepuasan), pengujian dapat dilakukan untuk mengukur tingkat kepuasan *player* terhadap fitur rekomendasi dan sejauh mana fitur tersebut memenuhi harapan dan kebutuhan mereka.

Untuk melakukan pengujian perlu adanya pengajuan beberapa pertanyaan kepada *player* seperti pada tabel 4.17.

Tabel 4.18 Pertanyaan untuk *player*

No.	Pertanyaan	Usability
1.	Apakah <i>player</i> dapat dengan cepat memahami cara menggunakan fitur rekomendasi motor dalam <i>game</i> ?	<i>Learnability</i>
2.	Apakah <i>player</i> merasa perlu waktu yang cukup singkat untuk mempelajari cara menggunakan fitur rekomendasi?	
3.	Bagaimana perbandingan waktu yang <i>player</i> perlukan untuk memilih motor dengan menggunakan fitur rekomendasi dan tanpa menggunakan fitur tersebut?	<i>Efficiency</i>
4.	Apakah <i>player</i> merasa bahwa fitur rekomendasi membantu mereka menghemat waktu dan usaha dalam memilih motor?	
5.	Setelah jeda waktu tertentu, apakah <i>player</i> masih bisa dengan mudah menggunakan kembali fitur rekomendasi?	<i>Memorability</i>
6.	Apakah <i>player</i> dapat dengan mudah memahami dan menggunakan fitur tersebut tanpa harus mode tutorial?	

7.	Apakah <i>player</i> sering mengalami kesalahan atau kebingungan saat menggunakan fitur rekomendasi?	<i>Error</i>
8.	Apakah <i>player</i> merasa ada bagian dari fitur rekomendasi yang membingungkan?	
9.	Sejauh mana <i>player</i> puas dengan rekomendasi motor yang diberikan oleh fitur ini?	<i>Satisfaction</i>
10.	Apakah <i>player</i> merasa fitur rekomendasi membantu mereka membuat keputusan yang lebih baik dalam memilih motor?	

4.3.1 Analisa *Usability* pada Fitur Rekomendasi dalam *Game*

Setelah dilakukan pengujian dan pengajuan pertanyaan kepada 12 *player* yang membutuhkan jawaban skor 1-4. Adapun skala penilaian *usability* berdasarkan skor yang diperoleh seperti yang terlihat pada tabel 4.18.

Tabel 4.19 Penilaian *Usability* berdasarkan skor

Skor	Nilai	Keterangan
3-4	Sangat Baik	Produk memiliki <i>usability</i> dengan sedikit atau tanpa masalah yang signifikan, sehingga <i>user</i> puas dengan pengalaman yang diberikan
2-3	Cukup Baik	Ada beberapa masalah <i>usability</i> , tetapi mayoritas <i>user</i> dapat menggunakannya dengan relatif lancar
1-2	Buruk	Produk memiliki banyak masalah <i>usability</i> meskipun mungkin ada beberapa aspek yang berfungsi dengan baik
> 1	Sangat Buruk	Produk memiliki banyak masalah <i>usability</i> yang serius, sehingga <i>user</i> mengalami kesulitan yang signifikan

Kemudian setelah pengujian pada sejumlah *player*, dilakukan rekapitulasi hasil jawaban *player* sehingga diperoleh hasil penilaian *usability* pada tabel 4.19.

Tabel 4.20 Pengukuran *Usability*

<i>Usability</i>	Jumlah <i>player</i>	Rata-rata
Keseluruhan pengukuran	12	3.43
<i>Learnability</i>	12	2.95
<i>Efficiency</i>	12	3.87
<i>Memorability</i>	12	3.67

<i>Error</i>	12	3.20
<i>Satisfaction</i>	12	3.45

Dari hasil yang didapatkan menunjukkan bahwa keseluruhan aspek *usability* diterima oleh *player*. Artinya fitur rekomendasi menggunakan perhitungan metode TOPSIS berbasis Bonferroni *mean* memenuhi aspek *learnability* (kemudahan), *efficiency* (efisiensi), *memorability* (kemudahan mengingat), *error* (kesalahan), dan *satisfaction* (kepuasan).

