

**KOMBINASI *GENETIC ALGORITHM*-TOPSIS BERBASIS CAPAIAN PEMAIN  
UNTUK MENENTUKAN CUACA DALAM LINGKUNGAN *GAME***

**SKRIPSI**

Oleh :  
**FAUZIL HIDAYATULLAH**  
**NIM. 210605110019**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**KOMBINASI *GENETIC ALGORITHM*-TOPSIS BERBASIS CAPAIAN  
PEMAIN UNTUK MENENTUKAN CUACA DALAM LINGKUNGAN  
*GAME***

**SKRIPSI**

Diajukan Kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :  
**FAUZIL HIDAYATULLAH**  
**NIM. 210605110019**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

## HALAMAN PERSETUJUAN


### KOMBINASI *GENETIC ALGORITHM*-TOPSIS BERBASIS CAPAIAN PEMAIN UNTUK MENENTUKAN CUACA DALAM LINGKUNGAN *GAME*

#### SKRIPSI


Oleh :  
**FAUZIL HIDAYATULLAH**  
NIM. 210605110019

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 28 November 2025

Pembimbing I,


  
Dr. Ir. Fresy Nugroho, M.T., IPM., ASEAN Eng  
NIP. 19710722 201101 1 001

Pembimbing II,

  
Dr. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

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



  
M. Kom  
NIP. 19841010 201903 1 012

## HALAMAN PENGESAHAN

### KOMBINASI *GENETIC ALGORITHM*-TOPSIS BERBASIS CAPAIAN PEMAIN UNTUK MENENTUKAN CUACA DALAM LINGKUNGAN *GAME*

#### SKRIPSI

Oleh :  
**FAUZIL HIDAYATULLAH**  
NIM. 210605110019

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 12 Desember 2025

#### Susunan Dewan Penguji

Ketua Penguji : Hani Nurhayati, M.T  
NIP. 19780625 200801 2 006

Anggota Penguji I : Nur Fitriyah Ayu Tunjung Sari, M.Cs  
NIP. 19911226 202012 2 001


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

Anggota Penguji III : Dr. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

(  )  
(  )  
(  )  
(  )

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



  
Fauzil Hidayatullah, M.Kom  
NIP. 19841010 201903 1 012

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Fauzil Hidayatullah  
NIM : 210605110019  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Kombinasi Genetic Algorithm-Topsis Berbasis  
Capaian Pemain Untuk Menentukan Cuaca Dalam  
Lingkungan Game

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, 12 Desember 2025

Yang membuat pernyataan,



Fauzil Hidayatullah  
NIM. 210605110019

## **MOTTO**

*...sois fort, ne sois pas faible, combats tout...*

## **HALAMAN PERSEMBAHAN**

*Alhamdulillah* rabbil ‘*alamin* puji syukur kepada Allah SWT karena berkat rahmat dan petunjuk-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Rasulullah SAW yang telah membawa kita dari zaman jahiliyah menuju addinul Islam.

Penulis mempersembahkan karya ini kepada diri penulis sendiri, orang tua, para dosen, kerabat, teman, sahabat, serta orang-orang yang telah membantu, mendoakan, serta menyemangati penulis dalam menuntaskan skripsi ini.

## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Segala puji dan rasa syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, semoga kita semua kelak memperoleh syafaat beliau di hari kiamat. Aamiin.

Penulis ucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis disampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Ir. Fresy Nugroho, M.T., IPM., ASEAN Eng, selaku dosen pembimbing 1 serta mentor yang telah memberikan bimbingan, saran, kesabaran dan dukungan yang sangat berarti bagi penulis selama proses penelitian.
5. Dr. M. Imamudin Lc, MA, selaku dosen pembimbing 2, atas arahan dan pencerahan yang sangat membantu dalam menyempurnakan karya ini.



6. Hani Nurhayati, M.T, selaku dosen Penguji I dan Nur Fitriyah Ayu Tunjung Sari, M.Cs, selaku dosen Penguji II yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini.
7. Seluruh staf dan dosen Program Studi Teknik Informatika yang telah memberikan ilmu, dukungan, dan fasilitas kepada penulis selama masa studi.
8. Ayahanda Herman,S.Sos, Ibunda Sri Ekaningsih, dan saudara-saudari penulis yang senantiasa menjadi sumber semangat, memberikan dukungan tanpa henti, serta menyertai dengan doa-doa terbaik, sehingga penulisan skripsi ini dapat terselesaikan dengan baik. Semoga Allah SWT senantiasa menganugerahkan kesehatan dan perlindungan, agar senantiasa dapat mendampingi setiap langkah dan pencapaian penulis ke depannya..
9. Rekan seperjuangan “Kontrakan Pria Tampan”, yang beranggotakan Haikal, Reyhan, Fuaidil, Noviansyah, Gianda, Safril, Musa, Khalif, Rafi, Al, Dika, Amirul, Daffa, dan Zufar. Terima kasih telah menemani hari-hari penulis dan terima kasih sudah mau menemani suka duka serta berbagi tawa dan air mata. Sukses untuk kalian semua.
10. Jovita Yumna Kultum, yang telah dengan sukarela membantu kelancaran proses pengerjaan skripsi ini. Terima kasih atas kebaikan dan bantuannya. Semoga segala kebaikan dibalas dengan keberkahan dan kemudahan dalam setiap urusan.
11. Seluruh keluarga besar Teknik Informatika, khususnya Angkatan 2021 “ASTER”, atas segala ilmu, semangat, serta pengalaman berharga yang

telah dibagikan selama masa perkuliahan. Semoga tali silaturahmi ini senantiasa terjaga, dan kita semua dapat menggapai cita-cita dan impian bersama.

12. Seluruh pihak yang telah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini

Penulis berharap semoga skripsi ini dapat memberikan manfaat di masa yang akan datang. Penulis juga menyadari bahwa karya ini masih memiliki kekurangan. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk masukan dan saran yang membangun untuk pengembangan lebih lanjut.

*Wassalamu'alaikum warahmatullahi wabarakatuh.*

Malang, 12 November 2025

Penulis

## DAFTAR ISI

<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>ABSTRAK .....</b>	<b>xvi</b>
<b>ABSTRACT .....</b>	<b>xvii</b>
<b>المخلص.....</b>	<b>xviii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	6
<b>BAB II STUDI PUSTAKA.....</b>	<b>7</b>
2.1 Studi Pustaka.....	7
2.2 <i>Maze Game</i> .....	11
2.3 DSS (Decision Support System).....	12
2.4 GA-TOPSIS .....	13
2.4.1 Genetic Algorithm (GA).....	14
2.4.2 TOPSIS .....	17
2.5 <i>System Usability Scale</i> .....	20
<b>BAB III DESAIN DAN IMPLEMENTASI .....</b>	<b>21</b>
3.1 Analisis dan Perancangan Game.....	21
3.1.1 Skenario Game.....	21
3.1.2 <i>Storyboard Game</i> .....	22
3.1.3 <i>Finite State Machine</i> .....	24
3.2 Perancangan Perhitungan Metode GA-TOPSIS .....	26
3.2.1 Alternatif.....	26
3.2.2 Kriteria.....	27
3.3 Perhitungan Metode GA-TOPSIS.....	31
3.3.1 Matriks Keputusan.....	33
3.3.2 Normalisasi Matriks Keputusan.....	33
3.3.3 Pembobotan Kriteria Menggunakan Genetic Algorithm .....	34
3.3.4 Matriks Keputusan Ternormalisasi dan Terbobot.....	38
3.3.5 Solusi Ideal Positif & Negatif.....	38
3.3.6 Jarak Solusi Ideal Positif & Negatif .....	39
3.3.7 Nilai Preferensi dan Perankingan .....	39
3.4 Validasi Model GA-TOPSIS .....	40
3.5 Desain Pengujian Usability .....	40
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>43</b>
4.1 Implementasi <i>Game</i> .....	43
4.2 Implementasi GA ( <i>Genetic Algorithm</i> ).....	48

4.3 Implementasi TOPSIS .....	54
4.4 Uji Coba Penyesuaian Cuaca .....	59
4.4.1 Pengujian Skenario Pertama .....	60
4.4.2 Pengujian Skenario Kedua.....	62
4.4.3 Pengujian Skenario Ketiga.....	64
4.4.4 Pengujian Skenario Keempat.....	67
4.4.5 Pengujian Skenario Kelima .....	69
4.4.6 Pengujian Skenario Keenam.....	71
4.4.7 Pengujian Skenario Ketujuh .....	74
4.4.8 Skenario Pengujian Kedelapan .....	76
4.5 Pengujian Validasi 10-Fold.....	78
4.6 Pengujian Usability.....	80
4.7 Integrasi Sains dalam Islam .....	86
4.7.1 <i>Muamalah Ma'a Allah</i> .....	87
4.7.2 <i>Muamalah Ma'a An-nas</i> .....	88
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>90</b>
5.1 Kesimpulan .....	90
5.2 Saran .....	91
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2. 1 Skala Skor SUS .....	20
Gambar 3. 1 Diagram FSM.....	25
Gambar 3. 2 Blok Diagram Metode GA-TOPSIS .....	32
Gambar 4. 1 Halaman Intro.....	44
Gambar 4. 2 Halaman Main Menu.....	44
Gambar 4. 3 <i>Panel Tutorial</i> .....	45
Gambar 4. 4 <i>Scene Gameplay</i> .....	46
Gambar 4. 5 Kotak Kardus Petunjuk Dan NPC Musuh.....	47
Gambar 4. 6 <i>Objek Damage Buff</i> .....	47
Gambar 4. 7 <i>Tampilan Game Result</i> .....	48
Gambar 4. 8 <i>Result Skenario Pertama</i> .....	60
Gambar 4. 9 Hasil Penyesuaian Cuaca Skenario Pertama .....	62
Gambar 4. 10 <i>Result Skenario Kedua</i> .....	62
Gambar 4. 11 Hasil Penyesuaian Cuaca Skenario Kedua.....	64
Gambar 4. 12 Result Skenario Ketiga.....	65
Gambar 4. 13 Hasil Penyesuaian Cuaca Skenario Ketiga.....	66
Gambar 4. 14 Result Skenario Keempat .....	67
Gambar 4. 15 Hasil Penyesuaian Cuaca Skenario Keempat.....	69
Gambar 4. 16 Result Skenario Kelima.....	69
Gambar 4. 17 Hasil Penyesuaian Cuaca Skenario Kelima .....	71
Gambar 4. 18 Result Skenario Keenam .....	72
Gambar 4. 19 Hasil Penyesuaian Cuaca Skenario Keenam .....	73
Gambar 4. 20 Result Skenario Ketujuh .....	74
Gambar 4. 21 Hasil Penyesuaian Cuaca Skenario Ketujuh .....	76
Gambar 4. 22 Result Skenario Kedelapan .....	76
Gambar 4. 23 Hasil Penyesuaian Cuaca Skenario Kedelapan .....	78
Gambar 4. 24 Distribusi Jawaban SUS .....	83

## DAFTAR TABEL

Tabel 3. 1 Storyboard Game .....	22
Tabel 3. 2 Tabel Alternatif .....	26
Tabel 3. 3 Tabel Kriteria .....	27
Tabel 3. 4 Tabel Penilaian Kriteria Waktu Yang Dhabiskan.....	28
Tabel 3. 5 Tabel Penilaian Petunjuk Yang Ditemukan .....	28
Tabel 3. 6 Tabel Penilaian Musuh Yang Dikalahkan .....	28
Tabel 3. 7 Tabel Penilaian HP Yang Tersisa .....	29
Tabel 3. 8 Tabel Penilaian Jumlah Kematian.....	29
Tabel 3. 9 Tabel Penilaian Jumlah Damage Yang Diterima.....	30
Tabel 3. 10 Tabel Penilaian Jumlah Buff Yang Ditemukan .....	30
Tabel 3. 11 Tabel Penilaian Jumlah Debuff Yang Didapatkan.....	31
Tabel 3. 12 Tabel Matriks Keputusan .....	33
Tabel 3. 13 Tabel Matriks Keputusan Ternormalisasi .....	33
Tabel 3. 14 Tabel Populasi Awal .....	34
Tabel 3. 15 Tabel Evaluasi Fitness .....	35
Tabel 3. 16 Tabel Hasil Seleksi Kromosom.....	35
Tabel 3. 17 Tabel Hasil Perhitungan Crossover.....	36
Tabel 3. 18 Tabel Hasil Mutasi .....	37
Tabel 3. 19 Tabel Hasil Optimasi Bobot.....	38
Tabel 3. 20 Tabel Ternormalisasi Dan Terbobot .....	38
Tabel 3. 21 Tabel SIP & SIN .....	39
Tabel 3. 22 Tabel Di+ & Di- .....	39
Tabel 3. 23 Nilai Preferensi Dan Perankingan.....	40
Tabel 3. 24 Tabel Pertanyaan SUS .....	41
Tabel 3. 25 Tabel Interpretasi Skor SUS .....	42
Tabel 4. 1 Optimasi Bobot Skenario Pertama.....	61
Tabel 4. 2 CCI & Perankingan Skenario Pertama.....	61
Tabel 4. 3 Optimasi Bobot Skenario Kedua.....	63
Tabel 4. 4 CCI dan Perankingan Skenario Kedua.....	63
Tabel 4. 5 Optimasi Bobot Skenario Ketiga .....	65
Tabel 4. 6 CCI Dan Perankingan Skenario Ketiga.....	66
Tabel 4. 7 Optimasi Bobot Skenario Keempat.....	68
Tabel 4. 8 CCI Dan Perankingan Skenario Keempat.....	68
Tabel 4. 9 Optimasi Bobot Skenario Kelima .....	70
Tabel 4. 10 CCI dan Perankingan Skenario Kelima .....	70
Tabel 4. 11 Optimasi Bobot Skenario Keenam.....	72
Tabel 4. 12 CCI dan Perankingan Skenario Keenam.....	73
Tabel 4. 13 Optimasi Bobot Skenario Ketujuh .....	75
Tabel 4. 14 CCI dan Perankingan Skenario Ketujuh .....	75
Tabel 4. 15 Optimasi Bobot Skenario Kedelapan.....	77
Tabel 4. 16 CCI dan Perankingan Skenario Kedelapan .....	77
Tabel 4. 17 Data Training dan Testing.....	79
Tabel 4. 18 Hasil Validasi 10 <i>fold</i> .....	79

Tabel 4. 19 Mean Akurasi.....	80
Tabel 4. 20 Kategori Pertanyaan SUS .....	81
Tabel 4. 21 Skor Jawaban Asli SUS .....	82
Tabel 4. 22 Skor per Aspek Usability .....	83
Tabel 4. 23 Skor Hasil Hitung SUS .....	85

## ABSTRAK

Hidayatullah, Fauzil. 2025. **Kombinasi Genetic Algorithm-Topsis Berbasis Capaian Pemain Untuk Menentukan Cuaca Dalam Lingkungan Game**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Ir. Fresy Nugroho, M.T., IPM., ASEAN Eng (II) Dr. M. Imamudin Lc, MA.

**Kata kunci:** *Genetic Algorithm*, TOPSIS, sistem adaptif, cuaca dinamis, *game*, *player performance*.

Perkembangan teknologi dalam industri game menuntut adanya elemen dinamis yang dapat meningkatkan pengalaman bermain. Penelitian ini bertujuan mengimplementasikan kombinasi metode *Genetic Algorithm* (GA) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) untuk menciptakan sistem cuaca adaptif berdasarkan capaian pemain dalam *game* labirin 3D “Salah Serenity”. GA berperan sebagai optimizer untuk menentukan bobot kriteria secara adaptif, sedangkan TOPSIS digunakan sebagai *decision maker* dalam memilih kondisi cuaca. Delapan kriteria capaian pemain yaitu waktu bermain, petunjuk ditemukan, musuh dikalahkan, HP tersisa, jumlah kematian, damage diterima, buff ditemukan, dan debuff didapatkan dijadikan input untuk menentukan satu dari delapan alternatif cuaca: cerah terik, cerah berawan, hujan ringan, hujan sedang, mendung, kabut, hujan deras, dan hujan deras + kabut. Implementasi sistem dilakukan menggunakan *game engine Unity* dengan bahasa pemrograman *C#*. Hasil pengujian fungsional pada delapan skenario menunjukkan bahwa sistem mampu menghasilkan cuaca yang berbeda sesuai dengan variasi capaian pemain. Validasi model menggunakan *10-fold cross-validation* pada 100 data skenario menghasilkan akurasi rata-rata sebesar 88%. Pengujian *usability* dengan *System Usability Scale* (SUS) terhadap 30 responden memperoleh skor rata-rata 73,5 yang tergolong dalam kategori *Good* dan tingkat penerimaan *Acceptable*. Dapat disimpulkan bahwa kombinasi GA-TOPSIS berhasil diterapkan untuk menciptakan sistem cuaca dinamis yang adaptif terhadap capaian pemain, sehingga dapat meningkatkan variasi *gameplay*, menjaga keseimbangan tantangan, dan memperkaya pengalaman bermain secara keseluruhan.



## ABSTRACT

Hidayatullah, Fauzil. 2025. **Kombinasi Genetic Algorithm-Topsis Berbasis Capaian Pemain Untuk Menentukan Cuaca Dalam Lingkungan Game.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Dr. Ir. Fresy Nugroho, M.T., IPM., ASEAN Eng (II) Dr. M. Imamudin Lc, MA.

**Keywords:** Genetic Algorithm, TOPSIS, adaptive system, dynamic weather, game, player performance.

Technological advancements in the gaming industry demand dynamic elements that can enhance the player experience. This study aims to implement a combination of Genetic Algorithm (GA) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to create an adaptive weather system based on player performance in the 3D maze game "Salah Serenity." GA serves as an optimizer to adaptively determine criteria weights, while TOPSIS is used as a decision-maker to select weather conditions. Eight player performance criteria: playtime, clues found, enemies defeated, remaining HP, number of deaths, damage received, buffs found, and debuffs obtained are used as input to determine one of eight weather alternatives: clear sunny, partly cloudy, light rain, moderate rain, cloudy, foggy, heavy rain, and heavy rain + fog. The system was implemented using the Unity game engine with the C# programming language. Functional testing on eight scenarios showed that the system can generate different weather conditions according to variations in player performance. Model validation using 10-fold cross-validation on 100 scenario data yielded an average accuracy of 88%. Usability testing with the System Usability Scale (SUS) on 30 respondents obtained an average score of 73.5, classified as Good and at an Acceptable acceptance level. It can be concluded that the GA-TOPSIS combination was successfully applied to create a dynamic weather system adaptive to player performance, thereby increasing gameplay variety, maintaining challenge balance, and enriching the overall gaming experience.

## الملخص

هداية الله، فوزيل. 2025. الجمع بين الخوارزمية الجينية وتوبسيس القائم على إنجاز اللاعب لتحديد الطقس في بيئة اللعبة. رسالة جامعية قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانغ. المشرف: (1) الدكتور المهندس فريسي نوجروهو، ماجستير في التكنولوجيا، (2) الدكتور محمد إمام الدين، ماجستير

الكلمات المفتاحية: الخوارزمية الجينية، توبسيس، النظام التكيفي، الطقس الديناميكي، اللعبة، أداء اللاعب

يقتضي تطور التكنولوجيا في صناعة الألعاب وجود عناصر ديناميكية لتعزيز تجربة اللعب. يهدف هذا البحث إلى تطبيق طريقة تجمع لإنشاء نظام طقس تكيفي (TOPSIS) وتقنية ترتيب التفضيل من خلال التشابه مع الحل المثالي (GA) بين الخوارزمية الجينية يعتمد على إنجاز اللاعب في اللعبة المتأهات ثلاثية الأبعاد "سلام السكينة". تعمل الخوارزمية الجينية كمحسن لتحديد أوزان المعايير بشكل تكيفي، في حين يُستخدم توبسيس كصانع قرار في اختيار حالة الطقس. يتم استخدام ثمانية معايير لإنجاز اللاعب - وقت اللعب، الأدلة المكتشفة، الأعداء المهزومين، نقاط الصحة المتبقية، عدد الوفيات، الأضرار المستلمة، التعزيزات المكتشفة، والضعف المكتسب - كمدخلات لتحديد واحدة من ثمانية بدائل للطقس: صحو حار، صحو غائم، مطر خفيف، مطر متوسط، غائم أظهرت نتائج الاختبار C# بلغة البرمجة Unity ضباب، مطر غزير، ومطر غزير مع ضباب. تم تنفيذ النظام باستخدام محرك اللعبة الوظيفي في ثمانية سيناريوهات قدرة النظام على إنتاج طقس مختلف يتناسب مع تباين إنجاز اللاعب. أعطت التحقق من النموذج على 100 سيناريو من البيانات دقة متوسطة (10-fold cross-validation) باستخدام التحقق المتقاطع العشري على 30 مستجيباً على (SUS) بلغت 88%. كما حصل اختبار سهولة الاستخدام باستخدام مقياس سهولة استخدام النظام متوسط درجة 73.5، والذي يصنف ضمن فئة جيد ومستوى قبول مقبول. يمكن الاستنتاج أن الدمج بين الخوارزمية الجينية وتوبسيس، قد تم تطبيقه بنجاح لإنشاء نظام طقس ديناميكي يتكيف مع إنجاز اللاعب، مما يعزز تنوع اللعب، ويحافظ على توازن التحدي. ويثري تجربة اللعب بشكل عام.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi dalam industri *game* telah membawa dampak besar terhadap peningkatan kualitas permainan dengan menghadirkan dunia virtual yang semakin realistis dan imersif (Astuti et al., 2023). Salah satu aspek yang penting dalam menciptakan pengalaman realistis dalam sebuah *game* adalah keberadaan sistem cuaca dinamis yang berubah sesuai dengan situasi dalam permainan (Perwira & Hidayat, 2021). Variasi cuaca, seperti hujan deras, kabut, atau cerah, tidak hanya berfungsi sebagai elemen estetika, namun juga dapat memengaruhi visibilitas, kesulitan permainan, dan strategi pemain. Sebagai contoh, cuaca hujan deras atau kabut dapat mengurangi visibilitas, sementara kondisi cerah memungkinkan pemain memiliki jangkauan pandang yang lebih baik untuk eksplorasi maupun pertempuran.

Selain meningkatkan realisme visual, cuaca dinamis juga berkontribusi besar terhadap dinamika permainan (Kumalasari Universitas Jember et al., 2024). Perubahan kondisi cuaca yang dihasilkan berdasarkan aksi dan capaian pemain dapat menghadirkan tantangan yang berbeda di setiap sesi permainan, sehingga menghindari repetisi *gameplay* dan meningkatkan motivasi pemain. Dengan demikian, mekanisme cuaca yang adaptif sangat penting dalam menciptakan pengalaman bermain yang unik, variatif, dan dinamis.