4.4 Integrasi dalam Islam

1. Muamalah Ma'a Allah Subhanahu Wa Ta'ala

Islam sendiri mengajarkan kita untuk senantiasa memilih hal yang baik dan menjauhi yang buruk. Misalnya saja, islam sangat memandang betapa pentingnya memilih pemimpin yang baik, memilih teman yang baik, serta memilih pasangan yang baik. Allah Subhanahu Wa Ta'ala berfirman dalam Al-Qur'an surah Al-Imran ayat 104 sebagai berikut.

وَلْتَكُنْ مِنْكُمْ أُمَّةٌ يَدْعُونَ إِلَى الْخَيْرِ وَيَأْمُرُونَ بِالْمَعْرُوفِ وَيَنْهَوْنَ عَنِ الْمُنْكَرِ ؕ وَأُولَٰئِكَ هُمُ الْمُفْلِحُونَ

“Dan hendaklah ada di antara kamu segolongan umat yang menyeru kepada kebajikan, menyuruh kepada yang ma'ruf dan mencegah dari yang munkar; merekalah orang-orang yang beruntung.” QS. Al-Imran:104

Dalam tafsir al-Manar karya Rasyid Ridho murid dari Muhammad Abduh, dijelaskan bahwa ayat tersebut menyerukan untuk senantiasa memilih yang baik dan mencegah kemunkaran bersifat fardu kifayah. Dalam islam, konsep ihsan atau berusaha mencapai kebaikan terbaik dalam segala hal yang kita lakukan adalah salah satu prinsip dasar. Ini bisa ditemukan dalam hadis berikut:

عَنْ أَبِي نُعْلَى شَدَّادِ بْنِ أَوْسٍ رَضِيَ اللَّهُ عَنْهُ عَنْ رَسُولِ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ : إِنَّ اللَّهَ كَتَبَ الْإِحْسَانَ عَلَى كُلِّ شَيْءٍ ، فَإِذَا

قَتَلْتُمْ فَأَحْسِنُوا الْقِتْلَةَ وَإِذَا ذَبَحْتُمْ فَأَحْسِنُوا الذَّبْحَ وَلْيُجِدَّ أَحَدُكُمْ شَفْرَتَهُ وَلْيُرِحْ ذَبِيحَتَهُ . رواه مسلم

"Sesungguhnya Allah Subhanahu Wa Ta'ala mensyariatkan ihsan (baik dan sempurna) di segala sesuatu. Maka apabila kalian membunuh, maka lakukanlah pembunuhan itu engan baik. Dan apabila kalian menyembelih, maka lakukanlah penyembelihan itu dengan baik. Hendaknya salah seorang di antara kalian mempertajam pisau dan melunakkan hewan yang akan disembelihnya." HR. Muslim

Tafsir oleh Syaikh Muhammad bin Shalih Al-Utsaimin dalam kitabnya Syarah Arba'in An-Nawawi menjelaskan bahwa hadits ini mengandung perintah untuk berbuat kebaikan terhadap diri sendiri serta terhadap setiap makhluk hidup, bahkan hingga pada saat menyembelih hewan dengan cara yang baik dan menyenangkan baginya. Adapun kriteria pisau yang dicontohkan oleh Imam Nawawi dalam menyembelih hewan dianjurkan untuk menajamkan pisau sehingga cepat untuk disembelih dan tidak mengasah pisau tersebut di hadapan hewan. Hal ini menunjukkan bahwa betapa pentingnya di segala sesuatu untuk senantiasa memilih yang baik. Karena ketika kita memilih sesuatu yang buruk maka akan mendatangkan kemungkaran bagi kita.

2. Muamalah Ma'a An-Nas

Islam juga menganjurkan kita untuk memilih barang yang halal dan tidak memilih yang haram yang merupakan bagian dari konsep yang disebut taqwa. Taqwa mengacu pada kesadaran individu terhadap Allah Subhanahu Wa Ta'ala, ketaatan terhadap-Nya, dan usaha untuk hidup dalam ketaatan kepada-Nya. Salah satu cara untuk mencapai taqwa adalah dengan memilih makanan, minuman, pakaian, pekerjaan, dan aktivitas lainnya yang halal. Konsep halal dan haram dalam

Islam berkaitan erat dengan aturan dan pedoman yang ditetapkan dalam Al-Quran dan Hadits, seperti dalam firman Allah Subhanahu Wa Ta'ala pada surah Al-Baqarah ayat 173 yang berbunyi.

إِنَّمَا حَرَّمَ عَلَيْكُمُ الْمَيْتَةَ وَالدَّمَ وَلَحْمَ الْخِنْزِيرِ وَمَا أُهْلَ بِهِ لِغَيْرِ اللَّهِ فَمَنْ اضْطُرَّ غَيْرَ بَاغٍ وَلَا عَادٍ فَلَا إِثْمَ عَلَيْهِ إِنَّ اللَّهَ غَفُورٌ رَحِيمٌ

“Sesungguhnya Dia hanya mengharamkan atasmu bangkai, darah, daging babi, dan (daging) hewan yang disembelih dengan (menyebut nama) selain Allah Subhanahu Wa Ta'ala. Tetapi barangsiapa terpaksa (memakannya), bukan karena menginginkannya dan tidak (pula) melampaui batas, maka tidak ada dosa baginya.” QS. Al-Baqarah:173

Syekh Nawawi Al-Bantani dalam tafsirnya menjelaskan bahwa hadits tersebut menjelaskan bahwa Allah Subhanahu Wa Ta'ala mengharamkan darah dan seluruh bagian dari babi, serta hewan yang disembelih dengan tidak menyebut nama Allah Subhanahu Wa Ta'ala. Dari hadits ini disebutkan beberapa kriteria makanan yang haram untuk dikonsumsi, sehingga kita sebagai umat muslim dianjurkan untuk tidak memilih sesuatu yang diharamkan oleh Allah Subhanahu Wa Ta'ala. Di Indonesia sendiri, indikasi barang halal adalah yang terdapat logo Halal yang bersertifikasi dan diuji oleh Majelis Ulama Indonesia (MUI).

3. Muamalah Ma'a Alam

Penelitian mengambil latar alam digunakan sebagai *environment* dalam *game*, seperti bukit salju, pegunungan, kawah, serta gurun. Alam semesta merupakan sebuah realitas yang dihadapi manusia yang hingga saat ini hanya dapat mengetahui dan mengungkap sebagian kecil dari alam. Allah Subhanahu Wa Ta'ala berfirman pada QS. Yunus ayat 101 sebagai berikut

قُلْ انظُرُوا مَاذَا فِي السَّمَوَاتِ وَالْأَرْضِ يَوْمَ تُغْنِي الْآيَاتُ وَالنُّذُرُ عَنْ قَوْمٍ لَا يُؤْمِنُونَ

“Katakanlah, “Perhatikanlah apa yang ada di langit dan di bumi!” Tidaklah bermanfaat tanda-tanda (kebesaran Allah Subhanahu Wa Ta’ala) dan rasul-rasul yang memberi peringatan bagi orang yang tidak beriman.” QS. Yunus:101