Penelitian ini mengusulkan pengembangan sebuah permainan bernama “Salah Serenity”, yang dirancang menggunakan perangkat lunak Unity. “Salah Serenity” mengisahkan perjalanan seorang pemuda Palestina yang ingin menuju masjid untuk melaksanakan salat, tetapi menghadapi hambatan berupa blokade jalan utama oleh pihak berwenang. Akibatnya, pemain harus mencari jalur alternatif melalui sebuah labirin yang penuh dengan tantangan, seperti menghadapi musuh, menemukan petunjuk, serta menghadapi berbagai kondisi lingkungan yang berubah-ubah secara dinamis.

Dalam *game* “Salah Serenity”, mekanisme cuaca menjadi elemen penting dalam menciptakan suasana serta mengatur tingkat kesulitan permainan. Sebagai contoh, cuaca cerah dapat menciptakan suasana yang lebih santai dengan visibilitas maksimal, sedangkan hujan deras dan badai petir meningkatkan ketegangan serta kesulitan permainan. Oleh karena itu, dibutuhkan sistem adaptif yang dapat menyesuaikan kondisi cuaca berdasarkan performa dan capaian pemain dalam permainan, agar pengalaman bermain menjadi lebih menarik, menantang, dan variatif.

Namun, menentukan kondisi cuaca yang sesuai berdasarkan capaian pemain bukanlah hal yang sederhana. Oleh karena itu, dibutuhkan pendekatan berbasis kecerdasan buatan dan metode pengambilan keputusan, seperti *Genetic Algorithm* (GA) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS). Pemilihan kombinasi metode GA dan TOPSIS didasarkan pada kebutuhan untuk menciptakan sistem cuaca yang adaptif dan realistis dalam *game* “Salah Serenity”. *Genetic Algorithm* pada penelitian ini tidak digunakan untuk

memilih cuaca secara langsung, melainkan untuk menghasilkan bobot kriteria yang lebih adaptif. Bobot tersebut kemudian dipakai dalam perhitungan TOPSIS, sehingga perankingan alternatif cuaca yang dihasilkan lebih sesuai dengan capaian pemain. Dengan pendekatan ini, TOPSIS berfungsi sebagai metode utama pengambilan keputusan, sementara GA berperan sebagai metode pembobotan adaptif. Kombinasi ini memungkinkan *game* untuk menyesuaikan kondisi cuaca berdasarkan performa pemain, sehingga meningkatkan tantangan dan pengalaman bermain yang lebih dinamis.

*Genetic Algorithm* (GA) merupakan salah satu teknik optimasi berbasis algoritma evolusi yang terinspirasi dari mekanisme seleksi alam (Suseno et al., 2022). GA digunakan untuk mencari solusi optimal dengan melakukan seleksi, *crossover*, dan mutasi pada populasi solusi (Suprayogi & Mahmudy, 2015). Sejumlah penelitian telah membuktikan efektivitas GA dalam meningkatkan adaptasi permainan berbasis capaian pemain. Setiawanda et al. (2019) menerapkan GA dalam *game* Ms. Pacman untuk mengontrol pergerakan karakter utama, yang terbukti meningkatkan skor rata-rata pemain dibandingkan dengan metode kontrol konvensional. Lebih lanjut, dalam pengembangan *game* berbasis kecerdasan buatan, Pratiwi & Putra (2023) menerapkan GA dalam *game* "Virus Hunter" untuk mengembangkan karakter musuh dengan evolusi nilai kekuatan secara dinamis, menciptakan tantangan permainan yang lebih menarik.

Dengan mengacu pada pendekatan tersebut, sistem cuaca dalam *game* "Salah Serenity" dirancang agar dapat secara dinamis berubah berdasarkan beberapa faktor pencapaian pemain, seperti waktu bermain, jumlah petunjuk yang

ditemukan, jumlah musuh yang dikalahkan, HP yang tersisa, jumlah kematian, *damage* yang diterima, serta jumlah buff dan debuff yang didapat pemain. Dengan demikian, cuaca tidak hanya berfungsi sebagai elemen estetika, namun juga berperan aktif dalam strategi permainan. Implementasi metode ini diharapkan dapat meningkatkan variasi *gameplay*, menghindari repetisi, serta menghadirkan pengalaman bermain yang lebih menarik, adaptif, dan imersif.

Dalam perspektif Islam, fenomena cuaca merupakan bagian dari tanda-tanda kebesaran Allah SWT, yang telah diatur dalam keseimbangan yang sempurna (Yusron, 2022). Al-Qur'an menggambarkan bagaimana Allah menciptakan dan mengatur perubahan cuaca sebagai bentuk kekuasaan-Nya. Sebagaimana firman Allah dalam QS. Ar-Rum ayat 48:

اللَّهُ الَّذِي يُرْسِلُ الرِّيحَ فَتُثِيرُ سَحَابًا فَيَبْسُطُهُ فِي السَّمَاءِ كَيْفَ يَشَاءُ وَيَجْعَلُهُ كِسْفًا فَيَزِي الْوَدْقَ مِنْ خِلَالِهِ فَإِذَا أَصَابَ بِهِ مَنْ يَشَاءُ مِنْ عِبَادَةٍ إِذَا هُمْ يَسْتَبْشِرُونَ ﴿٤٨﴾

*"Allahlah yang mengirim angin, lalu ia (angin) menggerakkan awan, kemudian Dia (Allah) membentangkannya di langit menurut yang dikehendaki-Nya dan Dia menjadikannya bergumpal-gumpal, lalu engkau melihat hujan keluar dari celah-celahnya. Maka, apabila Dia menurunkan kepada hamba-hamba-Nya yang dikehendaki-Nya, seketika itu pula mereka bergembira." (QS. Ar-Rum: 48)*

Dalam tafsir Kementerian Agama, ayat ini menunjukkan bahwa siklus cuaca bukanlah kejadian acak, melainkan proses yang diatur secara presisi oleh Allah SWT. Angin dikirim untuk menggerakkan awan dan membentuknya menjadi gumpalan yang menurunkan hujan. Distribusi hujan ini bersifat dinamis disebar dan dijatuhkan kepada hamba yang dikehendaki-Nya sehingga menciptakan perubahan suasana yang membawa kegembiraan bagi penerimanya. Tafsir ini menjadi

inspirasi dalam penelitian untuk merancang sistem cuaca adaptif pada *game* "Salah Serenity" demi menghadirkan pengalaman bermain yang realistis dan variatif (Tafsir Kemenag, 2022).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan sebelumnya, penelitian ini berfokus pada bagaimana penerapan GA dan TOPSIS dalam menentukan cuaca yang adaptif dalam lingkungan *game* berdasarkan capaian pemain?

## 1.3 Batasan Masalah

1. Penelitian ini berfokus pada implementasi kombinasi metode GA dan TOPSIS dalam menentukan kondisi cuaca yang adaptif berdasarkan capaian pemain dalam *game* "Salah Serenity".
2. Alternatif kondisi cuaca yang digunakan dalam penelitian ini terbatas pada delapan jenis cuaca, yaitu cerah terik, cerah berawan, hujan ringan, hujan sedang, mendung, kabut, hujan deras, dan hujan deras + kabut.
3. Capaian pemain diukur berdasarkan delapan kriteria, yaitu waktu bermain, petunjuk yang ditemukan, jumlah musuh yang dikalahkan, HP tersisa, jumlah kematian, jumlah *damage* yang diterima, *buff* yang ditemukan, dan *debuff* yang didapatkan.

## 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan kombinasi metode GA dan TOPSIS untuk menentukan cuaca dalam *game* berdasarkan capaian pemain

sehingga menghasilkan sistem cuaca yang adaptif dan dinamis untuk meningkatkan realisme dan tantangan dalam permainan.

### **1.5 Manfaat Penelitian**

1. Mengetahui hasil implementasi kombinasi metode GA dan TOPSIS dalam menentukan sistem cuaca yang adaptif dan dinamis di lingkungan *game*.
2. Memperluas penerapan GA maupun TOPSIS dalam lingkungan *game*, yang sebelumnya lebih banyak digunakan dalam bidang lain seperti bisnis dan manajemen keputusan.



## BAB II

### STUDI PUSTAKA

#### 2.1 Studi Pustaka

Beberapa penelitian yang telah dilakukan menggunakan metode GA maupun TOPSIS salah satunya oleh Setiawanda et al. (2019) yang berjudul “Penerapan Algoritma Genetika dan Algoritma *Ghost Framework* pada game Ms. Pacman”, penelitian ini menerapkan GA untuk mengontrol pergerakan *Pacman* dalam game Ms. Pacman dan membandingkannya dengan *Ghost Framework*, yang menggunakan algoritma *Depth First Search* (DFS) dan *Breadth First Search* (BFS) untuk mengontrol pergerakan musuh (*Ghost*). GA bekerja dengan inisialisasi, populasi, evaluasi fitness berdasarkan skor permainan, seleksi individu terbaik, *crossover*, dan mutasi untuk menghasilkan strategi optimal. Hasil penelitian menunjukkan bahwa GA mampu menghasilkan skor rata-rata 8,330, lebih tinggi dibandingkan pemain amatir dan beberapa *controller* berbasis aturan lainnya. Penelitian ini membuktikan bahwa GA efektif dalam mengembangkan strategi permainan yang lebih adaptif, dengan potensi peningkatan melalui kombinasi dengan metode AI lainnya seperti *Reinforcement Learning*.

Penelitian yang dibuat oleh Wijaya et al. (2021) Penelitian ini menggunakan Algoritma Genetika (GA) untuk menghasilkan desain level dinamis pada game edukasi kebakaran hutan dengan metode *Procedural Content Generation* (PCG). GA diterapkan menggunakan model *Steady State Population*, di mana setiap generasi baru menghasilkan dua kromosom baru melalui proses *crossover* dari

generasi sebelumnya yang telah diseleksi berdasarkan nilai fitness. Inisialisasi populasi dilakukan dengan *Random Initialization*, sementara seleksi orang tua menggunakan metode *Roulette Wheel Selection*, *crossover* diterapkan dengan *Uniform Crossover*, dan mutasi menggunakan *Random Resetting* untuk menghindari premature convergence. Evaluasi fitness mempertimbangkan empat faktor utama: kemungkinan level dapat diselesaikan, keseimbangan distribusi pohon, sebaran titik api, dan rasio blok pohon vertikal-horizontal. Setelah iterasi sebanyak 800 generasi, desain level yang dihasilkan memenuhi kriteria optimal dan diterapkan dalam *game* menggunakan *game engine Unity*. Hasilnya menunjukkan bahwa algoritma ini mampu menghasilkan level yang dinamis dan adaptif, meningkatkan pengalaman bermain dengan variasi level yang terus berubah.

Penelitian selanjutnya yang dibuat oleh Pratiwi & Putra (2023) berisikan tentang penerapan GA pada *game Virus Hunter* berbasis Android untuk mengembangkan evolusi nilai kekuatan karakter musuh dalam *game*. GA digunakan untuk menyesuaikan atribut *health* (darah), *movement speed* (kecepatan bergerak), *damage hit* (besar serangan), *jump strength* (kekuatan melompat), *skill* (kemampuan), dan *armor* (pelindung) dari NPC musuh, sehingga tingkat kesulitan permainan meningkat seiring berjalannya waktu. Metode yang digunakan meliputi pembentukan populasi awal secara acak, evaluasi fitness untuk menentukan efektivitas individu, seleksi orang tua menggunakan metode *Tournament Selection*, *crossover* untuk menghasilkan variasi baru, mutasi guna menghindari *premature convergence*, dan evaluasi *fittest* hingga kondisi optimal tercapai. Pengembangan *game* dilakukan menggunakan metode *Multimedia*

*Development Life Cycle* (MDLC) yang mencakup tahap konsep, desain, pengumpulan bahan, pembuatan, pengujian, dan distribusi. Implementasi algoritma dilakukan dalam *engine Construct 2*, dan pengujian menggunakan metode *black box* dan *white box*, yang membuktikan bahwa sistem berfungsi sesuai dengan tujuan. Hasilnya menunjukkan bahwa dengan evolusi nilai musuh yang dinamis, *game* menjadi lebih menantang dan adaptif terhadap pemain, meningkatkan *replayability* dan kompleksitas *gameplay*.

Penelitian selanjutnya yang dilakukan oleh Mu'alimin & Latipah (2021) penelitian ini dilakukan untuk mengembangkan Sistem Pendukung Keputusan (SPK) untuk pemilihan *game Android* yang sesuai bagi anak usia dini menggunakan metode "*Technique for Order Preference by Similarity to Ideal Solution*" (TOPSIS). TOPSIS bekerja dengan menormalisasi matriks keputusan, menentukan bobot, mengidentifikasi solusi ideal positif dan negatif, serta menghitung jarak antara alternatif dengan solusi ideal untuk menghasilkan peringkat *game* terbaik. Kriteria yang digunakan dalam sistem ini meliputi rating *game*, tanggal rilis, jumlah unduhan, harga item termurah, dan ukuran file *game*. Aplikasi ini dirancang untuk membantu orang tua dalam memilih *game* yang sesuai dengan kebutuhan anak mereka dengan hasil akhir berupa perbandingan alternatif *game* dari yang paling direkomendasikan hingga yang paling tidak sesuai. Hasil pengujian menunjukkan bahwa sistem ini memiliki akurasi 75% dalam menentukan *game* terbaik, dengan potensi peningkatan melalui integrasi dengan metode lain dalam penelitian mendatang.

Penelitian selanjutnya yang dilakukan oleh (Prakasa et al., 2025) yang berjudul "*Penerapan Metode GA-TOPSIS untuk Sistem Seleksi Karakter Game dengan Pembobotan Dinamis*". Penelitian tersebut berhasil mengintegrasikan GA dengan TOPSIS untuk mengatasi kelemahan utama TOPSIS, yaitu penentuan bobot kriteria yang seringkali bersifat subjektif. Dalam metode hybrid mereka, GA berperan untuk mencari dan menentukan set bobot kriteria secara otomatis dan optimal melalui proses evolusi. Nilai *Closeness Coefficient* (CCi) dari TOPSIS digunakan sebagai fungsi fitness untuk mengevaluasi kualitas setiap set bobot yang dihasilkan oleh GA. Hasilnya menunjukkan bahwa pendekatan GA-TOPSIS mampu menghasilkan sistem rekomendasi karakter yang adaptif sesuai dengan perubahan kondisi lingkungan. Penelitian ini memberikan landasan metodologis yang kuat bagi penelitian yang akan dilakukan, yaitu menerapkan pendekatan serupa untuk menentukan cuaca dinamis dalam *game* berdasarkan capaian pemain."

Berbeda dengan penelitian-penelitian sebelumnya yang menerapkan GA atau TOPSIS secara terpisah, penelitian ini mengadopsi dan mengembangkan pendekatan *hybrid* GA-TOPSIS seperti yang telah divalidasi oleh (Prakasa et al., 2025). Perbedaannya terletak pada objek penelitian, di mana metode ini akan diterapkan untuk mengoptimalkan pembobotan kriteria capaian pemain guna menentukan kondisi cuaca yang dinamis dalam *game* "Salah Serenity", bukan untuk seleksi karakter.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti & Tahun	Metode Yang Digunakan	Objek Penelitian	Hasil Penelitian
1	Setiawanda et al. (2019)	Genetic Algorithm	Game Ms. Pacman	GA meningkatkan performa karakter dalam menghindari musuh dibandingkan AI konvensional
2	Wijaya et al. (2021)	Genetic Algorithm	Game Edukasi Kebakaran Hutan	GA diterapkan untuk membangun level permainan secara prosedural dengan model Steady State Population.
3	Pratiwi & Putra (2023)	Genetic Algorithm	Game "Virus Hunter"	GA diterapkan pada NPC musuh untuk mengatur evolusi kekuatan karakter, meningkatkan tingkat kesulitan permainan secara dinamis.
4	Mu'alimin & Latipah (2021)	TOPSIS	Sistem Rekomendasi Game Android	TOPSIS dapat membantu pemain memilih game berdasarkan kriteria.
5	Prakasa et al. (2025)	Hybrid GA-TOPSIS	Sistem Seleksi Karakter Game	GA berhasil mengoptimalkan pembobotan kriteria TOPSIS secara dinamis berdasarkan skenario (pagi, siang, malam).

## 2.2 Maze Game

*Maze game* adalah jenis permainan yang mengajak pemain untuk menjelajahi labirin dan menemukan jalan keluar atau mencapai tujuan tertentu (Yuniasih et al., 2023). Tantangan dalam *game* ini sering kali datang dari rintangan seperti jalur buntu, jebakan, atau musuh yang berkeliaran. *Maze game* sering kali dirancang dengan berbagai tingkat kesulitan, yang membuat pemain terus tertantang dan termotivasi untuk menyelesaikan setiap level.

Seiring perkembangan teknologi, *maze game* mengalami peningkatan kompleksitas, dengan memanfaatkan kecerdasan buatan (AI) serta desain *procedural generation* yang memungkinkan labirin berubah secara dinamis. Hal ini tidak hanya meningkatkan tantangan permainan, tetapi juga memberikan pengalaman bermain yang selalu baru, menghindarkan pemain dari rasa bosan karena repetisi *gameplay* yang monoton (Nakhoda et al., 2020).

Penulis memilih labirin sebagai latar dalam *game* karena labirin mencerminkan perjalanan penuh tantangan yang menuntut pemain untuk berpikir strategis, menemukan jalan keluar, dan menghadapi berbagai rintangan di sepanjang perjalanan. Selain itu, penggunaan labirin memungkinkan perubahan kondisi cuaca secara dinamis berdasarkan capaian pemain, membuat permainan terasa lebih hidup dan adaptif. Dengan desain ini, “Salah Serenity” tidak hanya menjadi *game* yang menarik untuk dimainkan, tetapi juga membawa pesan mendalam tentang perjuangan dan keteguhan hati dalam menjalankan kewajiban di tengah tantangan yang ada. Desain ini tidak hanya memberikan tantangan yang lebih kompleks, tetapi juga memperkaya aspek *storytelling*, menciptakan pengalaman bermain yang lebih mendalam dan bermakna.

### 2.3 DSS (Decision Support System)

*Decision Support System* (DSS) adalah sistem informasi berbasis komputer yang membantu dalam pengambilan keputusan, terutama dalam situasi yang melibatkan masalah kompleks atau ketidakpastian (Pristanti, 2015). DSS bertujuan untuk mendukung proses keputusan dengan menyediakan informasi yang relevan dan analisis yang dibutuhkan untuk membuat keputusan yang lebih baik. Sistem ini dirancang untuk memungkinkan pengambil keputusan untuk menganalisis data, memilih alternatif, serta merencanakan dan memprediksi hasil yang mungkin terjadi (Wahono & Ali, 2021).

Dalam konteks *game*, DSS dapat diterapkan untuk meningkatkan pengalaman bermain dengan memberikan rekomendasi atau penyesuaian dinamis terhadap kondisi permainan, seperti tingkat kesulitan, skenario cuaca, atau

keputusan berbasis capaian pemain. Misalnya, dalam *game* “Salah Serenity”, DSS dapat digunakan untuk memonitor capaian pemain dan memberikan rekomendasi atau penyesuaian kondisi cuaca berdasarkan performa mereka. Dengan menggunakan metode seperti GA atau TOPSIS, DSS dapat menganalisis data capaian pemain dan menentukan solusi terbaik yang dapat meningkatkan tantangan dalam permainan, memberikan pengalaman bermain yang lebih menarik dan mendalam.

Keuntungan utama dari DSS adalah kemampuannya untuk mengurangi ketidakpastian dan meningkatkan efisiensi pengambilan keputusan. Sistem ini memungkinkan pengambil keputusan untuk mengevaluasi berbagai alternatif dengan lebih cepat dan lebih akurat, serta memprediksi konsekuensi dari setiap pilihan (Ramadhan, 2023). Oleh karena itu, DSS sangat berguna dalam pengelolaan *game* dinamis dan sistem interaktif yang membutuhkan adaptasi terus-menerus terhadap perilaku pemain atau faktor lingkungan.

## 2.4 GA-TOPSIS

Metode GA–TOPSIS merupakan pendekatan hibrida yang menggabungkan kekuatan *Genetic Algorithm* (GA) sebagai algoritma evolusioner untuk optimasi bobot dengan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) sebagai metode pengambilan keputusan multikriteria. GA digunakan untuk menghasilkan bobot kriteria secara otomatis dan adaptif, sehingga kelemahan TOPSIS konvensional yang sangat bergantung pada pembobotan subjektif dapat diminimalkan.

Integrasi ini memungkinkan sistem untuk menilai alternatif keputusan berdasarkan kondisi yang dinamis, misalnya capaian pemain dalam game, sehingga hasil keputusan lebih objektif. Pada prosesnya, GA menghasilkan kombinasi bobot terbaik melalui mekanisme seleksi, crossover, dan mutasi, sedangkan TOPSIS menghitung nilai kedekatan (*closeness coefficient*) dari setiap alternatif terhadap solusi ideal positif dan negatif. Hasil akhir berupa perankingan alternatif yang lebih akurat karena bobot yang digunakan telah dioptimasi oleh GA.

#### 2.4.1 Genetic Algorithm (GA)

*Genetic Algorithm* (GA) adalah algoritma evolusioner yang meniru mekanisme seleksi alam (seleksi, *crossover*, dan mutasi) untuk mencari solusi optimal pada ruang solusi yang luas. Dalam penelitian ini, GA tidak digunakan untuk secara langsung memilih alternatif cuaca, melainkan mengoptimasi bobot kriteria yang akan dipakai oleh TOPSIS. Strategi ini mengatasi kelemahan TOPSIS konvensional yang bergantung pada bobot subjektif, karena bobot diperoleh secara otomatis dari proses optimasi berbasis performa (*closeness coefficient*) TOPSIS itu sendiri.

##### 1. Representasi Kromosom

Bagian ini direpresentasikan sebagai vektor bobot kriteria. Setiap kromosom merepresentasikan satu set bobot lengkap untuk semua kriteria dengan panjang yang sesuai dengan jumlah kriteria yang ada, dimana setiap gen menyimpan nilai bobot kriteria tertentu. Guna untuk memenuhi *constraint* bahwa total bobot seluruh kriteria harus bernilai 1 agar konsistensi matematis proses optimasi terjaga.

$$\vec{w} = [w_1, w_2, w_3, \dots, w_n] \quad (2.1)$$



Di mana :

- $w$  direpresentasikan sebagai kromosom vektor bobot.

Setelah bobot ditemukan, selanjutnya yaitu normalisasi nilai. Normalisasi ini guna memastikan total bobot bernilai 1 sebagai syarat normalisasi bobot. *Constraint* akan menjaga validitas solusi agar tetap konsisten dalam konteks TOPSIS dengan rumus berikut.

$$\sum_{i=1}^n w_i = 1 \quad (2.2)$$

Di mana :

- $w_i$  adalah bobot kriteria ke- $i$ .