Menurut Ibnu Katsir dalam tafsirnya mengatakan bahwa ayat ini mendorong manusia untuk merenungkan tanda-tanda kuasanya serta semua ciptaannya, baik di langit maupun di bumi. Semua itu mengandung bukti yang nyata akan keagungan Allah Subhanahu Wa Ta’ala Yang Maha Besar bagi mereka yang berpikiran cerdas. Dengan kita menyelidiki alam semesta ini, kita dapat mengenal tanda-tanda kekuasaan Allah dan memahami rahasia-rahasia yang terkandung di dalamnya demi kepentingan manusia sendiri. Tanpa mengkaji dan mempelajari alam semesta, manusia tidak akan mencapai kemajuan dalam kehidupannya.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini menggunakan 2 metode yaitu TOPSIS asli dan TOPSIS berbasis Bonferroni *mean* untuk perhitungan rekomendasi pemilihan motor yang akan digunakan *user* dalam bermain *game*. Bobot yang diberikan menggunakan metode ROC yang mana tingkat kepentingan dari kriteria tergantung dengan *track* atau medan yang dipilih *user*, karena tiap medan memiliki tingkat kepentingan kriteria yang berbeda. Bonferroni *mean* pada dasarnya digunakan untuk menggeneralisasi metode TOPSIS. Ketika digeneralisasi dengan menggunakan Bonferroni *mean*, metode TOPSIS menjadi lebih fleksibel karena operator agregasi yang digunakan lebih luas dan dapat disesuaikan dengan jenis data yang dianalisis.

Implementasi dilakukan pada 2 medan yang berbeda menggunakan 2 metode yang menghasilkan persentase yang sama, sehingga pada permasalahan ini menggunakan metode TOPSIS asli atau TOPSIS yang berbasis Bonferroni *mean* keduanya memiliki performa yang baik. Perbedaan antara hasil perhitungan TOPSIS asli dan TOPSIS berbasis Bonferroni *mean* terletak pada nilai *similarity* dan skor preferensi untuk setiap alternatif. Dalam TOPSIS asli, nilai *similarity* dan skor preferensi untuk setiap alternatif memiliki perbedaan yang signifikan, sementara dalam TOPSIS berbasis Bonferroni *mean*, nilai *similarity* dan skor preferensi menjadi lebih serupa atau similar. Setelah itu dilakukan pengujian *usability* yang menghasilkan nilai rata-rata keseluruhan 3.43, dengan rincian *learnability* (kemudahan) 2.95, *efficiency* (efisiensi) 3.87, *memorability*

(kemudahan mengingat) 3.67, *error* (kesalahan) 3.20, dan *satisfaction* (kepuasan) 3.45. Dari hasil analisa *usability* fitur rekomendasi menunjukkan keseluruhan aspek memiliki tingkat penerimaan *usability* yang cukup baik oleh *player*.

5.2 Saran

Penelitian ini tentu saja belum sempurna sehingga diperlukan adanya beberapa perbaikan untuk mendapatkan hasil penelitian yang lebih maksimal. Berikut merupakan beberapa saran untuk penelitian selanjutnya.

Dari pengujian yang telah dilakukan, dalam penelitian ini, tentunya masih ada kekurangan yang perlu diperbaiki untuk penelitian berikutnya. Adapun saran yang dapat peneliti berikan untuk penelitian selanjutnya sebagai berikut.

1. Data alternatif bisa diperbanyak lagi, dengan variasi alternatif yang berbeda. Karena dalam sebuah *game* perlu adanya variasi dalam bermain agar tidak monoton.
2. Menganalisis lebih dalam pengaruh parameter p dan q dalam operator Bonferroni *mean* terhadap pemeringkatan.
3. Mencoba penelitian operator Bonferroni *mean* pada metode SPK lain.