## 2. Fungsi Fitness

Kualitas sebuah kromosom  $w$  dievaluasi menggunakan indikator kedekatan TOPSIS (*Closeness Coefficient*,  $CC_i$ ) terhadap solusi ideal. Pendekatan ini memastikan bahwa solusi yang dihasilkan tidak hanya optimal tetapi juga valid secara logika.

$$Fitness = CC_i = \frac{D^-}{D^+ + D^-} \quad (2.3)$$

Di mana :

- $D^-$  adalah jarak *Euclidean* alternatif ke- $i$  terhadap solusi ideal negatif,
- $D^+$  adalah jarak *Euclidean* alternatif ke- $i$  terhadap solusi ideal positif.

Rumus *Euclidean Distance* antara alternatif dengan solusi ideal positif dan negatif digunakan untuk mendapat nilai  $D^+$  dan  $D^-$ . Jarak ini dihitung berdasar nilai-nilai yang telah dinormalisasi dan dibobotkan.

$$D^+ = \sqrt{\sum (v_{ij} - v_j^+)^2}, \quad D^- = \sqrt{\sum (v_{ij} - v_j^-)^2} \quad (2.4)$$

Di mana :

- $v_{ij}$  adalah elemen matriks keputusan terbobot untuk alternatif ke- $i$  pada kriteria ke- $j$ ,
- $v_j^+, v_j^-$  adalah nilai ideal untuk kriteria ke- $j$ .

### 3. Operator Evolusioner

Proses evolusi dijalankan melalui tiga operator utama yang dieksekusi berurutan. *Seleksi* bertugas memilih kromosom unggul sebagai calon induk sesuai tingkat *fitness*-nya. Setelah itu, *crossover* menggabungkan sifat genetik dari dua induk untuk membentuk keturunan yang membawa karakteristik baru. Terakhir, mutasi menambahkan variasi genetik acak yang terkontrol guna menjaga keragaman populasi serta mencegah terjebak pada solusi lokal.

Peluang kromosom ke- $i$  terpilih sebagai induk diberikan oleh:

$$p_i = \frac{\text{Fitness}_i}{\sum \text{Fitness}} \quad (2.5)$$

Di mana :

- $p_i$  adalah peluang kromosom ke- $i$  terpilih sebagai induk pada tahap seleksi,
- $\text{Fitness}_i$  adalah nilai fitness kromosom ke- $i$ ,
- $\sum \text{Fitness}$  adalah jumlah nilai fitness seluruh kromosom dalam populasi.

Usai seleksi, tahap *crossover* dilakukan untuk memadukan dua induk dan menghasilkan individu baru. *Crossover* yang digunakan adalah aritmatika, yakni pencampuran dua vektor bobot induk dengan koefisien tertentu, sehingga keturunannya merupakan kombinasi linier dari keduanya:

$$\vec{W}_{\text{child}} = a\vec{w}_{\text{parent 1}} + (1 - a)\vec{w}_{\text{parent 2}} \quad (2.6)$$

Di mana :

- $w$  adalah vektor bobot kriteria,

- $\vec{w}_{\text{parent 1}}, \vec{w}_{\text{parent 2}}$  adalah vektor bobot dari dua induk terpilih,
- $a$  adalah koefisien pencampur bernilai real.

Setelah crossover, diterapkan mutasi *uniform* untuk memperkenalkan variasi pada populasi dan menghindari konvergensi prematur. Mutasi dilakukan dengan menambahkan sebuah nilai acak kecil dari rentang seragam (uniform) pada gen yang terpilih.

$$w'_i = w_i + r \quad (2.7)$$

Di mana :

- $w'_i$  adalah gen bobot kriteria ke- $i$  setelah mutasi,
- $w_i$  adalah nilai gen bobot sebelum mutasi,
- $r$  adalah sebuah bilangan acak kecil yang dibangkitkan dari rentang seragam, misalnya antara -0.1 dan +0.1.

## 2.4.2 TOPSIS

TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) adalah metode pengambilan keputusan multi-kriteria yang digunakan untuk memilih alternatif terbaik berdasarkan kedekatannya dengan solusi ideal (Gunawan, 2024). TOPSIS pertama kali diperkenalkan oleh Yoon dan Hwang, serta banyak digunakan dalam berbagai aplikasi pengambilan keputusan yang melibatkan banyak kriteria (Muningsih & Kiswati, 2022). Dalam penelitian ini, TOPSIS digunakan untuk menentukan kondisi cuaca yang paling sesuai berdasarkan capaian pemain, sehingga pengalaman bermain menjadi lebih adaptif dan menantang.

TOPSIS bekerja dengan konsep bahwa solusi terbaik adalah solusi yang memiliki jarak terdekat dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif (Falah et al., 2024). Metode ini melibatkan beberapa langkah utama sebagai berikut:

### 1. Normalisasi Matriks

Normalisasi dilakukan untuk mengubah data mentah ke skala yang sebanding sehingga tiap kriteria yang bisa memiliki satuan dan rentang berbeda dapat dibandingkan secara adil. Proses ini juga mengurangi bias akibat perbedaan skala pengukuran antar kriteria. Nilai ternormalisasi dihitung sebagai:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}} \quad (2.8)$$

Di mana :

- $r_{ij}$  adalah nilai normalisasi dari elemen matriks Keputusan,
- $x_{ij}$  adalah nilai asli dalam matriks keputusan.

### 2. Pembobotan

Tahap ini menerapkan bobot optimal hasil GA pada matriks yang telah dinormalisasi. Pembobotan memberikan penekanan relatif sesuai tingkat kepentingan setiap kriteria, sehingga matriks keputusan mencerminkan prioritas aktual. Nilai terbobot diperoleh dengan:

$$v_{ij} = r_{ij} \times w_j \quad (2.9)$$

Di mana :

- $v_{ij}$  adalah elemen matriks keputusan berbobot,
- $w_j$  adalah bobot dari setiap kriteria.

### 3. Solusi Ideal

Solusi ideal digunakan sebagai tolok ukur evaluasi alternatif. *Positive ideal solution* mewakili capaian terbaik, sedangkan *negative ideal solution* menggambarkan kondisi terburuk. Untuk kriteria bertipe benefit dan cost, titik ideal didefinisikan sebagai:

Benefit:

$$v_j^+ = \max_i(v_{ij}), v_j^- = \min_i(v_{ij}) \quad (2.10)$$

Di mana :

- $v_j^+$  Solusi ideal positif untuk kriteria benefit adalah nilai maksimum dari semua alternatif pada kriteria j,
- $v_j^-$  Solusi ideal negatif untuk kriteria benefit adalah nilai minimum dari semua alternatif pada kriteria j.

Cost:

$$v_j^+ = \min_i(v_{ij}), v_j^- = \max_i(v_{ij}) \quad (2.11)$$

Di mana :

- $v_j^+$  Solusi ideal positif untuk kriteria cost adalah nilai minimum dari semua alternatif pada kriteria j,
- $v_j^-$  Solusi ideal negatif untuk kriteria cost adalah nilai maksimum dari semua alternatif pada kriteria j.

### 4. Perankingan

Urutan preferensi alternatif ditetapkan dengan menilai kedekatan masing-masing alternatif terhadap solusi ideal positif dan negatif. Kedekatan relatif diringkas dalam *Closeness Coefficient*:

$$CC_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (2.12)$$

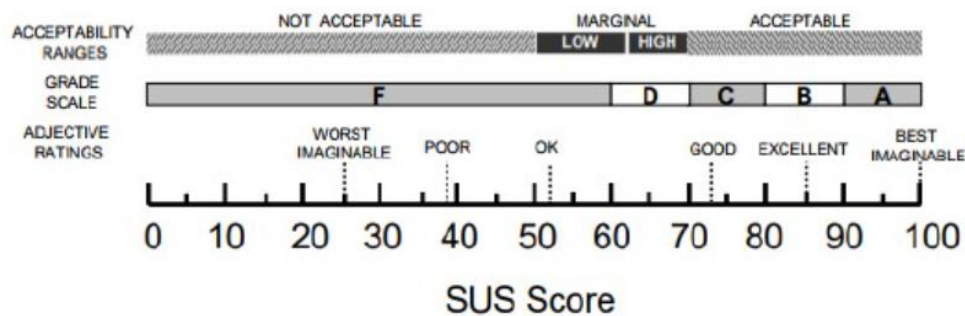
Di mana :

- $CC_i$  adalah Closeness Coefficient untuk alternatif ke-i,
- $D_i^+$  adalah jarak alternatif ke-i terhadap solusi ideal positif (nilai terbaik),
- $D_i^-$  adalah jarak alternatif ke-i terhadap solusi ideal negatif (nilai terburuk).

## 2.5 System Usability Scale

*System Usability Scale* (SUS merupakan suatu metode yang bersifat *quick and dirty*. Metode ini umum digunakan untuk menilai tingkat kegunaan suatu aplikasi atau sistem dengan menggunakan kuesioner. SUS sangat mudah dan populer karena sangat mudah dijalankan dan sangat praktis digunakan saat melakukan perhitungan. SUS mulai diperkenalkan oleh Jhon Broke pada tahun 1986 di Digital (Tri et al., 2023).

$$Rata - rataSUS = \frac{\text{Jumlah Skor SUS Semua Responden}}{\text{Jumlah Responden}} \quad (2.13)$$



Gambar 2. 1 Skala Skor SUS

Setiap rentang nilai dikelompokkan ke dalam kategori huruf (F sampai A) serta tingkat penerimaan (Not Acceptable, Marginal, dan Acceptable). Tampilan pada gambar 2.1 mempermudah dalam memahami kualitas kegunaan sistem baik secara kuantitatif maupun kualitatif (Ridwan et al., 2024).

## **BAB III**

### **DESAIN DAN IMPLEMENTASI**

#### **3.1 Analisis dan Perancangan Game**

Bagian ini menjelaskan semua yang berkaitan dengan game yang dikembangkan meliputi :

##### **3.1.1 Skenario Game**

"Salah Serenity" adalah *game* 3D ber genre labirin yang memadukan elemen petualangan dan strategi. Pemain berperan sebagai seorang pemuda yang berusaha menuju masjid, namun jalan utama terhalang oleh polisi, memaksanya untuk mencari rute alternatif melalui sebuah labirin yang penuh tantangan. Dalam perjalanan ini, pemain harus mencari petunjuk, menemukan jalan keluar, serta menghadapi musuh yang menghalangi. Penilaian terhadap kinerja pemain tidak hanya didasarkan pada waktu yang dihabiskan, tetapi juga pada petunjuk yang ditemukan, musuh yang dikalahkan, dan sisa HP saat keluar dari labirin. Skor ini akan mempengaruhi kondisi cuaca di level berikutnya, memberikan pengalaman bermain yang lebih dinamis dan sesuai dengan capaian pemain.



Perancangan *game* ini mencakup beberapa komponen penting, termasuk halaman menu utama yang menyediakan opsi seperti "*Play Games*," "*Option*," dan "*Exit*," untuk memberikan pengalaman bermain yang interaktif dan menyeluruh. Hasil dari perhitungan GA-TOPSIS akan diterapkan untuk menyesuaikan cuaca yang muncul setelah pemain menekan tombol "*Play Game*" dan menyelesaikan

level permainan. Skor yang diperoleh pemain pada level tersebut akan menjadi dasar untuk menentukan kondisi cuaca pada level selanjutnya.

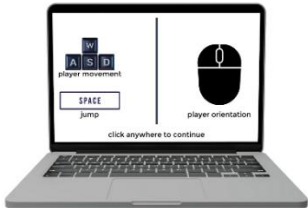
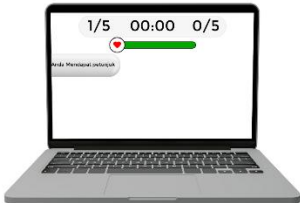


### 3.1.2 Storyboard Game


*Storyboard game* akan menggambarkan desain antarmuka pengguna (UI) untuk *game* "Salah Serenity". Berikut ini adalah *storyboard* yang menunjukkan tampilan dan alur interaksi dalam *game* "Salah Serenity".

Tabel 3. 1 Storyboard Game

No.	Gambar	Keterangan
1.		Sebelum masuk halaman utama, pemain akan disuguhkan dengan <i>introduction</i> halaman pembuka yang akan menuju halaman menu utama.
2.		Pada halaman menu utama, terdapat 3 pilihan button diantaranya <i>play game</i> untuk memulai permainan, <i>option</i> untuk beralih ke menu pengaturan, dan <i>exit</i> untuk keluar dari permainan.



3.		<p>Saat pemain menekan button <i>play game</i>, pemain akan diberikan instruksi awal yang berisi panduan sebelum memulai permainan dan menekan button bebas untuk beralih ke permainan.</p>
4.		<p>Setelah masuk ke dalam permainan, pemain akan disuguhkan dengan <i>gameplay</i> yang akan dimainkan oleh pemain serta tampilan UI seperti <i>health bar</i>, waktu, dan <i>objective</i>.</p>
5.		<p>Bagian petunjuk akan muncul setiap kali pemain mendapatkan petunjuk dalam <i>gameplay</i>.</p>
6.		<p>Halaman pause akan muncul apabila pemain menekan tombol "Esc" pada <i>keyboard</i>.</p>

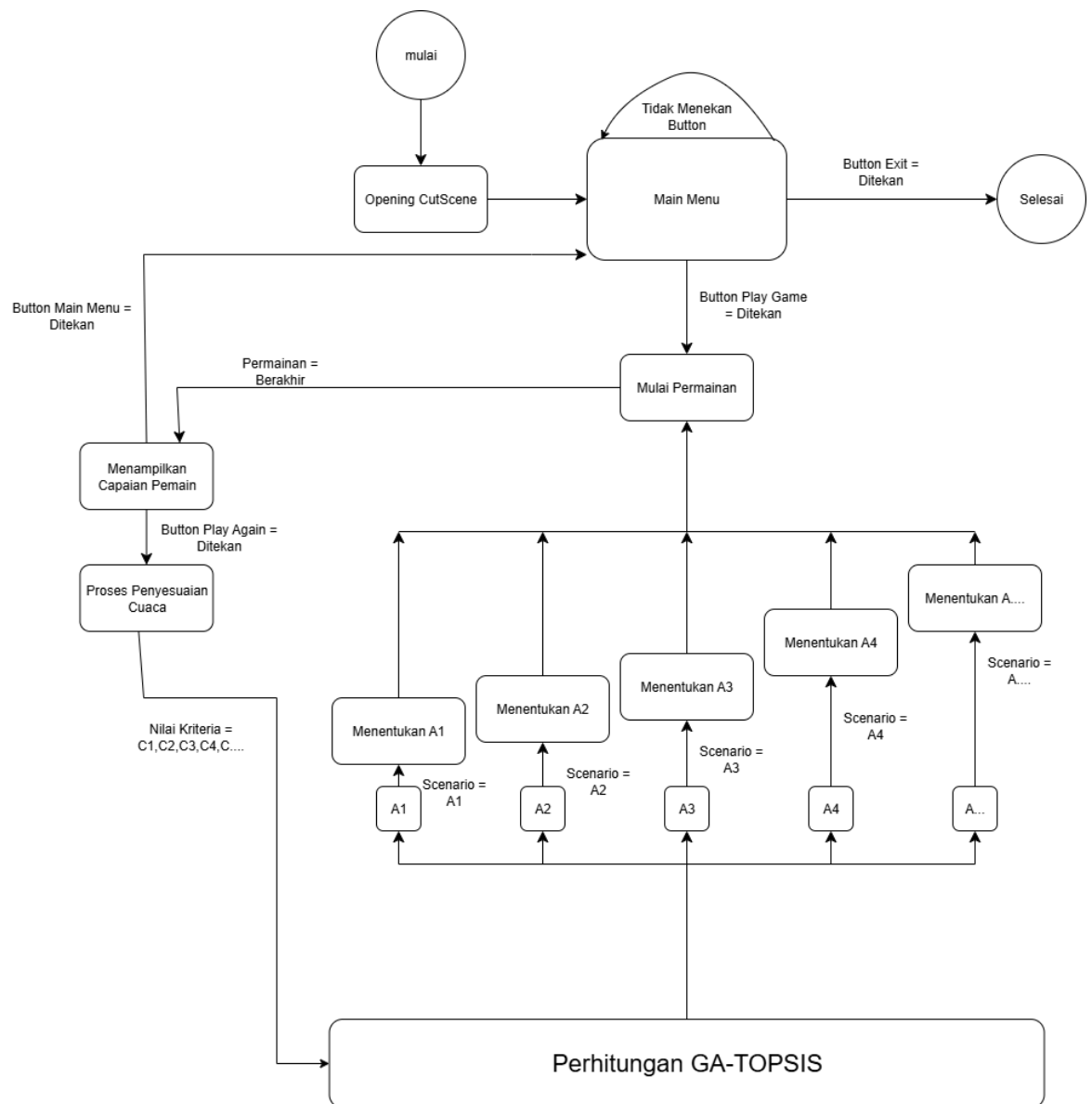
7.		<p>Setelah pemain menyelesaikan permainan maka akan di suguhkan dengan tampilan “game result” yang berisi capaian pemain. Ketika pemain menekan button <i>play again</i>, maka pemain akan memulai permainan lagi dengan penyesuaian cuaca yang sesuai dengan capaian pemain sebelumnya.</p>
----	---	--

### 3.1.3 Finite State Machine

*Finite state machine* (FSM) adalah suatu model abstrak yang menggambarkan berbagai kondisi dalam sebuah sistem serta menentukan aturan kapan suatu kondisi (*state*) harus berubah ke kondisi lainnya (Hormansyah et al., 2018). Keadaan ini mencerminkan kondisi internal yang unik dari sistem. Metode *finite state machine* sendiri dipilih karena merupakan metode yang cukup sederhana namun sangat baik dalam memodelkan perilaku agen. Metode *finite state machine* menggunakan *state* (kondisi), *event* (kejadian) dan *action* (aksi) yang saling berkaitan (Firdaus et al., 2024).

Dalam *state machine* sistem menempati satu *state* (keadaan). Sistem akan beralih atau bertransisi menuju ke *state* lain jika mendapatkan masukan event tertentu (Yulsilviana & Ekawati, 2019). Sistem akan tetap melakukan aksi yang sama pada suatu *state* sampai sistem menerima *event* tertentu baik yang berasal dari perangkat luar atau komponen dari sistem itu sendiri. Tujuan *finite state machine* (FSM) sendiri akan mempermudah mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi dalam permainan (Wisjaya et al., 2019).

Gambar 3.1 merupakan *finite state machine* (FSM) yang digunakan sebagai penerapan rekomendasi penyesuaian cuaca pada game “Salah Serenity” menggunakan kombinasi metode GA dan TOPSIS.



Gambar 3. 1 Diagram FSM

Dalam game "Salah Serenity", FSM digunakan untuk mengatur perubahan cuaca yang dipengaruhi oleh pencapaian pemain, serta untuk berpindah antar level permainan. Ini membuat pengalaman bermain menjadi lebih dinamis dan menarik.

### 3.2 Perancangan Perhitungan Metode GA-TOPSIS

#### 3.2.1 Alternatif

Kriteria merupakan elemen atau aspek yang digunakan untuk mengevaluasi dan membandingkan berbagai pilihan yang ada. Dalam metode TOPSIS, setiap kriteria diberikan bobot yang menunjukkan seberapa penting kriteria tersebut dibandingkan dengan yang lainnya. Kriteria ini menjadi landasan untuk menilai sejauh mana setiap pilihan dapat memenuhi kebutuhan atau tujuan yang diinginkan. Dalam *game* "Salah Serenity", kriteria yang digunakan adalah sebagai berikut:

Tabel 3. 2 Tabel Alternatif

No	Alternatif	Keterangan
1.	A1	Cerah Terik
2.	A2	Cerah Berawan
3.	A3	Hujan Ringan
4.	A4	Hujan Sedang
5.	A5	Mendung
6.	A6	Kabut
7.	A7	Hujan Deras
8.	A8	Hujan Deras + Kabut

### 3.2.2 Kriteria

Kriteria adalah faktor yang digunakan untuk menilai dan membandingkan berbagai pilihan. Dalam metode TOPSIS, setiap kriteria memiliki bobot yang menunjukkan seberapa pentingnya dibandingkan dengan yang lain. Kriteria ini membantu menentukan seberapa baik setiap alternatif memenuhi kebutuhan atau tujuan tertentu. Berikut adalah kriteria dalam permainan "Salah Serenity":

Tabel 3. 3 Tabel Kriteria

No.	Kriteria	Jenis Kriteria	Keterangan
1.	C1	Cost	Waktu bermain
2.	C2	Benefit	Petunjuk yang ditemukan
3.	C3	Benefit	Jumlah Musuh yang dikalahkan
4.	C4	<i>Benefit</i>	HP tersisa
5.	C5	<i>Cost</i>	Jumlah Kematian
6.	C6	<i>Cost</i>	Jumlah Damage yang diterima
7.	C7	<i>Benefit</i>	Jumlah Buff yang ditemukan
8.	C8	<i>Cost</i>	Jumlah Debuff yang didapatkan

Setiap kriteria memiliki dampak penting dalam memilih cuaca yang tepat untuk level selanjutnya, dengan bobot yang akan disesuaikan berdasarkan nilai akhir. Berikut adalah skala penilaian untuk setiap kriteria:

#### 1. Waktu Bermain

Waktu bermain adalah kriteria pertama dalam *game* "Salah Serenity". Pemain harus cepat keluar dari labirin, karena semakin lama mereka berada di dalam, semakin rendah nilai yang diperoleh. Berikut adalah skala nilai untuk kriteria waktu yang dihabiskan dalam menit.

Tabel 3. 4 Tabel Penilaian Kriteria Waktu Yang Dhabiskan

No	Subkriteria	Menit	Skala Nilai
1.	Kurang Baik	$> 4$	1
2.	Cukup	$\leq 4$	2
3.	Cukup Baik	$\leq 3$	3
4.	Baik	$\leq 2$	4
5.	Sangat Baik	$\leq 1$	5

## 2. Petunjuk Ditemukan

Di dalam labirin, pemain harus mencari petunjuk yang tersembunyi dalam kotak kardus berwarna coklat. Setiap kotak bisa berisi dua hal: petunjuk tentang tata cara salat atau debuff yang dapat merugikan pemain. Berikut adalah skala nilai untuk kriteria petunjuk yang ditemukan.

Tabel 3. 5 Tabel Penilaian Petunjuk Yang Ditemukan

No	Subkriteria	Petunjuk	Skala Nilai
1.	Sangat Baik	$> 4$	5
2.	Baik	4	4
3.	Cukup Baik	3	3
4.	Cukup	2	2
5.	Kurang Baik	1	1

## 3. Jumlah Musuh yang Dikalahkan

Selama berada di labirin, pemain akan menghadapi musuh yang menghalangi jalan keluar, sehingga mereka perlu mengalahkan musuh dengan senjata yang dimiliki. Setiap musuh yang dikalahkan akan menambah skor untuk kriteria ini. Berikut adalah skala nilai untuk kriteria musuh yang dikalahkan.

Tabel 3. 6 Tabel Penilaian Musuh Yang Dikalahkan

No	Subkriteria	Musuh dikalahkan	Skala Nilai
1.	Sangat Baik	$> 4$	5
2.	Baik	4	4
3.	Cukup Baik	3	3
4.	Cukup	2	2
5.	Kurang Baik	1	1

#### 4. HP Tersisa

HP tersisa merupakan kriteria yang bernilai benefit. Semakin banyak HP yang tersisa saat *player* keluar dari labirin akan semakin baik nilai performanya. HP *player* dapat berkurang saat *player* terkena debuff atau diserang musuh selama berada di dalam labirin. Darah *player* akan kembali menjadi penuh saat pemain mati dan memilih opsi untuk respawn. Saat respawn pemain akan berada di pintu awal masuk labirin dan harus mencari jalan dari awal dengan waktu yang terus berjalan. Berikut skala nilai dari kriteria HP tersisa.

Tabel 3. 7 Tabel Penilaian HP Yang Tersisa

No	Subkriteria	HP	Skala Nilai
1.	Sangat Baik	$\geq 80$	5
2.	Baik	$\geq 60$	4
3.	Cukup Baik	$\geq 40$	3
4.	Cukup	$\geq 20$	2
5.	Kurang Baik	$\geq 1$	1

#### 5. Jumlah Kematian

Ketika darah pemain habis di labirin, pemain akan mati dan respawn di titik awal dengan darah kembali ke 100. Jumlah kematian pemain akan terakumulasi, dan semakin sering mati, semakin lama waktu yang dibutuhkan untuk keluar dari labirin, yang juga berdampak pada kriteria waktu yang dihabiskan. Berikut adalah skala nilai untuk kriteria jumlah kematian.

Tabel 3. 8 Tabel Penilaian Jumlah Kematian

No	Subkriteria	Jumlah Kematian	Skala Nilai
1.	Sangat Baik	$< 1$	5
2.	Baik	1	4
3.	Cukup Baik	2	3
4.	Cukup	3 – 4	2
5.	Kurang Baik	$> 4$	1

## 6. Jumlah Damage Diterima

Saat permainan dimulai, pemain memiliki 100 HP. HP ini akan berkurang setiap kali terkena damage di labirin, dan total damage yang diterima akan terakumulasi selama level tersebut. Kriteria damage yang diterima berkaitan langsung dengan jumlah kematian pemain.

Tabel 3. 9 Tabel Penilaian Jumlah Damage Yang Diterima

No	Subkriteria	Damage diterima	Skala Nilai
1.	Sangat Baik	$\geq 0$	5
2.	Baik	$\geq 200$	4
3.	Cukup Baik	$\geq 300$	3
4.	Cukup	$\geq 400$	2
5.	Kurang Baik	$\geq 500$	1

## 7. Jumlah *Buff* Ditemukan

Pemain dapat meningkatkan *damage* senjatanya dengan mendekati tempat yang memberikan *buff damage*, yang menambah 10 *damage* untuk setiap *buff* yang diperoleh. Berikut adalah skala nilai untuk kriteria *buff*.

Tabel 3. 10 Tabel Penilaian Jumlah *Buff* Yang Ditemukan

No	Subkriteria	Buff didapat	Skala Nilai
1.	Sangat Baik	$> 4$	5
2.	Baik	4	4
3.	Cukup Baik	3	3
4.	Cukup	2	2
5.	Kurang Baik	1	1

## 8. Jumlah *Debuff* yang Didapatkan

Pemain dapat terkena *debuff* dari kotak kardus coklat yang mirip dengan kotak petunjuk. *Debuff* ini dapat memberikan efek merugikan, seperti *stun* yang membuat pemain tidak bisa bergerak selama 5 detik, *slow* yang memperlambat



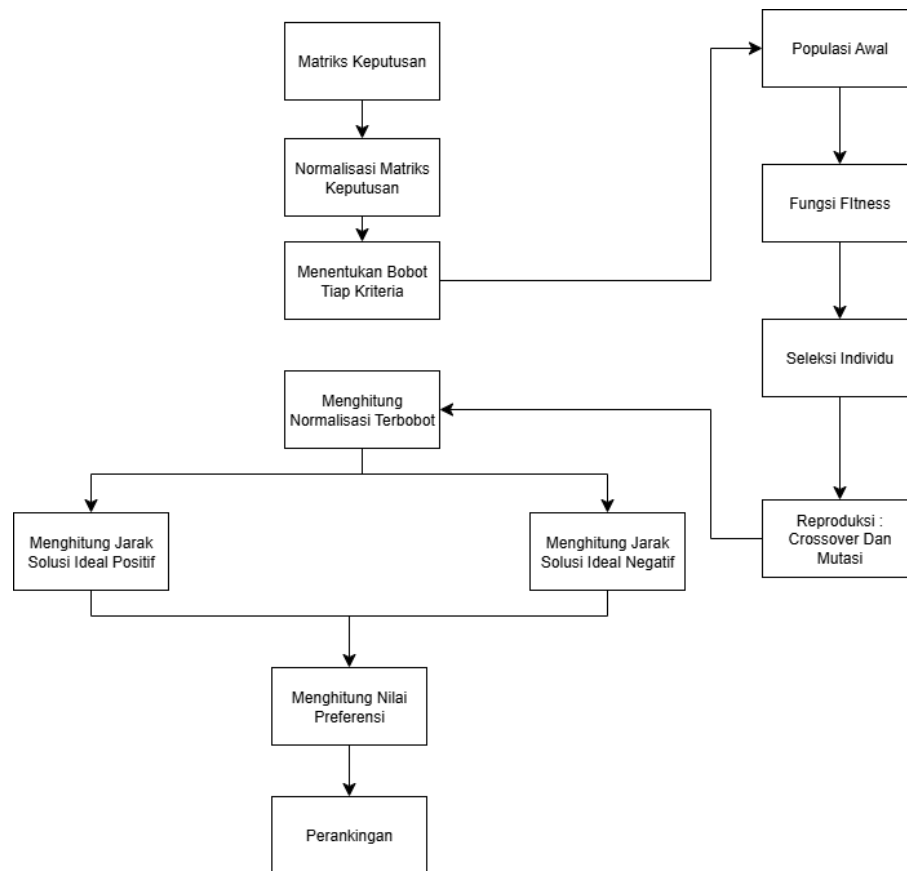
gerakan selama 10 detik, dan damage yang mengurangi HP pemain antara 30 hingga 50. Berikut adalah skala nilai untuk *debuff* yang diterima.

Tabel 3. 11 Tabel Penilaian Jumlah Debuff Yang Didapatkan

No	Subkriteria	Debuff didapat	Skala Nilai
1.	Kurang Baik	$\geq 4$	1
2.	Cukup	3	2
3.	Cukup Baik	2	3
4.	Baik	1	4
5.	Sangat Baik	0	5

### 3.3 Perhitungan Metode GA-TOPSIS

Perhitungan GA-TOPSIS terdiri dari beberapa langkah yang dilakukan. Berikut merupakan langkah-langkah perhitungan yang akan dijelaskan oleh blok diagram dibawah.



Gambar 3. 2 Blok Diagram Metode GA-TOPSIS

Alur perhitungan metode GA-TOPSIS diawali dengan memasukkan data capaian pemain ke dalam matriks keputusan, yang kemudian dinormalisasi untuk menyetarakan nilai dari setiap kriteria. Setelah itu, proses dilanjutkan dengan pencarian bobot kriteria menggunakan GA, di mana serangkaian proses evolusi meliputi inisialisasi populasi, evaluasi *fitness*, seleksi, *crossover*, dan mutasi dijalankan untuk menemukan satu set bobot yang paling optimal. Bobot hasil dari GA ini kemudian digunakan untuk menghitung matriks ternormalisasi terbobot. Berdasarkan matriks terbobot tersebut, sistem menentukan solusi ideal positif dan negatif, menghitung jarak setiap alternatif terhadap kedua solusi ideal, dan pada

akhirnya menghitung nilai preferensi untuk melakukan perankingan. Alternatif cuaca dengan nilai preferensi tertinggi akan dipilih sebagai output akhir dari sistem.

### 3.3.1 Matriks Keputusan

Langkah pertama dalam metode TOPSIS adalah membuat matriks keputusan. Matriks ini berisi data tentang kinerja setiap alternatif, dalam hal ini kondisi cuaca, berdasarkan kriteria yang telah ditentukan. Matriks ini akan menjadi fondasi untuk perhitungan selanjutnya (Reyhan et al., 2019).

Tabel 3. 12 Tabel Matriks Keputusan

No.		C1	C2	C3	C4	C5	C6	C7	C8
1.	A1	5	4	3	3	2	4	5	2
2.	A2	4	5	4	3	1	2	4	3
3.	A3	1	4	5	2	2	3	3	3
4.	A4	5	3	2	4	2	4	2	1
5.	A5	4	2	5	3	3	1	4	2
6.	A6	2	4	3	2	1	3	3	3
7.	A7	3	3	3	5	3	5	2	4
8.	A8	5	4	4	1	4	4	2	3

### 3.3.2 Normalisasi Matriks Keputusan

Setelah matriks keputusan dibuat, langkah berikutnya adalah melakukan normalisasi. Proses ini bertujuan untuk menyamakan skala pengukuran dari berbagai kriteria, sehingga kita bisa membandingkan kriteria dengan lebih adil.

Tabel 3. 13 Tabel Matriks Keputusan Ternormalisasi

No.		C1	C2	C3	C4	C5	C6	C7	C8
1.	A1	0,455	0,380	0,282	0,342	0,289	0,408	0,536	0,256
2.	A2	0,364	0,475	0,376	0,342	0,144	0,204	0,429	0,384
3.	A3	0,091	0,380	0,470	0,228	0,289	0,306	0,322	0,384
4.	A4	0,455	0,285	0,188	0,456	0,289	0,408	0,214	0,128
5.	A5	0,364	0,190	0,470	0,342	0,433	0,102	0,429	0,256
6.	A6	0,182	0,380	0,282	0,228	0,144	0,306	0,322	0,384
7.	A7	0,273	0,285	0,282	0,570	0,433	0,510	0,214	0,512
8.	A8	0,455	0,380	0,376	0,114	0,577	0,408	0,214	0,384

### 3.3.3 Pembobotan Kriteria Menggunakan Genetic Algorithm

Langkah selanjutnya adalah melakukan pembobotan pada setiap kriteria. Pembobotan dalam penelitian ini tidak dilakukan secara subjektif, melainkan menggunakan proses optimasi dengan *Genetic Algorithm* (GA) untuk mendapatkan bobot (W) yang paling objektif dan optimal. Diawali dengan langkah:

#### 1. Pembangkitan populasi awal

Proses evolusi dimulai dengan membangkitkan sekumpulan kromosom secara acak yang disebut populasi. Kromosom ini terdiri dari gen-gen, di mana setiap gen mewakili nilai bobot untuk satu kriteria spesifik.

Ukuran populasi (*pop\_size*) yang ditentukan adalah 50 untuk memastikan keragaman genetik yang cukup. Untuk setiap kromosom akan membangkitkan 8 nilai acak antara 0 dan 1. Berikut adalah contoh inisialisasi populasi awal dengan 4 kromosom pada tabel 3.14.

Tabel 3. 14 Tabel Populasi Awal

	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>G4</b>	<b>G5</b>	<b>G6</b>	<b>G7</b>	<b>G8</b>
<b>Kromosom 1</b>	0.12	0.08	0.15	0.20	0.11	0.09	0.17	0.08
<b>Kromosom 2</b>	0.25	0.10	0.05	0.10	0.20	0.15	0.05	0.10
<b>Kromosom 3</b>	0.07	0.13	0.11	0.09	0.15	0.12	0.21	0.12
<b>Kromosom 4</b>	0.10	0.20	0.10	0.15	0.05	0.20	0.10	0.10

#### 2. Penentuan fungsi fitness

Langkah selanjutnya adalah mengevaluasi fitness. Fungsi fitness dalam penelitian ini menggunakan mekanisme evaluasi kualitas setiap kromosom melalui metode TOPSIS. Fungsi fitness ini dirancang untuk mengukur seberapa baik sebuah set bobot (kromosom) dalam menjadikan cuaca target sebagai pilihan utama. Contoh hasil perhitungan dicantumkan pada tabel 3.15 dibawah.

Tabel 3. 15 Tabel Evaluasi Fitness

	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>G4</b>	<b>G5</b>	<b>G6</b>	<b>G7</b>	<b>G8</b>	<b>CCi</b>
<b>Kromosom 1</b>	0.12	0.08	0.15	0.20	0.11	0.09	0.17	0.08	0.580
<b>Kromosom 2</b>	0.25	0.10	0.05	0.10	0.20	0.15	0.05	0.10	0.582
<b>Kromosom 3</b>	0.07	0.13	0.11	0.09	0.15	0.12	0.21	0.12	0.668
<b>Kromosom 4</b>	0.10	0.20	0.10	0.15	0.05	0.20	0.10	0.10	0.635

### 3. Penentuan seleksi individu

Sebelum melakukan seleksi, Total nilai fitness dan perhitungan probabilitas kumulatif diperlukan untuk menentukan peluang relatif setiap individu dipilih. Probabilitas ini menunjukkan kemungkinan setiap alternatif untuk bertahan dan berkembang dalam generasi berikutnya. Berikut perhitungan total fitness dan probabilitas seleksi.

$$Total\ Fitness = \sum_{i=1}^n Fitness_i$$

$$= 0.580 + 0.582 + 0.668 + 0.635 = 2.465$$

Untuk perhitungan probabilitas kumulatif dengan menjumlahkan semua nilai probabilitas sebelumnya secara bertahap akan dicantumkan pada tabel 3.16.

Tabel 3. 16 Tabel Hasil Seleksi Kromosom

	<b>Fitness (CCi)</b>	<b>Probabilitas (Pi)</b>	<b>Probabilitas Kumulatif</b>
<b>Kromosom 1</b>	0.580	$0.580 / 2.465 = 0.235$	0.235
<b>Kromosom 2</b>	0.582	$0.582 / 2.465 = 0.236$	$0.235 + 0.236 = 0.471$
<b>Kromosom 3</b>	0.668	$0.668 / 2.465 = 0.271$	$0.236 + 0.271 = 0.742$
<b>Kromosom 4</b>	0.635	$0.635 / 2.465 = 0.258$	$0.271 + 0.258 = 1.000$

Langkah Selanjutnya yaitu seleksi kromosom menggunakan *Roulette Wheel Selection*, semakin tinggi nilai fitness suatu individu semakin besar kemungkinan untuk terpilih. Berikut 2 contoh individu terbaik dihasilkan berdasarkan *Roulette Wheel Selection* (menggunakan bilangan acak `RAND()`):

- a. Induk 1 : 0.915 = masuk dalam rentang 0.742 - 1.000 yang berarti kromosom 4 terpilih.
- b. Induk 2 : 0.552 = masuk dalam rentang 0.471 – 0.742 yang berarti kromosom 3 terpilih.

#### 4. Proses Reproduksi (crossover dan mutasi)

Langkah selanjutnya yaitu proses *crossover* dimana setelah kromosom 4 (Induk 1) dan kromosom 3 (Induk 2) terpilih melalui proses seleksi, keduanya akan direproduksi untuk menghasilkan satu individu anak (*offspring*). Proses ini bertujuan untuk mengkombinasikan materi genetik (nilai-nilai bobot) dari kedua induk menggunakan metode *Arithmetic Crossover*. Metode ini menghasilkan anak yang gen-gennya merupakan hasil kombinasi dari gen-gen induknya.

Berikut rumus serta perhitungan crossover :

$$\vec{W}_{\text{child}} = (0.6 * \text{GenInduk}_i) + (0.4 * \text{GenInduk}_{2i})$$

Di mana:

- 0.6 adalah bilangan acak dari faktor persilangan ( $\alpha$ ) antara 0 dan 1.

Pada tabel 3.17 dibawah akan dicantumkan contoh hasil dari perhitungan crossover.

Tabel 3. 17 Tabel Hasil Perhitungan Crossover

	G1	G2	G3	G4	G5	G6	G7	G8
Induk 1 (gen 4)	0.100	0.200	0.100	0.150	0.050	0.200	0.100	0.100
Induk 2 (gen 3)	0.070	0.130	0.110	0.090	0.150	0.120	0.210	0.120
0.6 * GenInduk <sub>i</sub>	0.060	0.120	0.060	0.090	0.030	0.120	0.060	0.060
0.4 * GenInduk <sub>2i</sub>	0.028	0.052	0.044	0.036	0.060	0.048	0.084	0.048
Anak (Hasil crossover)	0.088	0.172	0.104	0.126	0.090	0.168	0.144	0.108

Langkah berikutnya yaitu mutasi dengan tujuan mencegah algoritma terjebak pada solusi yang seragam (*premature convergence*) dan memastikan area

pencarian solusi tetap luas. Proses mutasi terjadi berdasarkan laju mutasi (*mutation rate*), yaitu probabilitas kecil (misalnya 5% atau 0.05) di mana sebuah gen akan dipilih untuk diubah nilainya secara acak.

Pada tabel 3.18 dibawah telah dicantumkan contoh proses perhitungan mutasi yang dimana gen 5 terpilih secara acak untuk mengalami mutasi.

Tabel 3. 18 Tabel Hasil Mutasi

Proses	G1	G2	G3	G4	G5	G6	G7	G8	Total
Anak awal (hasil crossover)	0.088	0.172	0.104	0.126	0.090	0.168	0.144	0.108	1.000
Mutasi	-	-	-	-	-0.030	-	-	-	
Anak setelah mutasi	0.088	0.172	0.104	0.126	0.060	0.168	0.144	0.108	0.970
Normalisasi ulang	0.088/0.970	0.172/0.970	0.104/0.970	0.126/0.970	0.060/0.970	0.168/0.970	0.144/0.970	0.108/0.970	
Anak Final	0.091	0.177	0.107	0.130	0.062	0.173	0.148	0.111	1.000

Setelah melalui seluruh proses seleksi, *crossover*, dan mutasi, sebuah individu (kromosom) yang sepenuhnya baru telah berhasil diciptakan. Proses dari seleksi hingga mutasi ini diulang kembali untuk menciptakan individu-individu baru lainnya hingga ukuran populasi generasi baru terpenuhi. Siklus evolusi ini terus berlanjut hingga jumlah generasi maksimum tercapai, di mana pada akhirnya akan didapatkan satu set bobot terbaik sebagai hasil akhir.

Berikut contoh hasil bobot optimal hingga generasi ke-100 yang dicantumkan pada tabel dibawah.

Tabel 3. 19 Tabel Hasil Optimasi Bobot

	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>	<b>Generasi</b>
<b>Bobot (W)</b>	0.091	0.177	0.107	0.130	0.062	0.173	0.148	0.111	1
	0.081	0.075	0.093	0.255	0.215	0.112	0.099	0.070	50
	0.051	0.185	0.101	0.125	0.085	0.179	0.163	0.111	100

### 3.3.4 Matriks Keputusan Ternormalisasi dan Terbobot

Tahap selanjutnya adalah melakukan pembobotan terhadap matriks keputusan ternormalisasi. Bobot yang digunakan dalam perhitungan ini merupakan bobot optimal yang dihasilkan dari proses *Genetic Algorithm* (GA), proses pembobotan dilakukan dengan mengalikan setiap nilai kriteria pada matriks ternormalisasi dengan bobot kriteria yang bersesuaian.

Tabel 3. 20 Tabel Ternormalisasi Dan Terbobot

No.		<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
1.	<b>A1</b>	0.023	0.070	0.029	0.043	0.025	0.073	0.087	0.028
2.	<b>A2</b>	0.019	0.088	0.038	0.043	0.012	0.037	0.070	0.043
3.	<b>A3</b>	0.005	0.070	0.048	0.028	0.025	0.055	0.052	0.043
4.	<b>A4</b>	0.023	0.053	0.019	0.057	0.025	0.073	0.035	0.014
5.	<b>A5</b>	0.019	0.035	0.048	0.043	0.037	0.018	0.070	0.028
6.	<b>A6</b>	0.009	0.070	0.029	0.028	0.012	0.055	0.052	0.043
7.	<b>A7</b>	0.014	0.053	0.029	0.071	0.037	0.091	0.035	0.057
8.	<b>A8</b>	0.023	0.070	0.038	0.014	0.049	0.073	0.035	0.043

### 3.3.5 Solusi Ideal Positif & Negatif

Solusi ideal positif mencerminkan kondisi terbaik yang bisa dicapai (nilai maksimum untuk kriteria yang menguntungkan dan nilai minimum untuk kriteria biaya), sementara solusi ideal negatif menunjukkan kondisi terburuk. Solusi ideal ini akan menjadi acuan untuk mengukur seberapa dekat setiap alternatif dengan kondisi ideal tersebut.



Tabel 3. 21 Tabel SIP &amp; SIN

No.		C1	C2	C3	C4	C5	C6	C7	C8
1.	A+	0.005	0.088	0.048	0.071	0.012	0.018	0.087	0.014
2.	A-	0.023	0.035	0.019	0.014	0.049	0.091	0.035	0.057

### 3.3.6 Jarak Solusi Ideal Positif & Negatif

Langkah selanjutnya adalah menghitung jarak setiap alternatif dari kedua solusi tersebut menggunakan rumus jarak Euclidean. Semakin dekat jarak suatu alternatif ke solusi ideal positif dan semakin jauh dari solusi ideal negatif, semakin baik alternatif tersebut.

Tabel 3. 22 Tabel Di+ &amp; Di-

No.		Di+	Di-
1.	A1	0.072	0.081
2.	A2	0.050	0.099
3.	A3	0.075	0.071
4.	A4	0.092	0.070
5.	A5	0.070	0.096
6.	A6	0.077	0.070
7.	A7	0.110	0.062
8.	A8	0.109	0.046

### 3.3.7 Nilai Preferensi dan Perankingan

Setelah menghitung jarak, kita mendapatkan nilai preferensi yang menunjukkan seberapa dekat setiap alternatif dengan solusi ideal positif dibandingkan dengan solusi ideal negatif. Nilai ini digunakan untuk meranking setiap alternatif. Langkah terakhir dalam metode TOPSIS adalah menentukan peringkat berdasarkan nilai preferensi yang telah dihitung. Alternatif dengan nilai tertinggi akan menjadi pilihan terbaik dan menempati peringkat pertama.