DAFTAR PUSTAKA

- Abror, M. U. (2017). Sistem Pendukung Keputusan Pemilihan Barang Elektronik Berbasis Web Dengan Metode Topsis. *Program Studi Informatika, Teknik Komputer, Fakultas Ilmu Dian Nuswantoro*, 7(2), 1–7.
- Annubaha, C. (2014). *Implementasi Finite State Machine (Fsm) Pada Game 3D Edukasi Bahasa Arab*. Skripsi. Universitas Maulana Malik Ibrahim Malang.
- Arif, Y. M., Nugroho, S. M. S., & Hariadi, M. (2019). Selection of Tourism Destinations Priority using 6AsTD Framework and TOPSIS. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 346–351. <https://doi.org/10.1109/ISRITI48646.2019.9034671>
- Arif, Y. M., Nurhayati, H., Nugroho, S. M. S., & Hariadi, M. (2022). Destinations Ratings Based Multi-Criteria Recommender System for Indonesian Halal Tourism Game. *International Journal of Intelligent Engineering and Systems*, 15(1), 282–294. <https://doi.org/10.22266/IJIES2022.0228.26>
- Ayu, L. D. (2019). *Game Endless Runner 2D Menggunakan Metode Fuzzy Mamdani*. Skripsi. Universitas Semarang.
- Behler, C. (2020). *GameDev: Character Selection*. Medium. <https://pingpoli.medium.com/gamedev-character-selection-754217426ed2>
- Beliakov, G., James, S., Mordelová, J., Rückschlossová, T., & Yager, R. R. (2010). Generalized Bonferroni mean operators in multi-criteria aggregation. *Fuzzy Sets and Systems*, 161(17), 2227–2242. <https://doi.org/10.1016/j.fss.2010.04.004>
- Bonferroni, C. (1950). Sulle medie multiple di potenze. *Bollettino Dell'Unione Matematica Italiana*, 5(3–4), 267–270.
- Collan, M., & Luukka, P. (2014). Evaluating R&D projects as investments by using an overall ranking from four new fuzzy similarity measure-based topsis variants. *IEEE Transactions on Fuzzy Systems*, 22(3), 505–515. <https://doi.org/10.1109/TFUZZ.2013.2260758>
- Fathurrahman. (2020). *Game Promosi Wisata Kota Malang “Kakang Mbakyu” dengan Menggunakan Decision Tree dan Hierarchy Finite State Machine*. Skripsi. Universitas Maulana Malik Ibrahim Malang.
- Gajah, N. W. (2019). *Implementasi Metode Topsis Dalam Pemilihan Kelompok Tani Terbaik*. Skripsi Universitas Islam Negeri Sumatera Utara Medan. [http://repository.uinsu.ac.id/13010/%0Ahttp://repository.uinsu.ac.id/13010/1/Skripsi Nuraini Wadhiyah Gajah.pdf](http://repository.uinsu.ac.id/13010/%0Ahttp://repository.uinsu.ac.id/13010/1/Skripsi%20Nuraini%20Wadhiyah%20Gajah.pdf)
- Hou, R., Ye, X., Zaki, H. B. O., & Omar, N. A. B. (2023). Marketing Decision

- Support System Based on Data Mining Technology. *Applied Sciences (Switzerland)*, 13(7). <https://doi.org/10.3390/app13074315>
- Jailani, R., & Purwanto, A. (2019). Rancang Bangun Game 2D Dayak Run Bergenre Endles Running Berbasis Android. *Inform : Jurnal Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 4(2). <https://doi.org/10.25139/inform.v4i2.1787>
- Kaur, S., Sehra, S. K., & Sehra, S. S. (2017). A framework for software quality model selection using TOPSIS. *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings*, 736–739. <https://doi.org/10.1109/RTEICT.2016.7807922>
- Kaurie, F., Purwanto, A., & Minarni, M. (2020). Pengembangan Teknologi Game Indonesia untuk Game 2D “HEROES OF INDONESIA” Menggunakan Unity 2D Engine Berbasis Mobile. *Jurnal Ilmu Komputer Dan Bisnis*, 11(2), 2483–2494. <https://doi.org/10.47927/jikb.v11i2.12>
- Luukka, P., & Collan, M. (2016). Bonferroni mean based similarity based TOPSIS. *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016*, 2, 704–709. <https://doi.org/10.1109/FUZZ-IEEE.2016.7737756>
- Michael, Pragantha, J., & Haris, D. A. (2020). Pembuatan Game Arcade-Style Racing “Carstime” dengan Tema Antigravitasi Berbasis Android. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 278–285.
- Napitupulu, D. (2016). Perancangan Program Game PC Hack & Slash Berbasis Multimedia Menggunakan Unity 3D. *Seminar Nasional Technopreneurship Dan Alih Teknologi 2016, October 2016*.
- Novayani, W. (2019). Game Genre for History Education Game based on Pedagogy and Learning Content. *Jurnal Komputer Terapan*, 5(Vol 5 No 2 (2019)), 54–63. <https://doi.org/10.35143/jkt.v5i2.3360>
- Novriantama, V. (2022). *Pemilihan Skenario Game Wisata Kota Batu Menggunakan Metode Topsis Berbobot Dinamis*. Skripsi. Universitas Maulana Malik Ibrahim Malang.
- Pratama, W. (2022). *Perhitungan Score pada Game Pembelajaran Bahasa Arab Menggunakan Metod Fuzzt Type-2*. Skripsi. Universitas Maulana Malik Ibrahim Malang.
- Pratiwi, H. (2020). *Syarat-syarat Sistem Pendukung Keputusan*. STMIK Widya Cipta Dharma.
- Qarthazanny, M. H. (2022). *Pengaruh Perkembangan Game Online di Indonesia*. Kumparan. <https://kumparan.com/mohd-hazim/pengaruh-perkembangan-game-online-di-indonesia-1zRtMH3G3cK/full>
- Rahmansyah, N., & Lusinia, S. A. (2016). *Buku Ajar Sistem Pendukung Keputusan*