Tabel 3. 23 Nilai Preferensi Dan Perankingan

No.	Alternatif	Preferensi (V)	Rank
1.	Cerah Terik	0.531	4
2.	Cerah Berawan	0.662	2
3.	Hujan Ringan	0.486	3
4.	Hujan Sedang	0.431	5
5.	Mendung	0.578	7
6.	Kabut	0.478	1
7.	Hujan Deras	0.361	6
8.	Hujan Deras + Kabut	0.297	8

### 3.4 Validasi Model GA-TOPSIS

Untuk memastikan ketangguhan (*robustness*) model, penelitian ini mengadopsi metode *10-Fold Cross-Validation*. Metode ini dipilih karena kemampuannya dalam memberikan estimasi kinerja model yang lebih akurat dan stabil dengan memanfaatkan keseluruhan dataset untuk proses pelatihan (*training*) dan pengujian (*testing*) secara bergantian.

Proses ini dapat membantu menentukan apakah model memiliki kinerja yang baik dan mencegah *overfitting* atau *underfitting*. Dataset akan dibagi menjadi *10-fold* dengan ukuran yang sama pada setiap foldnya, sehingga memiliki 10 subset data.

Kelebihan menggunakan 10-Fold Cross Validation adalah dalam rangka untuk peningkatan estimasi kinerja yang lebih stabil dan akurat dengan cara menghitung rata rata hasil dari “10-fold” iterasi.

### 3.5 Desain Pengujian Usability

Salah satu metode evaluasi yang digunakan penelitian ini adalah *System Usability Scale*. *System Usability Scale* (SUS) merupakan suatu metode yang bersifat *quick and dirty*. Metode ini umum digunakan untuk menilai tingkat

kegunaan suatu aplikasi atau sistem dengan menggunakan kuesioner (Ridwan et al., 2024).

Dalam penelitian ini, SUS digunakan untuk mengevaluasi dampak fungsional dari sistem penentuan cuaca dinamis terhadap pengalaman pengguna. Meskipun sistem kombinasi *Genetic Algorithm* dan TOPSIS berjalan di latar belakang, output yang dihasilkannya yaitu perubahan cuaca secara langsung memengaruhi visual, tantangan, dan suasana yang dirasakan pemain. SUS digunakan sebagai alat ukur kuantitatif untuk memvalidasi apakah implementasi sistem cerdas ini dapat terintegrasi secara mulus ke dalam permainan tanpa menimbulkan kebingungan atau frustrasi yang dapat menurunkan tingkat usability.

Tabel 3. 24 Tabel Pertanyaan SUS

No	Kode	Pertanyaan
1.	SUS 1	Saya berpikir bahwa saya akan sering memainkan game ini karena sangat mudah dalam penggunaannya.
2.	SUS 2	Saya merasa kontrol game ini terlalu rumit untuk saya.
3.	SUS 3	Saya merasa kontrol game ini mudah digunakan.
4.	SUS 4	Saya memerlukan bantuan orang lain untuk menggunakan fitur dalam game ini.
5.	SUS 5	Saya merasa fitur ini memiliki fungsi yang sudah terintegrasi dengan baik.
6.	SUS 6	Saya merasa terdapat ketidakkonsistenan dalam fitur di game ini.
7.	SUS 7	Saya yakin orang lain akan mudah memahami navigasi pada game ini.
8.	SUS 8	Saya merasa fitur dan sistem game ini membingungkan.
9.	SUS 9	Saya merasa percaya diri saat menggunakan fitur ini.
10.	SUS 10	Saya harus banyak belajar sebelum dapat memahami gameplay pada game ini.

Tabel 3. 25 Tabel Interpretasi Skor SUS

<b>Skor</b>	<b>Keterangan</b>
5	Sangat Memuaskan
4	Puas
3	Biasa Saja
2	Tidak Puas
1	Sangat Mengecewakan

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

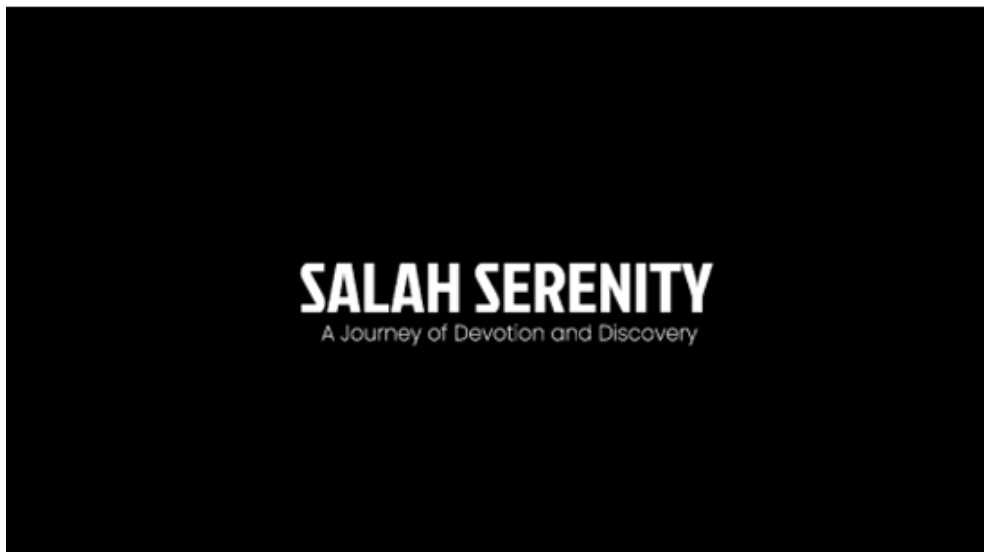
Bab ini membahas proses implementasi dan pengujian yang telah dirancang pada bab sebelumnya. Penjelasan diawali dengan penerapan antarmuka *game* “Salah Serenity” berdasarkan *storyboard* yang telah dibuat. Pengujian metode GA dan TOPSIS untuk penyesuaian cuaca, uji coba *game*, hasil validasi metode GA-TOPSIS dan pengujian usability menggunakan metode SUS (*System Usability Scale*).

#### **4.1 Implementasi *Game***

Implementasi *game* merupakan tahap penerapan antarmuka pengguna (UI) berdasarkan *storyboard* yang telah dijelaskan pada bab sebelumnya. Tahap ini melibatkan proses penggabungan berbagai elemen visual dan fungsional yang telah dirancang dalam *storyboard* ke dalam *game* “Salah Serenity”. Dengan penerapan UI yang optimal, pengalaman bermain pemain akan menjadi lebih menarik dan selaras dengan konsep serta rancangan yang telah ditetapkan sebelumnya.

##### **1. Halaman Awal**

Halaman awal *game* “Salah Serenity” menampilkan antarmuka utama yang mencakup *opening game*, menu utama, serta tutorial sebagai tahap awal interaksi pemain sebelum memulai permainan.



Gambar 4. 1 Halaman Intro

Gambar 4.1 menunjukkan *scene* awal yang muncul saat pemain memulai permainan. Pada tahap ini, pemain diperkenalkan dengan latar belakang dan tujuan game, yaitu kisah seorang pemuda yang berusaha menunaikan salat di masjid. Pemain akan melewati sebuah labirin penuh rintangan dan harus menemukan jalur menuju masjid, sambil menghadapi berbagai tantangan dari NPC musuh yang berusaha menghalangi perjalanan tersebut.



Gambar 4. 2 Halaman Main Menu

Gambar 4.2 menampilkan tampilan awal yang muncul setelah tahap pengenalan *game*. Pada bagian ini, pemain disajikan tiga opsi utama, yaitu *Play Games*, *Option*, dan *Exit*. Pemain dapat memilih untuk memulai permainan, membuka menu pengaturan guna menyesuaikan preferensi permainan, atau keluar dari *game*. Desain menu utama dibuat sederhana dan intuitif agar memudahkan pemain dalam melakukan navigasi.



Gambar 4. 3 *Panel Tutorial*

Gambar 4.3 memperlihatkan tampilan panduan kontrol yang berisi informasi mengenai cara menggerakkan karakter dengan tombol WASD, melompat menggunakan tombol spasi, serta mengatur arah pandangan melalui gerakan mouse. Tampilan ini ditampilkan sebelum permainan dimulai untuk membantu pemain memahami kontrol dasar yang digunakan dalam *game*.

## 2. Tampilan *Gameplay*

Setelah menyelesaikan tutorial, pemain akan langsung memasuki *gameplay* utama. Di dalamnya, akan ditampilkan *User Interface* (UI) yang memandu pemain

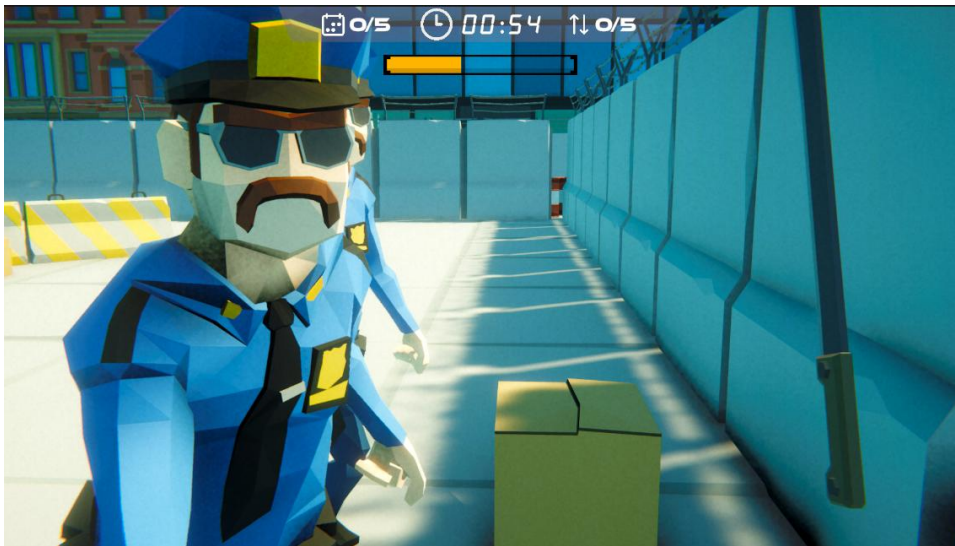
selama permainan, seperti indikator *health point*, waktu yang telah digunakan, jumlah petunjuk yang telah diperoleh, dan informasi penting lainnya.



Gambar 4. 4 Scene Gameplay

Dalam *gameplay*, pemain akan menemukan berbagai objek, salah satunya adalah kardus kotak. Kardus ini memiliki isi yang ditentukan secara acak; saat dibuka, bisa berisi petunjuk yang menguntungkan pemain atau *Debuff* yang justru memberikan efek merugikan. Selama perjalanan pemain akan di halangi oleh npc aparat yang akan mengganggu selama pemain berada di dalam labirin.





Gambar 4. 5 Kotak Kardus Petunjuk Dan NPC Musuh

Pemain juga dapat menemukan objek meja kayu. Jika pemain berhasil menemukan meja ini dan mengambil senjata yang ada di atasnya, mereka akan memperoleh tambahan damage sebesar 10. Terdapat lima objek meja kayu dalam setiap level, dengan posisi yang diacak setiap kali level dimulai.



Gambar 4. 6 Objek Damage Buff

### 3. *Result Game*

Panel “*Game Result*” akan ditampilkan ketika pemain berhasil keluar dari labirin. Panel ini menampilkan seluruh pencapaian yang diperoleh pemain selama menjelajahi labirin.



Gambar 4. 7 Tampilan Game Result

Nilai pencapaian pemain telah dinormalisasi ke dalam skala 1–5. Pada panel ini juga terdapat dua tombol, yaitu tombol “*Exit*” untuk kembali ke menu utama, dan tombol “*Play Again*” yang akan memicu perhitungan menggunakan metode GA-TOPSIS untuk menyesuaikan kondisi cuaca dalam *gameplay* pada level berikutnya.

#### 4.2 Implementasi GA (*Genetic Algorithm*)

Implementasi metode GA untuk Optimasi bobot kriteria pada penelitian ini menggunakan pemrograman dengan bahasa C# yang diintegrasikan dengan unity. Tujuan utama dari implementasi ini adalah untuk menjalankan proses evolusi yang secara dinamis mencari 8 nilai bobot kriteria yang paling optimal. Bobot optimal

inilah yang kemudian akan digunakan oleh metode TOPSIS untuk menentukan cuaca yang paling sesuai dengan capaian pemain.

Proses GA ini nantinya akan dipanggil oleh *controller* utama (PenyesuaianCuaca.cs) setelah pemain menyelesaikan level. Tujuannya untuk menemukan bobot optimal untuk TOPSIS.

#### a. Populasi Awal

Pseudocode 4. 1 Populasi Awal

```
private void InitializePopulation()
{
    for (int i = 0; i < populationSize; i++)
    {
        // Buat kromosom baru

        var chromosome = new Chromosome(numGenes);

        // Isi 8 gen (bobot) dengan nilai acak (0.0 - 1.0)
        for (int j = 0; j < numGenes; j++)
        {
            chromosome.Genes[j] = random.NextDouble();
        }

        // Pastikan total 8 bobot = 1.0
        Normalize(chromosome.Genes);

        population.Add(chromosome);
    }
}

private void Normalize(double[] genes)
{
    double sum = genes.Sum();

    if (sum > 0)
```

```

{
    for (int i = 0; i < genes.Length; i++)
    {
        genes[i] /= sum; // Bagi setiap gen dengan total
    }
}
/

```

Proses GA dimulai dengan membuat populasi awal. Pada Fungsi ini, setiap gen (bobot) dari 8 kriteria diisi dengan nilai acak. Langkah *Normalize* kemudian dijalankan. Fungsi ini menjumlahkan 8 nilai bobot acak tersebut, lalu membagi setiap bobot dengan totalnya. Proses ini memastikan bahwa total 8 bobot dari setiap kromosom adalah 1.0, yang merupakan syarat dalam metode pembobotan

#### b. Fungsi Fitness

##### Pseudocode 4. 2 Fungsi Fitness

```

private void CalculatePopulationFitness()
{
    foreach (var chromosome in population)
    {
        // 1. Menjalankan simulasi TOPSIS menggunakan bobot (Genes) dari kromosom
        var topsis = new TopsisCalculator(this.decisionMatrix, chromosome.Genes);
        var preferences = topsis.Calculate();

        // 2. Fitness = nilai preferensi (CCi) dari cuaca target
        chromosome.Fitness = preferences[this.targetAlternativeIndex];
    }
}

```

Evaluasi individu dilakukan oleh fungsi *CalculatePopulationFitness*. Fungsi ini bertujuan untuk mengukur performa setiap kromosom dalam populasi dengan mengintegrasikan metode TOPSIS. Fungsi ini melakukan iterasi pada setiap kromosom dalam populasi. Untuk setiap kromosom, sebuah kalkulasi TOPSIS penuh dijalankan dengan menggunakan himpunan bobot (*chromosome.Genes*) dari kromosom tersebut sebagai *input* bobot kriteria. Hasil dari kalkulasi TOPSIS adalah *array preferences* yang berisi 8 nilai preferensi (skor CCI) untuk 8 alternatif cuaca. Nilai *fitness* untuk kromosom yang sedang dievaluasi kemudian ditetapkan sebagai nilai CCI dari *targetAlternativeIndex* (indeks cuaca target yang telah ditentukan).

### c. Seleksi

Pseudocode 4. 3 Seleksi

```
private Chromosome SelectParent()
{
    double totalFitness = population.Sum(c => c.Fitness);
    if (totalFitness == 0) return population[random.Next(population.Count)];
    // Memilih titik acak pada "roda"
    double randomPoint = random.NextDouble() * totalFitness;
    // Memutar roda
    foreach (var chromosome in population)
    {
        randomPoint -= chromosome.Fitness;
        if (randomPoint <= 0) return chromosome;
    }
    // Kromosom terpilih
    return population.Last();
}
```

Setelah fitness dihitung, GA memilih "orang tua" (*parent*) untuk menghasilkan generasi baru. Metode yang digunakan adalah Roulette Wheel Selection. Metode *SelectParent* mengimplementasikan seleksi *Roulette Wheel*. Pertama, total *fitness* dari seluruh populasi dihitung. Kemudian, sebuah random Point dipilih antara 0 dan total Fitness. Sistem melakukan iterasi pada populasi, mengurangi randomPoint dengan *fitness* setiap kromosom. Kromosom pertama yang menyebabkan *randomPoint* menjadi  $\leq 0$  adalah yang terpilih. Ini secara proporsional memberikan kesempatan lebih besar bagi kromosom dengan *fitness* tinggi untuk terpilih.

#### d. Crossover

Pseudocode 4. 4 *Crossover*

```
private Chromosome Crossover(Chromosome p1, Chromosome p2)
{
    var child = new Chromosome(numGenes);
    double a = random.NextDouble();
    // Faktor blend (alpha) acak
    for (int i = 0; i < numGenes; i++)
    {
        // Gen anak adalah campuran/rata-rata terbobot dari gen orang tua
        child.Genes[i] = a * p1.Genes[i] + (1 - a) * p2.Genes[i];
    }
    return child;
}
```

Dua *parent* yang terpilih akan menghasilkan keturunan (*offspring*) melalui operator *crossover*. Metode yang diimplementasikan adalah *Arithmetic Crossover*, yang cocok untuk gen bernilai riil (*real-valued genes*). Fungsi Crossover menghasilkan satu kromosom anak baru dari dua kromosom induk (p1 dan p2). Sebuah faktor *blending* alpha ( $\alpha$ ) dibangkitkan secara acak untuk setiap operasi *crossover*. Gen anak (*child.Genes[i]*) kemudian dikalkulasi sebagai kombinasi linear (rata-rata terbobot) dari gen kedua induknya.

#### e. Mutasi

Pseudocode 4. 5 Mutasi

```
private void Mutate(Chromosome chromosome)
{
    for (int i = 0; i < numGenes; i++)
    {
        // Probabilitas mutasi (mutationRate) adalah 5% (0.05)
        if (random.NextDouble() < mutationRate)
        {
            // Menggeser nilai gen sedikit (antara -0.1 s/d +0.1)
            chromosome.Genes[i] += (random.NextDouble() - 0.5) * 0.2;
            // Memastikan nilai gen tidak menjadi negatif
            if (chromosome.Genes[i] < 0) chromosome.Genes[i] = 0;
        }
    }
}
```

Untuk mencegah konvergensi prematur dan memperkenalkan keragaman baru, operator mutasi diterapkan pada kromosom anak. Fungsi *Mutate* melakukan

iterasi pada setiap gen dalam kromosom anak. Untuk setiap gen, sebuah bilangan acak dibandingkan dengan *mutationRate* (0.05). Jika bilangan acak tersebut lebih kecil dari *mutationRate*, mutasi akan terjadi pada gen tersebut.

Mutasi diimplementasikan sebagai *creep mutation*, di mana nilai gen tidak diganti total, melainkan digeser (ditambah atau dikurangi) dengan nilai acak kecil (dalam rentang  $[-0.1, +0.1]$ ). Ini adalah mekanisme eksplorasi lokal yang membantu GA melakukan *fine-tuning* (penyetelan halus) pada solusi dan menghindari jebakan *optimum lokal*.

#### 4.3 Implementasi TOPSIS

Metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) diimplementasikan sebagai *engine* pengambilan keputusan utama. Metode ini bertugas untuk merangking 8 alternatif cuaca berdasarkan 8 kriteria capaian pemain, dengan menggunakan bobot kriteria yang telah dioptimasi oleh GA (*Genetic Algorithm*).

##### 1. Matriks Keputusan

Pseudocode 4. 6 Matriks Keputusan

```

For each kriteria j:
    Calculate sum of squares for all alternatif i in column j.
    sqrtSum = sqrt(sum of squares)
    For each alternatif i:
        matriksNormalisasi[i][j] = matriksKeputusan[i][j] /
sqrtSum

```

Dalam Matriks keputusan ini akan menentukan jumlah kriteria yang digunakan dalam perhitungan TOPSIS. Matriks Keputusan merupakan representasi



data awal yang mendefinisikan dimensi perhitungan TOPSIS, yaitu jumlah alternatif dan kriteria. Matriks ini menyajikan gambaran kuantitatif performa setiap alternatif terhadap setiap kriteria yang telah ditetapkan. Nilai-nilai mentah inilah yang kemudian akan diproses melalui tahap normalisasi dan pembobotan.

## 2. Matriks Ternormalisasi Terbobot

Pseudocode 4. 7 Matriks Ternormalisasi Terbobot

```
private double[,] CalculateWeightedMatrix(double[,] normalizedMatrix, double[] weights)
{
    int rows = normalizedMatrix.GetLength(0);
    int cols = normalizedMatrix.GetLength(1);
    double[,] weightedMatrix = new double[rows, cols];

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            // 1. Mengalikan elemen ternormalisasi (r_ij) dengan bobot (w_j)
            weightedMatrix[i, j] = normalizedMatrix[i, j] * weights[j];
        }
    }

    return weightedMatrix;
}
```

Proses diawali dengan normalisasi matriks keputusan. Matriks ternormalisasi kemudian dikalikan dengan bobot yang dihasilkan oleh GA. Implementasi ini melakukan perkalian skalar antara setiap elemen pada matriks ternormalisasi dengan bobot kriteria yang bersesuaian. Proses ini menghasilkan

matriks terbobot, di mana nilai kriteria yang dianggap lebih penting (bobot lebih tinggi) akan memiliki dampak yang lebih besar pada hasil akhir.

### 3. Solusi Ideal Positif dan Negatif

Pseudocode 4. 8 Solusi Ideal Positif dan Negatif

```
// Solusi Ideal

private void DetermineIdealSolutions(double[,] weightedMatrix)
{
    // Asumsi: A_plus dan A_minus adalah array [8]
    // Asumsi: isBenefit = [true, true, true, false, true, true, true, false] ...

    for (int j = 0; j < numCriteria; j++)
    {
        double maxVal = double.MinValue;
        double minVal = double.MaxValue;

        for (int i = 0; i < numAlternatives; i++)
        {
            if (weightedMatrix[i, j] > maxVal) maxVal = weightedMatrix[i, j];
            if (weightedMatrix[i, j] < minVal) minVal = weightedMatrix[i, j];
        }

        if (isBenefit[j]) // Jika kriteria adalah Benefit (semakin besar semakin baik)
        {
            A_plus[j] = maxVal;
            A_minus[j] = minVal;
        }
        else // Jika kriteria adalah Cost (semakin kecil semakin baik)
```

```

    {
        A_plus[j] = minVal;
        A_minus[j] = maxVal;
    }
}
}

```

Langkah selanjutnya adalah menentukan nilai terbaik dan terburuk untuk setiap kriteria dari matriks terbobot. Penentuan ini bergantung pada sifat kriteria (*Benefit* atau *Cost*). Fungsi ini akan menganalisis setiap kolom (kriteria) pada matriks terbobot. Jika kriteria tersebut bersifat *Benefit* maka nilai A+ adalah nilai maksimum di kolom itu dan A- adalah nilai minimum. Sebaliknya, jika kriteria bersifat *Cost* maka A+ adalah nilai minimum dan A- adalah nilai maksimum.

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

Pseudocode 4. 9 Jarak Solusi Ideal Positif dan Negatif

```

// Jarak Ideal

private void CalculateIdealDistances(double[,] weightedMatrix)
{
    // Asumsi: S_plus dan S_minus adalah array [8]

    for (int i = 0; i < numAlternatives; i++) // Untuk setiap cuaca
    {
        double sumPlus = 0;
        double sumMinus = 0;

        for (int j = 0; j < numCriteria; j++) // Untuk setiap kriteria
        {
            // Jarak ke A+

```

```

        sumPlus += Math.Pow(weightedMatrix[i, j] - A_plus[j], 2);
// Jarak ke A-
        sumMinus += Math.Pow(weightedMatrix[i, j] - A_minus[j], 2);
    }
    S_plus[i] = Math.Sqrt(sumPlus);
    S_minus[i] = Math.Sqrt(sumMinus);
}
}

```

Setelah A+ dan A- diketahui, langkah berikutnya adalah menghitung jarak *Euclidean* dari setiap alternatif (cuaca) ke Solusi Ideal Positif dan Solusi Ideal Negatif. Implementasi ini menghitung jarak total untuk setiap baris (alternatif cuaca). Untuk setiap cuaca, sistem menghitung selisih kuadrat antara nilai cuaca tersebut dengan nilai ideal (A+ dan A-) di semua 8 kriteria, lalu mengakarkannya. Hasilnya adalah dua nilai untuk setiap cuaca: S+ (jarak ke solusi terbaik) dan S- (jarak ke solusi terburuk).

## 5. Nilai Preferensi

### Pseudocode 4. 10 Preferensi

```

// Nilai Preferensi

private double[] CalculatePreferences()
{
    // Asumsi: preferences (CCi) adalah array [8]

    for (int i = 0; i < numAlternatives; i++)
    {

```

```

double totalDistance = S_plus[i] + S_minus[i];

if (totalDistance > 0)
    preferences[i] = S_minus[i] / totalDistance;
else
    preferences[i] = 0;
}

return preferences;
}

```

Langkah terakhir adalah menghitung nilai preferensi (skor  $CC_i$ ) untuk setiap alternatif. Nilai inilah yang digunakan untuk perankingan. Fungsi ini menerapkan rumus final TOPSIS. Nilai preferensi ( $CC_i$ ) dihitung dengan membagi jarak ke solusi terburuk dengan total jarak. Alternatif dengan nilai  $CC_i$  tertinggi (mendekati 1) adalah alternatif terbaik, karena ia secara relatif paling dekat dengan Solusi Ideal Positif dan paling jauh dari Solusi Ideal Negatif. Setelah proses ini, dilakukan perankingan untuk pilihan yang paling sesuai dalam pengambilan keputusan.

#### 4.4 Uji Coba Penyesuaian Cuaca

Pengujian penyesuaian cuaca dalam *game* ini telah diimplementasikan menggunakan metode GA dan TOPSIS. Tujuan dari implementasi ini adalah untuk memastikan bahwa penyesuaian cuaca dapat dilakukan berdasarkan capaian pemain yang mencakup waktu bermain, petunjuk yang ditemukan, jumlah musuh yang dikalahkan, HP tersisa, jumlah kematian, jumlah *damage* yang diterima, *buff* yang ditemukan, dan *debuff* yang didapatkan. Melalui pengujian ini, dapat diketahui

apakah sistem GA dan TOPSIS yang diterapkan mampu menghasilkan cuaca yang sesuai dengan capaian pemain dan memberikan tantangan bagi setiap pemain.

#### 4.4.1 Pengujian Skenario Pertama

Pada Pengujian Skenario pertama pemain mendapatkan nilai capaian pemain seperti pada gambar dibawah ini.



Gambar 4. 8 Result Skenario Pertama

Pada gambar 4.8 memperlihatkan data capaian pemain yang diambil setelah menyelesaikan permainan. Tampilan *game result* yang muncul pada pengujian skenario pertama menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 3 yang berarti pemain menyelesaikan *game* dalam waktu 2 – 3 menit, petunjuk yang ditemukan mendapatkan nilai 2 yang berarti pemain mendapatkan 2 petunjuk, HP (*Health Point*) mendapatkan nilai 2 yang berarti HP pemain yang tersisa setelah keluar dari labirin sebesar 30 HP, nilai max terdapat pada jumlah musuh yang di kalahkan. Data input dari *game result* ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 1 Optimasi Bobot Skenario Pertama

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.445	0.084	0.271	0.030	0.063	0.014	0.042	0.048

Tabel 4.1 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 2 CCI &amp; Perankingan Skenario Pertama

<b>Alternatif</b>	<b>CCI</b>	<b>Ranking</b>
Cerah Terik	0.191	7
Cerah Berawan	0.377	5
Hujan Ringan	0.891	1
Hujan Sedang	0.138	8
Mendung	0.414	4
Kabut	0.653	2
Hujan Deras	0.461	3
Hujan Deras + Kabut	0.243	6

Tabel 4.2 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Semakin besar CCI maka semakin baik atau semakin layak alternatif tersebut. Dalam skenario pengujian pertama didapatkan hasil CCI terbesar yaitu pada alternatif “Hujan Ringan”.

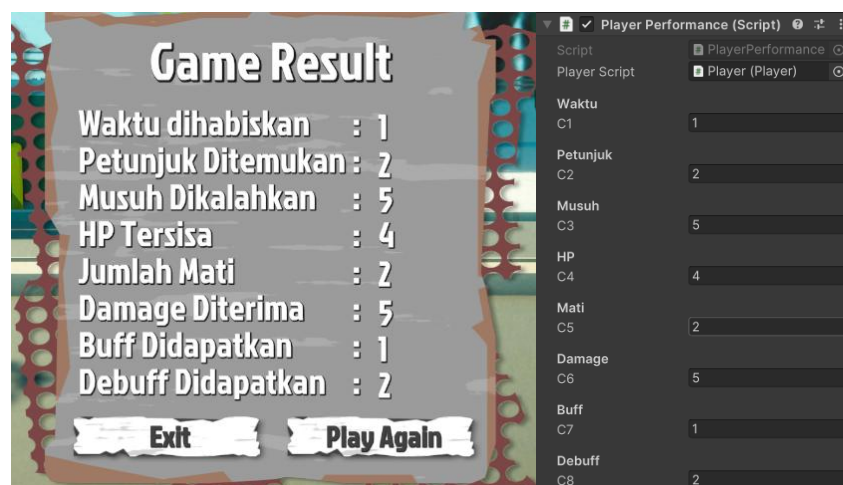


Gambar 4. 9 Hasil Penyesuaian Cuaca Skenario Pertama

Pada gambar 4.9 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Hujan Ringan” yang merupakan alternatif ke-3 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.2 Pengujian Skenario Kedua

Pada pengujian skenario kedua pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 10 Result Skenario Kedua



Pada gambar 4.10 memperlihatkan bahwa pemain mendapatkan skor waktu dengan nilai 1 yang berarti pemain menyelesaikan *game* dalam waktu  $> 4$  menit, petunjuk yang ditemukan mendapatkan skor dengan nilai 2 yang berarti pemain mendapatkan 2 petunjuk, HP tersisa menunjukkan nilai 4 yang berarti pemain dapat menyelesaikan *game* dengan sisa HP sebesar 70 HP. Tampilan *game result* yang muncul pada pengujian skenario kedua menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 3 Optimasi Bobot Skenario Kedua

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.061	0.103	0.067	0.541	0.075	0.04	0.073	0.036

Tabel 4.3 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 4 CCI dan Perankingan Skenario Kedua

<b>ALTERNATIF</b>	<b>CCI</b>	<b>RANKING</b>
Cerah Terik	0.504	4
Cerah Berawan	0.515	3
Hujan Ringan	0.287	7
Hujan Sedang	0.710	2
Mendung	0.496	5
Kabut	0.288	6
Hujan Deras	0.842	1
Hujan Deras + Kabut	0.087	8

Tabel 4.4 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian kedua didapatkan hasil CCI terbesar yaitu pada alternatif “Hujan Deras”.



Gambar 4. 11 Hasil Penyesuaian Cuaca Skenario Kedua

Pada gambar 4.11 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Hujan Deras” yang merupakan alternatif ke-7 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.3 Pengujian Skenario Ketiga

Pada pengujian skenario ketiga pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 12 Result Skenario Ketiga

Pada gambar 4.12 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan seperti gambar tersebut. Tampilan *game result* yang muncul pada pengujian skenario ketiga menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 5 yang berarti pemain menyelesaikan *game* dalam waktu  $< 1$  menit, petunjuk yang ditemukan mendapatkan nilai 4 yang berarti pemain mendapatkan 4 petunjuk, HP (*Health Point*) mendapatkan nilai 4 yang berarti HP pemain yang tersisa setelah keluar dari labirin sebesar 70 HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 5 Optimasi Bobot Skenario Ketiga

Bobot	C1	C2	C3	C4	C5	C6	C7	C8
	0.047	0.187	0.056	0.018	0.067	0.002	0.469	0.151

Tabel 4.5 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap

kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 6 CCI Dan Perankingan Skenario Ketiga

ALTERNATIF	CCI	RANKING
Cerah Terik	0.823	1
Cerah Berawan	0.648	2
Hujan Ringan	0.394	5
Hujan Sedang	0.289	6
Mendung	0.580	3
Kabut	0.396	4
Hujan Deras	0.126	8
Hujan Deras + Kabut	0.206	7

Tabel 4.6 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian ketiga didapatkan hasil CCI terbesar yaitu pada alternatif “Cerah Terik”



Gambar 4. 13 Hasil Penyesuaian Cuaca Skenario Ketiga

Pada gambar 4.13 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Cerah Terik” yang merupakan alternatif ke-1 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.4 Pengujian Skenario Keempat

Pada pengujian skenario keempat pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 14 Result Skenario Keempat

Pada gambar 4.14 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan skor seperti gambar tersebut. Tampilan game result yang muncul pada pengujian skenario keempat menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 1 yang berarti pemain menyelesaikan *game* dalam waktu  $> 4$  menit, petunjuk yang ditemukan mendapatkan nilai 3 yang berarti pemain mendapatkan 3 petunjuk, HP (*Health Point*) mendapatkan nilai 2 yang berarti HP pemain yang tersisa setelah

keluar dari labirin sebesar 30 HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 7 Optimasi Bobot Skenario Keempat

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.129	0.201	0.048	0.032	0.446	0.084	0.056	0.008

Tabel 4.7 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 8 CCI Dan Perankingan Skenario Keempat

<b>ALTERNATIF</b>	<b>CCI</b>	<b>RANKING</b>
Cerah Terik	0.611	4
Cerah Berawan	0.844	2
Hujan Ringan	0.669	3
Hujan Sedang	0.580	5
Mendung	0.344	6
Kabut	0.856	1
Hujan Deras	0.339	7
Hujan Deras + Kabut	0.165	8

Tabel 4.8 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian keempat didapatkan hasil CCI terbesar yaitu pada alternatif “Kabut”.



Gambar 4. 15 Hasil Penyesuaian Cuaca Skenario Keempat

Pada gambar 4.15 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Kabut” yang merupakan alternatif ke-6 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.5 Pengujian Skenario Kelima

Pada pengujian skenario kelima pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 16 Result Skenario Kelima

Pada gambar 4.16 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan skors seperti gambar tersebut. Tampilan game result yang muncul pada pengujian skenario kelima menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 3 yang berarti pemain menyelesaikan *game* dalam waktu 2 - 3 menit, petunjuk yang ditemukan mendapatkan nilai 3 yang berarti pemain mendapatkan 3 petunjuk, HP (*Health Point*) mendapatkan nilai 5 yang berarti HP pemain yang tersisa setelah keluar dari labirin sebesar  $> 80$  HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 9 Optimasi Bobot Skenario Kelima

Bobot	C1	C2	C3	C4	C5	C6	C7	C8
	0.029	0.054	0.031	0.296	0.055	0.014	0.048	0.470

Tabel 4.9 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 10 CCI dan Perankingan Skenario Kelima

ALTERNATIF	CCI	RANKING
Cerah Terik	0.604	2
Cerah Berawan	0.409	5
Hujan Ringan	0.316	7
Hujan Sedang	0.832	1
Mendung	0.598	3
Kabut	0.320	6
Hujan Deras	0.426	4
Hujan Deras + Kabut	0.251	8



Tabel 4.10 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian kelima didapatkan hasil CCI terbesar yaitu pada alternatif “Hujan Sedang”.

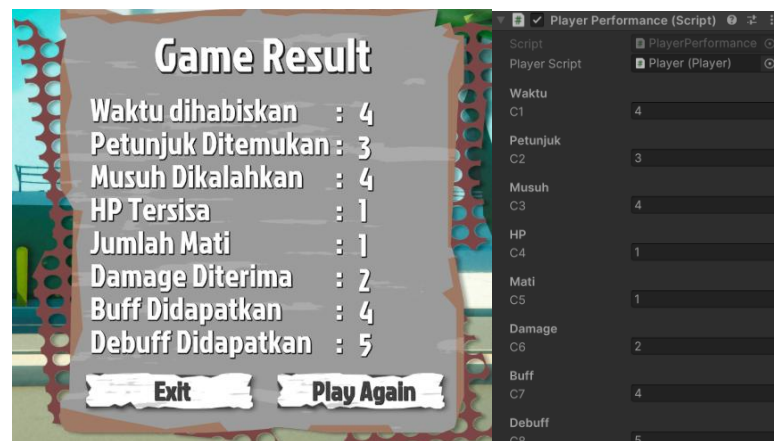


Gambar 4. 17 Hasil Penyesuaian Cuaca Skenario Kelima

Pada gambar 4.17 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Hujan Sedang” yang merupakan alternatif ke-4 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.6 Pengujian Skenario Keenam

Pada pengujian skenario keenam, pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 18 Result Skenario Keenam

Pada gambar 4.18 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan skors seperti gambar tersebut. Tampilan game result yang muncul pada pengujian skenario kelima menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 4 yang berarti pemain menyelesaikan *game* dalam waktu 1 - 2 menit, petunjuk yang ditemukan mendapatkan nilai 3 yang berarti pemain mendapatkan 3 petunjuk, HP (*Health Point*) mendapatkan nilai 1 yang berarti HP pemain yang tersisa setelah keluar dari labirin sebesar  $< 20$  HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 11 Optimasi Bobot Skenario Keenam

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.0112	0.1823	0.1012	0.0313	0.4932	0.0905	0.067	0.0233

Tabel 4.11 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap

kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 12 CCI dan Perankingan Skenario Keenam

ALTERNATIF	CCi	RANKING
Cerah Terik	0.648	4
Cerah Berawan	0.925	1
Hujan Ringan	0.659	3
Hujan Sedang	0.613	5
Mendung	0.364	6
Kabut	0.855	2
Hujan Deras	0.328	7
Hujan Deras + Kabut	0.157	8

Tabel 4.12 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian keempat didapatkan hasil CCI terbesar yaitu pada alternatif “Cerah Berawan”.



Gambar 4. 19 Hasil Penyesuaian Cuaca Skenario Keenam

Pada gambar 4.19 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Cerah Berawan” yang merupakan alternatif ke-2 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.7 Pengujian Skenario Ketujuh

Pada pengujian skenario ketujuh, pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 20 Result Skenario Ketujuh

Pada gambar 4.20 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan skors seperti gambar tersebut. Tampilan *game result* yang muncul pada pengujian skenario kelima menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 5 yang berarti pemain menyelesaikan *game* dalam waktu < 1 menit, petunjuk yang ditemukan mendapatkan nilai 2 yang berarti pemain mendapatkan 2 petunjuk, HP (*Health Point*) mendapatkan nilai 5 yang berarti HP pemain yang tersisa setelah

keluar dari labirin sebesar  $> 80$  HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 13 Optimasi Bobot Skenario Ketujuh

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.0176	0.0198	0.2711	0.0444	0.0154	0.4281	0.1036	0.0999

Tabel 4.13 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Untuk setiap bobot mewakili setiap kriteria (C1 – C8) yang telah didapatkan setelah pemain menyelesaikan *game*. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 14 CCI dan Perankingan Skenario Ketujuh

<b>ALTERNATIF</b>	<b>CCI</b>	<b>RANKING</b>
Cerah Terik	0.320	6
Cerah Berawan	0.709	2
Hujan Ringan	0.554	3
Hujan Sedang	0.279	7
Mendung	0.900	1
Kabut	0.464	4
Hujan Deras	0.148	8
Hujan Deras + Kabut	0.326	5

Tabel 4.14 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian keempat didapatkan hasil CCI terbesar yaitu pada alternatif “Mendung”



Gambar 4. 21 Hasil Penyesuaian Cuaca Skenario Ketujuh

Pada gambar 4.21 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Mendung” yang merupakan alternatif ke-5 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.4.8 Skenario Pengujian Kedelapan

Pada pengujian skenario kedelapan, pemain mendapatkan nilai capaian pemain seperti gambar berikut.



Gambar 4. 22 Result Skenario Kedelapan

Pada gambar 4.22 memperlihatkan bahwa pemain menyelesaikan labirin dengan perolehan skors seperti gambar tersebut. Tampilan game result yang muncul pada pengujian skenario kelima menampilkan sesuai dengan yang tertera pada *script PlayerPerformance*. Pemain mendapatkan skor waktu dengan nilai 5 yang berarti pemain menyelesaikan *game* dalam waktu  $< 1$  menit, petunjuk yang ditemukan mendapatkan nilai 4 yang berarti pemain mendapatkan 4 petunjuk, HP (*Health Point*) mendapatkan nilai 1 yang berarti HP pemain yang tersisa setelah keluar dari labirin sebesar  $> 80$  HP. Data input ini kemudian akan diproses oleh sistem GA-TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain.

Tabel 4. 15 Optimasi Bobot Skenario Kedelapan

<b>Bobot</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>	<b>C6</b>	<b>C7</b>	<b>C8</b>
	0.1721	0.1368	0.0984	0.2145	0.1153	0.0937	0.0832	0.086

Tabel 4.15 merupakan hasil optimasi bobot dari proses GA yang diintegrasikan ke dalam unity dengan bahasa pemrograman C# yang dimana menunjukkan pola pembobotan yang berbeda secara signifikan untuk setiap kriteria. Hasil dari Optimasi ini akan dilanjutkan oleh TOPSIS untuk menentukan cuaca yang sesuai dengan capaian pemain tersebut.

Tabel 4. 16 CCi dan Perankingan Skenario Kedelapan

<b>ALTERNATIF</b>	<b>CCi</b>	<b>RANKING</b>
Cerah Terik	0.621	5
Cerah Berawan	0.498	8
Hujan Ringan	0.692	3
Hujan Sedang	0.745	2
Mendung	0.654	4
Kabut	0.587	6
Hujan Deras	0.543	7
Hujan Deras + Kabut	0.873	1

Tabel 4.16 merupakan hasil dari proses kombinasi GA-TOPSIS dalam menentukan cuaca yang sesuai berdasarkan input pemain. Kolom CCI menunjukkan nilai kedekatan relatif alternatif terhadap kondisi ideal yang merupakan skor akhir TOPSIS setelah bobot ditentukan oleh GA. Dalam skenario pengujian keempat didapatkan hasil CCI terbesar yaitu pada alternatif “Hujan Deras + Kabut”.



Gambar 4. 23 Hasil Penyesuaian Cuaca Skenario Kedelapan

Pada gambar 4.23 menampilkan hasil output sistem, yaitu memberikan penyesuaian cuaca “Hujan Deras + Kabut” yang merupakan alternatif ke-8 yang sudah sesuai dengan hasil dari metode GA-TOPSIS.

#### 4.5 Pengujian Validasi 10-Fold

Pengujian validasi 10-fold digunakan untuk memvalidasi kinerja, akurasi, dan keandalan (reliabilitas) model GA-TOPSIS dalam menentukan cuaca. Pengujian dilakukan dengan menggunakan *dataset* yang terdiri dari 100 skenario capaian pemain yang dicantumkan pada halaman lampiran. Kinerja model



dievaluasi menggunakan metrik Akurasi, yang dihitung berdasarkan hasil prediksi akhir GA-TOPSIS.

*Dataset* berisi 100 data skenario, dibagi menjadi 10 lipatan (*folds*), di mana setiap *fold* terdiri dari 10 data uji. *Fold* 1 berisi data nomer 1-10, *fold* 2 berisi data nomer 11-20, hingga *fold* 10 berisi data nomer 91-100.

Tabel 4. 17 Data Training dan Testing

Iterasi	Data Training	Data Testing	Jumlah Data uji
1	Fold 2,3,4,5,6,7,8,9,10	Fold 1 (Data 1-10)	10
2	Fold 1,3,4,5,6,7,8,9,10	Fold 2 (Data 11-20)	10
3	Fold 1,2,4,5,6,7,8,9,10	Fold 3 (Data 21-30)	10
4	Fold 1,2,3,5,6,7,8,9,10	Fold 4 (Data 31-40)	10
5	Fold 1,2,3,4,6,7,8,9,10	Fold 5 (Data 41-50)	10
6	Fold 1,2,3,4,5,7,8,9,10	Fold 6 (Data 51-60)	10
7	Fold 1,2,3,4,5,6,8,9,10	Fold 7 (Data 61-70)	10
8	Fold 1,2,3,4,5,6,7,9,10	Fold 8 (Data 71-80)	10
9	Fold 1,2,3,4,5,6,7,8,10	Fold 9 (Data 81-90)	10
10	Fold 1,2,3,4,5,6,7,8,9,	Fold 10 (Data 91-100)	10

Tabel 4.17 merupakan pembagian data training dan testing pada 100 dataset yang telah disiapkan. Proses pengujian 10-fold dijalankan, di mana model GA-TOPSIS dieksekusi untuk setiap data uji. Pada iterasi pertama, dalam *fold* 1 dengan data testing yang telah ditentukan mendapatkan hasil data uji yang tepat sebanyak 8 dari 10 data seperti pada tabel 4.18. Jumlah prediksi yang tepat dihitung.

Tabel 4. 18 Hasil Validasi 10 *fold*

fold	Jumlah data uji	Data uji tepat	akurasi
1	10	8	80.00%
2	10	8	80.00%
3	10	9	90.00%
4	10	8	80.00%
5	10	9	90.00%
6	10	10	100.00%
7	10	9	90.00%
8	10	10	100.00%
9	10	7	70.00%
10	10	10	100.00%

$$accuracy(\%) = \left(\frac{8}{10}\right) \times 100 = 80.00\% \quad (4.1)$$

Akurasi pada fold 1 menghasilkan nilai 80.00% yang menunjukkan dari 10 data uji dengan data uji yang tepat sebanyak 8 mendapatkan skor akurasi 80.00%. Dari 10 hasil akurasi fold pada tabel di atas, ditarik nilai statistik utama yaitu Rata-rata Akurasi (Mean) untuk mengukur kinerja keseluruhan dan stabilitas model. Rangkuman hasil kinerja disajikan pada Tabel 4.19.