(N. Rahmansyah & S. A. Lusia (eds.); 1st ed.). Pustaka Galeri Mandiri.

Roszkowska, E. (2011). Multi-Criteria Decision Making Models By Applying the Topsis Method To Crisp. *Multiple Criteria Decision Making'10-11, Mcdm*, 200–230.

Sari, S. N., & Naim, M. A. (2022). Perancangan game Arcade " The Adventures in Maze ". *MEANS (Media Informasi Analisa Dan Sistem)*, 7(1), 137–143.

Shaheen, T., & Toor, H. G. (2022). Multiple Attribute Decision-Making Based on Bonferroni Mean Operators under Square Root Fuzzy Set Environment Multiple Attribute Decision- Making Based on Bonferroni Mean Operators under Square Root Fuzzy Set Environment. *Journal of Computational and Cognitive Engineering*, 1–22. <https://doi.org/10.47852/bonviewJCCE2202366>

Turban, E., Sharda, R., & Delen, D. (2017). *Decision Support System and Business Intelligence Systems* (E. Turban, R. Sharda, & D. Delen (eds.); 9th ed., Issue 1). Pearson.

Turunen, E. (1999). *Mathematics Behind Fuzzy Logic* (E. Turunen (ed.)). Physica-Verlag.

Vaghasiya, A. (2019). *Key features for the Perfect Mobile Game Development Companies*. LinkedIn. https://www.linkedin.com/pulse/key-features-perfect-mobile-game-development-alpesh-vaghasiya/?trk=public_profile_article_view

Wen, H., Shi, L., Wang, Y., & Ni, T. (2023). Layout Scheme Comparison of the Switching Station of Guoduo Hydroelectric Project Based on AHP-TOPSIS. *E3S Web of Conferences*, 03004, 1–6. <https://doi.org/https://doi.org/10.1051/e3sconf/202338503004> Layout

Yager, R. R. (2009). On generalized Bonferroni mean operators for multi-criteria aggregation. *International Journal of Approximate Reasoning*, 50(8), 1279–1286. <https://doi.org/10.1016/j.ijar.2009.06.004>

Yang, Y., Yang, F., Chen, J., Zeng, Y., & Liu, L. (2022). Pythagorean Fuzzy Bonferroni Mean with Weighted Interaction Operator and Its Application in Fusion of Online Multidimensional Ratings. *International Journal of Computational Intelligence Systems*, 15(1). <https://doi.org/10.1007/s44196-022-00152-y>

Zadeh, L. A. (1970). Similarity Relations And Fuzzy Orderings. *Electronics Research Laboratory*, 2(October), 765–770.