Tabel 4. 19 Mean Akurasi

Total Data Uji	Akurasi Rata-rata (mean)
100	88.00%

Berdasarkan Tabel 4.19, model GA-TOPSIS yang diusulkan berhasil mencapai akurasi rata-rata (mean) sebesar 88.00%. Nilai ini didapat dari rata-rata 10 kali pengujian *cross-validation*. Angka ini menunjukkan bahwa dari 100 data skenario yang diuji, model mampu menentukan cuaca yang sesuai sebanyak 88 kali. Tingkat akurasi yang tinggi ini memvalidasi bahwa arsitektur model sudah tepat. Ini membuktikan bahwa Algoritma Genetika (GA) mampu secara efektif menjalankan perannya sebagai *optimizer*. GA terbukti berhasil menemukan himpunan bobot kriteria yang mampu mengarahkan perhitungan TOPSIS agar menghasilkan keputusan akhir.

#### 4.6 Pengujian Usability

Pengujian *System Usability Scale* (SUS) digunakan untuk menilai tingkat kemudahan penggunaan serta kenyamanan permainan “*Salah Serenity*” berdasarkan persepsi pengguna. Metode ini memakai kuesioner berisi 10 butir pertanyaan yang dijawab oleh 30 responden setelah mereka memainkan *game*

tersebut. Tujuan dilakukannya pengujian ini adalah untuk mengetahui seberapa mudah antarmuka dan mekanisme permainan dipahami, dijalankan, serta mampu memberikan pengalaman bermain yang efektif dan menyenangkan.

Tabel 4. 20 Kategori Pertanyaan SUS

No	Pertanyaan	Aspek	Jenis Pertanyaan
1	Saya berpikir bahwa saya akan sering memainkan game ini karena sangat mudah dalam penggunaannya.	Satisfaction	Positif
2	Saya merasa kontrol game ini terlalu rumit untuk saya.	Efficiency	Negatif
3	Saya merasa kontrol game ini mudah digunakan.	Efficiency	Positif
4	Saya memerlukan bantuan orang lain untuk menggunakan fitur dalam game ini.	Learnability	Negatif
5	Saya merasa fitur ini memiliki fungsi yang sudah terintegrasi dengan baik.	Efficiency	Positif
6	Saya merasa terdapat ketidakkonsistenan dalam fitur di game ini.	Memorability	Negatif
7	Saya yakin orang lain akan mudah memahami navigasi pada game ini.	Learnability	Positif
8	Saya merasa fitur dan sistem game ini membingungkan.	Error Handling	Negatif
9	Saya merasa percaya diri saat menggunakan fitur ini.	Satisfaction	Positif
10	Saya harus banyak belajar sebelum dapat memahami gameplay pada game ini.	Learnability	Negatif

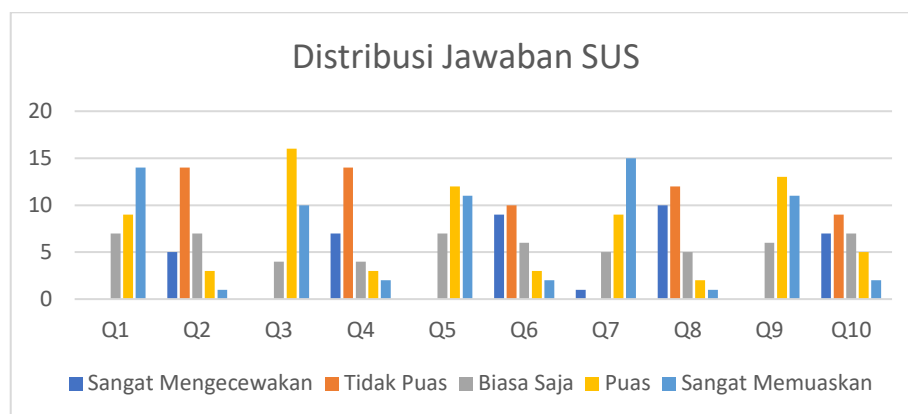
Tabel 4.13 merupakan daftar pertanyaan SUS yang telah diklasifikasikan sesuai dengan komponennya masing-masing. Evaluasi Usability pada Game dilakukan dengan menggunakan seperangkat pertanyaan yang dikelompokkan berdasarkan lima aspek utama, yaitu *Learnability* (kemudahan dipelajari), *Efficiency* (efisiensi penggunaan), *Memorability* (kemudahan diingat), *Error* (kesalahan yang mungkin terjadi), dan *Satisfaction* (kepuasan pengguna).

Tabel 4. 21 Skor Jawaban Asli SUS

No	Responden	Skor Jawaban Asli									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	R1	3	2	4	2	3	4	4	2	3	2
2	R2	5	3	5	4	4	3	5	2	4	3
3	R3	4	2	4	2	4	1	4	2	4	5
4	R4	3	2	4	1	3	3	3	3	4	1
5	R5	3	2	4	2	3	4	5	3	3	2
6	R6	3	2	3	3	3	2	3	2	3	3
7	R7	5	1	5	1	5	1	5	1	5	1
8	R8	5	1	4	1	3	5	1	1	5	5
9	R9	4	3	4	2	5	1	5	1	4	1
10	R10	5	1	5	2	4	2	5	1	4	1
11	R11	3	1	5	1	5	1	4	1	4	2
12	R12	5	4	5	5	4	5	5	4	5	4
13	R13	4	2	5	5	5	1	5	2	5	1
14	R14	4	3	4	3	4	3	4	3	4	3
15	R15	4	2	4	2	5	1	4	1	4	2
16	R16	5	1	5	1	4	2	5	1	5	1
17	R17	4	2	4	2	4	1	5	2	5	2
18	R18	5	2	5	2	5	2	5	1	5	2
19	R19	4	3	3	3	5	3	3	4	4	3
20	R20	3	4	4	2	4	2	4	2	3	2
21	R21	5	5	4	2	4	2	5	2	5	2
22	R22	5	2	4	2	4	2	4	2	3	4
23	R23	5	2	4	2	5	2	5	2	5	3
24	R24	5	4	5	4	5	4	5	5	4	4
25	R25	5	2	5	2	5	1	5	2	5	4
26	R26	5	2	4	1	5	1	5	1	5	2
27	R27	3	3	3	3	3	3	3	3	3	3
28	R28	4	3	3	2	3	2	3	2	4	3
29	R29	5	2	4	1	4	2	4	1	4	1
30	R30	4	3	4	4	4	3	4	3	4	4

Setelah seluruh data terkumpul seperti pada tabel 4.14, langkah pengolahan dilakukan mengikuti ketentuan penilaian *System Usability Scale* (SUS). Untuk pertanyaan dengan nomor ganjil, nilai yang diberikan responden dihitung dengan rumus skor jawaban – 1. Sementara itu, untuk pertanyaan bernomor genap, perhitungannya dilakukan dengan rumus 5 – skor jawaban. Pendekatan ini bertujuan menghasilkan skor akhir yang mampu menggambarkan tingkat usability antarmuka game berdasarkan persepsi serta pengalaman pengguna.

Selanjutnya, seluruh skor dari tiap responden dijumlahkan dan hasilnya dikalikan dengan faktor 2,5 sesuai metode SUS (total skor responden x 2,5) sehingga diperoleh nilai SUS untuk masing-masing individu.



Gambar 4. 24 Distribusi Jawaban SUS

Grafik pada gambar 4.24 menunjukkan distribusi jawaban untuk *System Usability Scale* (SUS) berdasarkan masing-masing pertanyaan (Q1 hingga Q10). Setiap batang mewakili jumlah responden yang memilih jawaban tertentu pada skala dari Sangat Mengecewakan hingga Sangat Memuaskan. Warna biru tua menunjukkan responden yang Sangat Mengecewakan, oranye untuk Tidak Puas, abu-abu untuk Biasa Saja, kuning untuk Puas, dan biru muda untuk Sangat Memuaskan. Dari grafik ini, kita dapat melihat bagaimana persepsi responden terhadap setiap pertanyaan terkait *usability* sistem.

Tabel 4. 22 Skor per Aspek Usability

Skor per-Aspek Usability				
No	Aspek Usability	Butir Pertanyaan SUS	Skor Rata-Rata	Kategori
1	Satisfaction	Q1 dan Q9	80.0	Excellent
2	Efficiency	Q2,Q3,Q5	74.7	Good
3	Learnability	Q4,Q7,Q10	70.0	Good
4	Memorability	Q6	67.5	Good
5	Error Handling	Q8	73.3	OK

Tabel 4.21 merupakan analisis mendalam berdasarkan data kuesioner dari 30 responden dengan memetakan 10 butir pertanyaan SUS ke dalam 5 aspek usability, yaitu *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction* yang telah dihitung seperti yang tercantum pada halaman lampiran. Pemetaan ini bertujuan untuk mengetahui sisi spesifik mana dari *game* yang sudah berjalan baik dan sisi mana yang masih memerlukan perbaikan. Aspek *Satisfaction* (Kepuasan) mendominasi perolehan nilai dengan skor tertinggi, diikuti oleh aspek *Efficiency* dan *Error Handling*. Tingginya skor pada aspek kepuasan dan penanganan kesalahan menegaskan bahwa pemain merasa percaya diri saat berinteraksi dengan sistem, serta tidak menemui hambatan atau kebingungan yang signifikan dalam menjalankan kontrol permainan.

Namun demikian, terdapat catatan evaluasi pada aspek *Learnability* dan *Memorability* yang memperoleh skor relatif lebih rendah, masing-masing sebesar 70.00 dan 67.50. Meskipun angka tersebut masih berada dalam kategori baik, hal ini menunjukkan bahwa pemain pemula memerlukan waktu adaptasi tambahan untuk memahami logika perubahan cuaca yang dinamis. Khususnya pada aspek *Memorability* yang mendapat skor terendah.

Tabel 4. 23 Skor Hasil Hitung SUS

Responden	Skor Hasil Hitung SUS										Jumlah	Nilai (Jumlah x 2.5)
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
R1	2	3	3	3	2	1	3	3	2	3	25	62.5
R2	4	2	4	1	3	2	4	3	3	2	28	70
R3	3	3	3	3	3	4	3	3	3	0	28	70
R4	2	3	3	4	2	2	2	2	3	4	27	67.5
R5	2	3	3	3	2	1	4	2	2	3	25	62.5
R6	2	3	2	2	2	3	2	3	2	2	23	57.5
R7	4	4	4	4	4	4	4	4	4	4	40	100
R8	4	4	3	4	2	0	0	4	4	0	25	62.5
R9	3	2	3	3	4	4	4	4	3	4	34	85
R10	4	4	4	3	3	3	4	4	3	4	36	90
R11	2	4	4	4	4	4	3	4	3	3	35	87.5
R12	4	1	4	0	3	0	4	1	4	1	22	55
R13	3	3	4	0	4	4	4	3	4	4	33	82.5
R14	3	2	3	2	3	2	3	2	3	2	25	62.5
R15	3	3	3	3	4	4	3	4	3	3	33	82.5
R16	4	4	4	4	3	3	4	4	4	4	38	95
R17	3	3	3	3	3	4	4	3	4	3	33	82.5
R18	4	3	4	3	4	3	4	4	4	3	36	90
R19	3	2	2	2	4	2	2	1	3	2	23	57.5
R20	2	1	3	3	3	3	3	3	2	3	26	65
R21	4	0	3	3	3	3	4	3	4	3	30	75
R22	4	3	3	3	3	3	3	3	2	1	28	70
R23	4	3	3	3	4	3	4	3	4	2	33	82.5
R24	4	1	4	1	4	1	4	0	3	1	23	57.5
R25	4	3	4	3	4	4	4	3	4	1	34	85
R26	4	3	3	4	4	4	4	4	4	3	37	92.5
R27	2	2	2	2	2	2	2	2	2	2	20	50
R28	3	2	2	3	2	3	2	3	3	2	25	62.5
R29	4	3	3	4	3	3	3	4	3	4	34	85
R30	3	2	3	1	3	2	3	2	3	1	23	57.5
Jumlah											2205	
Skor Rata-rata											73.5	

Setelah seluruh nilai SUS dari para responden diperoleh seperti pada tabel 4.15 tersebut, tahap berikutnya adalah menghitung nilai rata-rata dengan membagi total keseluruhan skor tersebut dengan jumlah responden. Adapun perhitungannya adalah sebagai berikut:

$$Rata - rataSUS = \frac{2205}{30} = 73.5$$

Dengan nilai rata-rata 73,5, sistem berada dalam kategori *Good* dan masuk pada tingkat *Acceptable* berdasarkan interpretasi tingkat kelayakan pada gambar 2.1. Ini menunjukkan bahwa *game* “*Salah Serenity*”, mampu memberikan pengalaman yang positif serta memuaskan bagi mayoritas pengguna. Visualisasi ini mendukung hasil perhitungan kuantitatif dan memperkuat kesimpulan bahwa sistem memiliki tingkat usability yang baik, sehingga layak digunakan dalam pengalaman bermain yang lebih luas.

#### 4.7 Integrasi Sains dalam Islam

Penelitian ini berfokus pada dua konsep utama dalam *muamalah*, yaitu *Muamalah Ma’a Allah* (hubungan manusia dengan Allah) dan *Muamalah Ma’a an-nas* (hubungan manusia dengan sesama). Islam sebagai agama yang komprehensif mencakup seluruh aspek kehidupan, termasuk sains dan teknologi. Dalam konteks ini, integrasi antara ilmu pengetahuan dan ajaran Islam menjadi penting sebagai upaya menghubungkan konsep ilmiah dengan nilai-nilai spiritual yang membimbing manusia menuju kesejahteraan dunia dan akhirat. Pengetahuan dalam Islam tidak semata-mata bertujuan untuk pencapaian duniawi, tetapi juga sebagai sarana mendekatkan diri kepada Allah serta memberikan manfaat bagi sesama. Oleh karena itu, penelitian ini menjadikan integrasi nilai-nilai *muamalah* sebagai kerangka dasar dalam memahami bagaimana prinsip-prinsip keagamaan dapat menginspirasi dan mengarahkan pelaksanaan penelitian secara ilmiah dan bernilai spiritual.



#### 4.7.1 *Muamalah Ma'a Allah*

Dalam penelitian ini, konsep *Muamalah Ma'a Allah* merefleksikan hubungan antara manusia dengan Allah SWT, yang menjadi dasar spiritual dalam setiap aktivitas kehidupan, termasuk dalam pengembangan karya berbasis teknologi seperti *game* edukatif. Nilai-nilai keislaman yang terkandung dalam hubungan ini menegaskan bahwa setiap usaha manusia hendaknya disandarkan pada niat yang tulus dan kesadaran akan tanggung jawab moral kepada Allah.

Penelitian ini mengangkat kisah seorang pemuda yang berusaha menuju masjid meskipun menghadapi berbagai rintangan dari pihak-pihak tertentu. Narasi tersebut merepresentasikan perjuangan spiritual seorang Muslim dalam menegakkan nilai keimanan dan keteguhan hati untuk melaksanakan ibadah. Dalam hal ini, *Muamalah Ma'a Allah* menjadi landasan maknawi yang menggambarkan ketaatan, kesabaran, dan keteguhan iman seorang hamba dalam menghadapi ujian. Seperti yang tertulis dalam Al-Quran surah Al-Baqarah ayat 286:

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ

*“Allah tidak membebani seseorang, kecuali menurut kesanggupannya. Baginya ada sesuatu (pahala) dari (kebajikan) yang diusahakannya dan terhadapnya ada (pula) sesuatu (siksa) atas (kejahatan) yang diperbuatnya.” (QS. Al-Baqarah: 286)*

Menurut tafsir Ibnu Katsir, Surah Al-Baqarah ayat 286 menjelaskan bahwa Allah SWT Maha Adil dan tidak membebani seorang hamba melebihi batas kemampuannya. Setiap ujian, cobaan, atau tanggung jawab yang diberikan kepada manusia telah disesuaikan dengan kapasitas dan kesanggupannya untuk menanggungnya. Ayat ini juga menegaskan prinsip keseimbangan antara amal dan konsekuensinya bahwa kebaikan akan dibalas dengan pahala, sementara keburukan

akan dibalas dengan dosa sesuai perbuatannya. Dalam konteks penelitian ini, tafsir tersebut mencerminkan nilai keteguhan dan tawakal yang ditampilkan oleh tokoh utama game, di mana perjuangannya menuju masjid menjadi simbol dari upaya manusia dalam menghadapi ujian hidup dengan kesabaran, keikhlasan, dan keyakinan bahwa setiap rintangan yang dihadapi merupakan bagian dari ketentuan Allah yang diberikan sesuai kemampuan manusia (Ad-Dimasyqi, 2019).

#### 4.7.2 *Muamalah Ma'a An-nas*

Konsep *Muamalah Ma'a an-nas* menekankan hubungan antara manusia dengan manusia lainnya, yang mencakup interaksi sosial, tolong-menolong, dan penyebaran manfaat. Dalam penelitian ini, aspek *Muamalah Ma'a an-nas* diwujudkan melalui mekanisme *game* yang menampilkan objek petunjuk panduan shalat, yang berfungsi sebagai sarana edukatif untuk membimbing pemain dalam memahami tata cara ibadah. Petunjuk ini merefleksikan prinsip Islam bahwa seorang Muslim memiliki tanggung jawab untuk saling menuntun pada kebaikan dan membantu sesama agar dapat menjalankan ibadah dengan benar, sehingga interaksi dalam game menjadi simbolisasi nilai sosial dan edukatif yang bermanfaat. Seperti yang tertulis dalam Al-Quran surah Al-Ma'idah ayat 2:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَحِلُّوا شَعَائِرَ اللَّهِ وَلَا الشُّهُرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَائِدَ وَلَا آمِينَ الْبَيْتِ الْحَرَامَ يَبْتَغُونَ فَضْلًا مِّن رَّبِّهِمْ وَرِضْوَانًا وَإِذَا حَلَلْتُمْ فَاصْطَادُوا وَلَا يَجْرِمَنَّكُمْ شَنَا نُ قَوْمٍ أَن صَدُّوكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ أَن تَعْتَدُوا وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ ﴿٢﴾

"Wahai orang-orang yang beriman, janganlah kamu melanggar syiar-syiar (kesucian) Allah, jangan (melanggar kehormatan) bulan-bulan haram, jangan (mengganggu) hadyu (hewan-hewan kurban) dan qalā'id (hewan-hewan kurban yang diberi tanda), dan jangan (pula mengganggu) para pengunjung Baitulharam

*sedangkan mereka mencari karunia dan rida Tuhannya! Apabila kamu telah bertahalul (menyelesaikan ihram), berburulah (jika mau). Janganlah sekali-kali kebencian(-mu) kepada suatu kaum, karena mereka menghalang-halangi kamu dari Masjidilharam, mendorongmu berbuat melampaui batas (kepada mereka). Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya." (QS. Al-Ma'idah: 2)*

Menurut tafsir Al-Jalalayn, ayat ini menegaskan pentingnya kerja sama dan saling membantu dalam hal yang membawa kebaikan, baik dalam ibadah maupun perilaku sosial. Prinsip ini menekankan bahwa setiap bentuk interaksi manusia harus diarahkan pada manfaat bersama dan ketaatan kepada Allah, serta menjauhi perbuatan yang merugikan diri sendiri maupun orang lain. Dalam konteks game, keberadaan petunjuk panduan shalat menjadi representasi konkret dari ajaran ini, di mana pemain dibimbing untuk memahami praktik ibadah yang benar, sambil menanamkan nilai kepedulian terhadap sesama melalui penyebaran manfaat pengetahuan (As-Suyuti, 2016).

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan seluruh rangkaian penelitian yang dilaksanakan, dapat disimpulkan bahwa kombinasi metode GA dan TOPSIS berhasil diterapkan untuk menciptakan sistem cuaca dinamis dalam *game* “Salah Serenity”. GA berperan efektif dalam mengoptimasi pembobotan delapan kriteria capaian pemain secara adaptif, sementara TOPSIS berfungsi sebagai pengambil keputusan akhir untuk memilih satu dari delapan alternatif cuaca yang tersedia. Sinergi kedua metode ini menghasilkan mekanisme penentuan cuaca yang responsif, objektif, dan dinamis, menggantikan pola cuaca statis atau acak yang umum digunakan dalam game konvensional.

Hasil pengujian sistem menunjukkan kinerja yang andal dan dapat diterima. Melalui validasi model dengan *10-fold cross-validation*, sistem mencapai akurasi rata-rata 88%, yang mengonfirmasi kemampuan GA dalam mengoptimasi bobot kriteria untuk menghasilkan keputusan TOPSIS yang akurat. Sementara itu, pengujian *usability* dengan “*System Usability Scale*” (SUS) menghasilkan skor 73,5 yang termasuk dalam kategori baik (*Good*) dan tingkat penerimaan dapat diterima (*Acceptable*), menunjukkan bahwa sistem dinilai mudah digunakan dan memberikan pengalaman bermain yang positif. Dengan demikian, penelitian ini tidak hanya berhasil mengimplementasikan pendekatan hybrid GA-TOPSIS dalam

konteks adaptasi *game*, tetapi juga berkontribusi dalam menciptakan pengalaman bermain yang lebih variatif, adaptif, dan sesuai dengan pencapaian individu pemain.

## 5.2 Saran

Demi pengembangan penelitian dan penyempurnaan sistem di masa mendatang, terdapat beberapa saran yang dapat dipertimbangkan:

- 1. Optimasi Komputasi:** Disarankan untuk penelitian selanjutnya menerapkan teknik optimasi koding atau pemrosesan asinkron (*asynchronous*) agar tidak membebani *frame rate* (FPS) *game*, terutama pada perangkat *mobile* dengan spesifikasi rendah.
- 2. Variasi Parameter dan Cuaca:** Penelitian selanjutnya dapat memperluas variabel input dengan menambahkan parameter psikologis pemain atau sensor eksternal, serta menambah variasi alternatif cuaca yang lebih kompleks (misalnya badai pasir atau salju) yang memiliki dampak fisik langsung terhadap karakter (seperti kecepatan jalan melambat).
- 3. Peningkatan Aset Visual:** Fokus penelitian ini adalah pada logika algoritma. Pengembangan selanjutnya disarankan untuk meningkatkan kualitas aset visual efek cuaca (partikel hujan, *volumetric fog*, pencahayaan) agar transisi perubahan cuaca terlihat lebih halus dan realistis secara grafis.
- 4. Penerapan pada Genre Lain:** Sistem kombinasi GA-TOPSIS ini dapat diuji coba pada genre *game* selain *maze/adventure*, seperti RPG atau RTS (*Real Time Strategy*), untuk menguji fleksibilitas metode dalam menangani variabel permainan yang berbeda.

## DAFTAR PUSTAKA

- Ad-Dimasyqi, A.-I. A. F. I. I. K. (2019). Tafsir Ibnu Kasir. 3.
- As-Suyuti, I. J. A.-M. I. J. (2016). Tafsir Jalalain. 1.
- Astuti, M., Herlina, H., Ibrahim, I., Rahma, M., Salsabilah, S., & Soleha, I. J. (2023). Mengoptimalkan Penggunaan Teknologi Dalam Pendidikan Islam. *Journal Of Social Humanities an Education*, 2(3), 28–40.
- Falah, F., Siregar, R., & Aulia, R. (2024). Pemilihan Meta Sinergi Terbaik Pada Game Magic Chess Mobile Legends Menggunakan Metode Topsis Prosiding Seminar Nasional Teknologi Inovasi dan Kolaborasi Disiplin Ilmu. 1(1), 126–132.
- Firdaus, M. B., Arba, M. H., Tejawati, A., Taruk, M., Irsyad, A., & Anam, M. K. (2024). Penerapan Metode Finite State Machine Pada Mobile Game Side Scrolling “Utan Adventure.” *SemanTIK*, 10(1), 151. <https://doi.org/10.55679/semantik.v10i1.47474>
- Gunawan, R. D. (2024). Penerapan Sistem Pendukung Keputusan dalam Pemilihan Supplier dengan Metode TOPSIS. 150–159.
- Hormansyah, D. S., Ariadi Retno Tri Hayati Ririd, & Pribadi, D. T. (2018). Implementasi Fsm (Finite State Machine) Pada Game Perjuangan Pangeran Diponegoro. *Jurnal Informatika Polinema*, 4(4), 290. <https://doi.org/10.33795/jip.v4i4.222>
- Kumalasari Universitas Jember, D., Studi Sosiologi, P., Fitrotul Kamila Universitas Jember, I., Reina Salsabila Universitas Jember, N., & Artikel, I. (2024). Arus Jurnal Sosial dan Humaniora (AJSH) Peran Simulakra dalam Pembentukan Realitas: Konstruksi Dunia Permainan Roleplaying dan Dampaknya terhadap Identitas Individu INFO PENULIS. 4(1). <http://jurnal.ardenjaya.com/index.php/ajshhttp://jurnal.ardenjaya.com/index.php/ajsh>
- Mu'alimin, M., & Latipah. (2021). Sistem Pendukung Keputusan Aplikasi Pemilihan Game Android Untuk Anak Usia Dini. *JSiI (Jurnal Sistem Informasi)*, 8(1), 24–30. <https://doi.org/10.30656/jsii.v8i1.3027>
- Muningsih, E., & Kiswati, S. (2022). Penerapan Metode Technique for Order By Similarity To Ideal Solution (Topsis) Dalam Pemilihan Unit Kegiatan Mahasiswa (Ukm) Yang Diminati Di Stmik Indonesia Padang. *Bianglala Informatika*, 3(1), 229–236.
- Nakhoda, I., Soetedjo, A., & S, P. O. (2020). JASTEN Jurnal Aplikasi Sains Teknologi Nasional. *Jurnal Aplikasi Sains Teknologi Nasional*, 1(1), 7–13.
- Perwira, L. T., & Hidayat, M. (2021). Memahami Dinamika Bekerja dalam Ketidakpastian: Tinjauan Fenomenologis Pengalaman Bekerja Pengemudi Ojek Online. *Psymphathic: Jurnal Ilmiah Psikologi*, 7(2), 249–266. <https://doi.org/10.15575/psy.v7i2.7995>

- Prakasa, A. B., Nugroho, F., Faisal, M., Lestari, T. M., & N, A. N. (2025). TIN : Terapan Informatika Nusantara Penerapan Metode GA-TOPSIS untuk Sistem Seleksi Karakter Game dengan Pembobotan Dinamis Berbasis Sensor Suhu TIN : Terapan Informatika Nusantara. 6(1), 34–44. <https://doi.org/10.47065/tin.v6i1.7646>
- Pratiwi, H., & Putra, Y. S. (2023). PENERAPAN ALGORITMA GENETIKA PADA GAME “ VIRUS HUNTER ” MENGGUNAKAN CONSTRUCT 2 BERBASIS ANDROID.
- Pristanti, Y. D. (2015). Pengembangan decision support system untuk memilih jenis handphone. Jurnal STT STIKMA Internasional, 6(1), 54–66.
- Ramadhan, R. F. (2023). Implementasi dan Analisis Metode MOORA dan SMART pada Pemilihan Platform Jual Beli Online menggunakan Decision Support System. Komputika: Jurnal Sistem Komputer, 12(1), 63–71. <https://doi.org/10.34010/komputika.v12i1.9300>
- Reyhan, M., Hendrawan, A., Nugroho, F., Karami, A. F., Malik, M., State, I., Info, A., Game, I. E., Weather, D., & Achivement, P. (2019). Integrating Player Performance into Weather Adjustment Mechanisms in “ Salah Serenity ” Using the TOPSIS Approach. 7(1), 101–113. <https://doi.org/10.11591/ijeei.v7i1.xxxx>
- Ridwan, T., Yulia, R., & Heryana, N. (2024). Analisis Pengalaman Pengguna dengan Metode System Usability Scale dan User Experience Questionnaire pada Aplikasi Kampus Gratis. Nuansa Informatika, 18(2), 102–108. <https://doi.org/10.25134/ilkom.v18i2.154>
- Setiawanda, Y. B., Rasyid, M. K., Ramadhan, M. J., & Hartanto, A. D. (2019). Penerapan Algoritma Genetika dan Algoritma Ghost Framework pada Game Ms. Pacman. Creative Information Technology Journal, 5(3), 174. <https://doi.org/10.24076/citec.2018v5i3.206>
- Suprayogi, D. A., & Mahmudy, W. F. (2015). Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry. Jurnal Buana Informatika, 6(2), 121–130. <https://doi.org/10.24002/jbi.v6i2.407>
- Suseno, E. W., Ma'arif, A., & Puriyanto, R. D. (2022). Tuning Parameter Pengendali PID dengan Metode Algoritma Genetik pada Motor DC. TELKA - Telekomunikasi Elektronika Komputasi Dan Kontrol, 8(1), 1–13. <https://doi.org/10.15575/telka.v8n1.1-13>
- Tafsir Kemenag. (2022). <https://quran.kemenag.go.id/quran/per-ayat/surah/30?from=41&to=60>
- Tri, I. N., Putra, A., Agung, I. G., Wijaya, C., Saputra, I. K. D., Studi, P., Informatika, T., Teknologi, F., & Informatika, D. (2023). USABILTY testing game pubg mobile dengan metode system usability scale ( sus ). 2(2), 113–120.

- Wahono, S., & Ali, H. (2021). Peranan Data Warehouse, Software Dan Brainware Terhadap Pengambilan Keputusan (Literature Review Executive Support Sistem for Business). *Jurnal Ekonomi Manajemen Sistem Informasi*, 3(2), 225–239. <https://doi.org/10.31933/jemsi.v3i2.781>
- Wijaya, D., Haryanto, H., Astuti, E. Z., & Wijanarto, W. (2021). Algoritme Genetika untuk Desain Level Dinamis pada Game Edukasi Kebakaran Hutan. *Komputika : Jurnal Sistem Komputer*, 10(1), 69–76. <https://doi.org/10.34010/komputika.v10i1.3586>
- Wisjaya, I. D., Fajarina, N. A., & Subhi, D. H. (2019). Simulasi Pembelajaran Berternak Burung Puyuh Menggunakan Metode Finite State Machine Berbasis Android. *Jurnal Teknologi Informasi Dan Terapan*, 6(1), 12–18. <https://doi.org/10.25047/jtit.v6i1.103>
- Yulsilviana, E., & Ekawati, H. (2019). Penerapan Metode Finite State Machine (Fsm) Pada Game Agent Legenda Anak Borneo. *Sebatik*, 23(1), 116–123. <https://doi.org/10.46984/sebatik.v23i1.453>
- Yuniasih, Y., Loita, A., & Aprily, N. M. (2023). Permainan Maze dalam Meningkatkan Kemampuan Membaca pada Anak Usia Dini. *JECIE (Journal of Early Childhood and Inclusive Education)*, 7(1), 64–71. <https://doi.org/10.31537/jecie.v7i1.1205>
- Yusron, M. A. (2022). TAFAKKUR: Jurnal Ilmu Al- Qur ' an dan Tafsir INDONESIA. *Jurnal Ilmu Al-Qur'an Dan Tafsir*, 3, 189.



# LAMPIRAN

## Lampiran 1

**Tabel Dataset Pemain**

No	C1	C2	C3	C4	C5	C6	C7	C8	Target Cuaca	Cuaca Terpilih	Target Tercapai
1	4	3	4	2	4	1	3	3	Mendung	Mendung	TRUE
2	1	5	3	4	3	5	3	1	Hujan Ringan	Cerah Berawan	FALSE
3	3	1	3	5	5	5	1	2	Hujan Deras	Hujan Deras	TRUE
4	2	4	4	2	4	5	2	1	Hujan Ringan	Hujan Ringan	TRUE
5	1	2	1	1	1	2	2	3	Kabut	Kabut	TRUE
6	4	5	2	3	3	2	3	1	Hujan Sedang	Hujan Sedang	TRUE
7	4	3	4	1	1	2	4	5	Cerah Berawan	Cerah Berawan	TRUE
8	4	1	2	5	4	3	5	5	Hujan Deras	Hujan Deras	TRUE
9	5	2	2	1	1	2	1	2	Hujan Sedang	Hujan Sedang	TRUE
10	2	4	1	1	3	1	2	1	Kabut	Cerah Berawan	FALSE
11	3	1	3	3	4	1	1	5	Mendung	Mendung	TRUE
12	4	1	1	3	3	2	3	1	Hujan Sedang	Hujan Sedang	TRUE
13	1	5	3	3	1	3	3	2	Kabut	Kabut	TRUE
14	1	1	2	4	2	1	1	4	Kabut	Cerah Berawan	FALSE
15	4	2	5	1	2	3	2	5	Hujan Deras + Kabut	Cerah Berawan	FALSE
16	3	2	3	2	3	2	3	4	Kabut	Kabut	TRUE
17	3	5	1	4	2	3	5	3	Cerah Terik	Cerah Terik	TRUE
18	2	4	1	2	4	3	4	3	Kabut	Kabut	TRUE
19	4	1	3	5	3	1	2	1	Mendung	Mendung	TRUE
20	4	1	3	4	1	1	5	4	Mendung	Mendung	TRUE

No	C1	C2	C3	C4	C5	C6	C7	C8	Target Cuaca	Cuaca Terpilih	Target Tercapai
21	5	1	4	1	1	2	2	5	Mendung	Mendung	TRUE
22	3	4	5	4	5	2	3	5	Mendung	Mendung	TRUE
23	5	5	5	4	4	3	3	5	Hujan Deras + Kabut	Hujan Ringan	FALSE
24	1	3	1	2	1	3	4	5	Kabut	Kabut	TRUE
25	5	4	1	5	5	3	2	4	Hujan Deras	Hujan Deras	TRUE
26	1	5	1	1	3	1	4	1	Kabut	Kabut	TRUE
27	1	4	1	2	1	2	2	1	Kabut	Kabut	TRUE
28	5	1	1	5	4	4	3	1	Hujan Sedang	Hujan Sedang	TRUE
29	1	3	2	2	2	2	2	3	Kabut	Kabut	TRUE
30	2	2	3	1	2	2	1	3	Kabut	Kabut	TRUE
31	2	3	4	2	1	5	3	2	Kabut	Kabut	TRUE
32	1	4	2	3	1	3	5	1	Kabut	Kabut	TRUE
33	3	5	5	5	5	4	1	5	Hujan Deras	Hujan Deras	TRUE
34	5	5	4	2	5	1	2	2	Hujan Deras + Kabut	Hujan Ringan	FALSE
35	4	3	3	3	1	1	2	3	Cerah Berawan	Cerah Berawan	TRUE
36	2	3	1	2	4	5	3	5	Hujan Deras	Hujan Deras	TRUE
37	2	1	4	1	2	4	1	5	Hujan Ringan	Hujan Ringan	TRUE
38	2	2	4	1	5	4	2	3	Hujan Deras + Kabut	Hujan Ringan	FALSE
39	1	1	2	3	1	3	1	3	Kabut	Kabut	TRUE
40	2	5	3	5	4	3	1	5	Hujan Deras	Hujan Deras	TRUE

No	C1	C2	C3	C4	C5	C6	C7	C8	Target Cuaca	Cuaca Terpilih	Target Tercapai
41	4	2	2	4	1	1	1	4	Hujan Sedang	Hujan Sedang	TRUE
42	3	5	4	5	5	1	2	1	Mendung	Mendung	TRUE
43	4	4	2	4	4	3	3	1	Hujan Sedang	Hujan Sedang	TRUE
44	2	3	1	1	5	4	4	1	Kabut	Cerah Berawan	FALSE
45	4	5	2	1	5	4	5	1	Cerah Terik	Cerah Terik	TRUE
46	1	5	4	2	4	5	4	3	Hujan Ringan	Hujan Ringan	TRUE
47	4	3	5	1	1	4	3	5	Kabut	Kabut	TRUE
48	5	1	3	4	2	2	1	4	Hujan Sedang	Hujan Sedang	TRUE
49	4	3	2	2	3	2	5	3	Cerah Terik	Cerah Terik	TRUE
50	2	4	3	1	1	1	1	4	Cerah Berawan	Cerah Berawan	TRUE
51	1	5	3	5	2	5	5	2	Kabut	Kabut	TRUE
52	1	2	3	3	3	3	3	1	Hujan Ringan	Hujan Ringan	TRUE
53	4	3	2	2	3	5	1	2	Hujan Sedang	Hujan Sedang	TRUE
54	4	3	2	5	4	5	5	3	Cerah Terik	Cerah Terik	TRUE
55	3	1	4	1	5	2	3	4	Mendung	Mendung	TRUE
56	2	1	1	1	1	5	3	1	Kabut	Kabut	TRUE
57	1	4	1	5	4	2	4	1	Kabut	Kabut	TRUE
58	2	3	5	1	3	5	4	5	Hujan Ringan	Hujan Ringan	TRUE
59	1	5	5	1	5	4	5	1	Hujan Ringan	Hujan Ringan	TRUE
60	1	2	2	4	3	2	3	4	Kabut	Kabut	TRUE

No	C1	C2	C3	C4	C5	C6	C7	C8	Target Cuaca	Cuaca Terpilih	Target Tercapai
61	3	5	5	2	3	5	4	1	Cerah Terik	Cerah Terik	TRUE
62	4	5	4	2	1	5	3	4	Kabut	Kabut	TRUE
63	3	5	5	5	5	5	1	4	Hujan Deras	Hujan Deras	TRUE
64	3	5	5	5	2	4	3	1	Cerah Berawan	Cerah Berawan	TRUE
65	3	3	4	5	1	2	3	3	Cerah Berawan	Cerah Berawan	TRUE
66	1	5	5	2	4	5	1	5	Hujan Ringan	Hujan Ringan	TRUE
67	1	4	4	5	2	3	1	1	Hujan Ringan	Hujan Ringan	TRUE
68	4	1	1	2	4	4	4	2	Hujan Sedang	Hujan Sedang	TRUE
69	4	3	2	2	5	5	1	1	Hujan Deras + Kabut	Cerah Berawan	FALSE
70	5	2	2	5	4	1	5	2	Mendung	Mendung	TRUE
71	4	2	3	4	3	2	5	5	Mendung	Mendung	TRUE
72	5	1	1	2	4	4	5	3	Cerah Terik	Cerah Terik	TRUE
73	4	1	2	3	4	4	1	1	Hujan Sedang	Hujan Sedang	TRUE
74	2	4	1	3	4	4	5	5	Hujan Deras	Hujan Deras	TRUE
75	5	2	4	5	2	5	2	5	Hujan Deras	Hujan Deras	TRUE
76	1	5	3	3	5	5	5	5	Hujan Deras	Hujan Deras	TRUE
77	3	2	3	5	3	1	5	3	Mendung	Mendung	TRUE
78	2	3	5	3	4	5	1	2	Hujan Deras	Hujan Deras	TRUE
79	2	2	3	2	2	5	1	4	Hujan Deras	Hujan Deras	TRUE
80	3	5	4	2	2	2	4	2	Cerah Berawan	Cerah Berawan	TRUE

No	C1	C2	C3	C4	C5	C6	C7	C8	Target Cuaca	Cuaca Terpilih	Target Tercapai
81	1	1	3	3	1	4	3	2	Kabut	Cerah Berawan	FALSE
82	5	3	2	1	2	1	2	4	Kabut	Cerah Berawan	FALSE
83	4	3	4	1	2	1	5	4	Cerah Berawan	Cerah Berawan	TRUE
84	4	5	3	3	1	4	5	2	Cerah Terik	Cerah Terik	TRUE
85	1	5	5	1	2	1	1	1	Hujan Ringan	Hujan Ringan	TRUE
86	2	3	1	2	2	3	4	5	Kabut	Kabut	TRUE
87	1	1	4	3	3	3	5	3	Hujan Ringan	Hujan Ringan	TRUE
88	4	3	3	3	3	1	3	5	Cerah Berawan	Cerah Berawan	TRUE
89	5	5	5	4	4	4	2	5	Hujan Deras + Kabut	Cerah Berawan	FALSE
90	4	2	1	5	2	2	5	5	Cerah Terik	Cerah Terik	TRUE
91	5	1	3	5	1	3	1	1	Hujan Sedang	Hujan Sedang	TRUE
92	3	2	3	1	2	1	4	4	Kabut	Kabut	TRUE
93	2	3	4	3	3	4	3	2	Hujan Ringan	Hujan Ringan	TRUE
94	5	2	2	5	5	1	1	3	Hujan Sedang	Hujan Sedang	TRUE
95	1	1	5	4	3	4	3	5	Hujan Deras	Hujan Deras	TRUE
96	5	5	2	4	1	4	1	1	Hujan Sedang	Hujan Sedang	TRUE
97	3	4	2	4	5	3	2	4	Hujan Deras	Hujan Deras	TRUE
98	3	4	4	5	3	3	5	1	Cerah Terik	Cerah Terik	TRUE
99	2	2	1	1	5	1	2	3	Kabut	Kabut	TRUE
100	3	4	1	5	4	3	3	1	Hujan Sedang	Hujan Sedang	TRUE

## Lampiran 2

**Tabel Normalisasi Jawaban Asli SUS**

No	Responden	Normalisasi									
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	R1	2	3	3	3	2	1	3	3	2	3
2	R2	4	2	4	1	3	2	4	3	3	2
3	R3	3	3	3	3	3	4	3	3	3	0
4	R4	2	3	3	4	2	2	2	2	3	4
5	R5	2	3	3	3	2	1	4	2	2	3
6	R6	2	3	2	2	2	3	2	3	2	2
7	R7	4	4	4	4	4	4	4	4	4	4
8	R8	4	4	3	4	2	0	0	4	4	0
9	R9	3	2	3	3	4	4	4	4	3	4
10	R10	4	4	4	3	3	3	4	4	3	4
11	R11	2	4	4	4	4	4	3	4	3	3
12	R12	4	1	4	0	3	0	4	1	4	1
13	R13	3	3	4	0	4	4	4	3	4	4
14	R14	3	2	3	2	3	2	3	2	3	2
15	R15	3	3	3	3	4	4	3	4	3	3
16	R16	4	4	4	4	3	3	4	4	4	4
17	R17	3	3	3	3	3	4	4	3	4	3
18	R18	4	3	4	3	4	3	4	4	4	3
19	R19	3	2	2	2	4	2	2	1	3	2
20	R20	2	1	3	3	3	3	3	3	2	3
21	R21	4	0	3	3	3	3	4	3	4	3
22	R22	4	3	3	3	3	3	3	3	2	1
23	R23	4	3	3	3	4	3	4	3	4	2
24	R24	4	1	4	1	4	1	4	0	3	1
25	R25	4	3	4	3	4	4	4	3	4	1
26	R26	4	3	3	4	4	4	4	4	4	3
27	R27	2	2	2	2	2	2	2	2	2	2
28	R28	3	2	2	3	2	3	2	3	3	2
29	R29	4	3	3	4	3	3	3	4	3	4
30	R30	3	2	3	1	3	2	3	2	3	1

### Lampiran 3

**Tabel Skor Masing-Masing Aspek Usability**

No	Responden	Skor per-Aspek				
		Satisfaction (Q1 dan Q9)	Efficiency (Q2,Q3,Q5)	Learnability (Q4,Q7,Q10)	Memorability (Q6)	Error Handling (Q8)
1	R1	50.0	66.7	75.0	25.0	75.0
2	R2	87.5	75.0	58.3	50.0	75.0
3	R3	75.0	75.0	50.0	100.0	75.0
4	R4	62.5	66.7	83.3	50.0	50.0
5	R5	50.0	66.7	83.3	25.0	50.0
6	R6	50.0	58.3	50.0	75.0	75.0
7	R7	100.0	100.0	100.0	100.0	100.0
8	R8	100.0	75.0	33.3	0.0	100.0
9	R9	75.0	75.0	91.7	100.0	100.0
10	R10	87.5	91.7	91.7	75.0	100.0
11	R11	62.5	100.0	83.3	100.0	100.0
12	R12	100.0	66.7	41.7	0.0	25.0
13	R13	87.5	91.7	66.7	100.0	75.0
14	R14	75.0	66.7	58.3	50.0	50.0
15	R15	75.0	83.3	75.0	100.0	100.0
16	R16	100.0	91.7	100.0	75.0	100.0
17	R17	87.5	75.0	83.3	100.0	75.0
18	R18	100.0	91.7	83.3	75.0	100.0
19	R19	75.0	66.7	50.0	50.0	25.0
20	R20	50.0	58.3	75.0	75.0	75.0
21	R21	100.0	50.0	83.3	75.0	75.0
22	R22	75.0	75.0	58.3	75.0	75.0
23	R23	100.0	83.3	75.0	75.0	75.0
24	R24	87.5	75.0	50.0	25.0	0.0
25	R25	100.0	91.7	66.7	100.0	75.0
26	R26	100.0	83.3	91.7	100.0	100.0
27	R27	50.0	50.0	50.0	50.0	50.0
28	R28	75.0	50.0	58.3	75.0	75.0
29	R29	87.5	75.0	91.7	75.0	100.0
30	R30	75.0	66.7	41.7	50.0	50.0
Rata-Rata		80.0	74.7	70.0	67.5	73.3