

**PEMILIHAN SEPATU OPTIMAL PADA GAME ADAPTIF
BERDASARKAN CAPAIAN PEMAIN DENGAN
METODE FIREFLY-TOPSIS**

SKRIPSI

Oleh:
DESHINTA SETYAWAN PUTRI
NIM. 220605110034



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**PEMILIHAN SEPATU OPTIMAL PADA GAME ADAPTIF
BERDASARKAN CAPAIAN PEMAIN DENGAN
METODE *FIREFLY*-TOPSIS**

SKRIPSI

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

Oleh :
DESHINTA SETYAWAN PUTRI
220605110034

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

**PEMILIHAN SEPATU OPTIMAL PADA GAME ADAPTIF
BERDASARKAN CAPAIAN PEMAIN DENGAN
METODE *FIREFLY*-TOPSIS**

SKRIPSI

Oleh :
DESHINTA SETYAWAN PUTRI
NIM. 220605110034

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 26 Juni 2026

Pembimbing I,



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

Pembimbing II,



Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

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



Supriyanto, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

PEMILIHAN SEPATU OPTIMAL PADA GAME ADAPTIF BERDASARKAN CAPAIAN PEMAIN DENGAN METODE *FIREFLY*-TOPSIS

SKRIPSI

Oleh :
DESHINTA SETYAWAN PUTRI
NIM. 220605110034

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 26 Juni 2026

Susunan Dewan Penguji

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

Anggota Penguji I : Ahmad Fahmi Karami, M.Kom
NIP. 19870909 202012 1 001

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

Anggota Penguji III : Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

()
()
()
()

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



Supriyono, M.Kom
NIP. 19800110 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Deshinta Setyawan Putri
NIM : 220605110034
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Pemilihan Sepatu Optimal pada Game Adaptif
berdasarkan Capaian Pemain dengan Metode
Firefly-TOPSIS

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, 26 Juni 2026
Yang membuat pernyataan,

A rectangular postage stamp with a yellowish background. It features a red Garuda emblem in the center. The text "SPULUN RIBU RUPIAH" is on the left, "10000" is in the middle, and "METERAI TEMPEK" is on the right. A black signature is written over the stamp. The serial number "SD91FA0X037389011" is at the bottom.

Deshinta Setyawan Putri
NIM. 220605110063

MOTTO

“Aku dilahirkan untuk menjadi hebat.”

HALAMAN PERSEMBAHAN

Dengan rasa syukur kehadiran Allah Swt., atas limpahan nikmat, rahmat, dan karunia-Nya yang tiada terhingga, sehingga penulis senantiasa diberikan kekuatan, keteguhan, dan kesempatan untuk menyelesaikan skripsi ini. Karya ini terlahir dari harapan sebuah keluarga kecil, ditempa oleh keringat perjuangan, dipanjatkan melalui untaian doa yang tak pernah putus, dan diwujudkan dengan rasa tanggung jawab oleh penulis, Deshinta Setyawan Putri.

Penulis mempersembahkan karya ini kepada dua insan paling berharga dalam kehidupan, ayahanda Himawan Deby Damura dan ibunda Lilik Sulistyowati, yang telah mengorbankan waktu, tenaga, dan harapan demi masa depan penulis. Setiap huruf di dalamnya menjadi saksi atas cinta, pengorbanan, dan ketulusan yang diberikan. Terima kasih telah menjadi anugerah terindah yang senantiasa menguatkan langkah penulis dalam setiap perjalanan hidup.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah Swt atas segala limpahan rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan penulisan Skripsi yang berjudul “Pemilihan Sepatu Optimal pada Game Adaptif berdasarkan Capaian Pemain dengan Metode *Firefly*-TOPSIS”. Dalam penulisan skripsi ini, penulis menyadari banyak pihak yang terlibat sehingga skripsi ini dapat terselesaikan. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, CAHRM., CRMP., 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 yang telah sabar memberikan bimbingan, saran, dan dukungan yang selama proses penyusunan skripsi ini.
5. Shoffin Nahwa Utama, M.T, selaku Dosen Pembimbing 2 yang membantu menyempurnakan karya ini.
6. Hani Nurhayati, M.T., dan Ahmad Fahmi Karami, M.Kom., selaku Dosen Penguji atas kritik dan saran yang berharga selama proses ujian skripsi.

7. Seluruh jajaran Dosen dan Staff Program Studi Teknik Informatika, atas ilmu, dukungan, dan fasilitas yang telah diberikan kepada penulis selama masa studi.
8. Prof. Triyo Supriyatno, PhD., yang telah memberikan banyak ilmu, dukungan, dan kemudahan dalam proses penulis menjadi mahasiswa sekaligus bagian dari Rektorat UIN Malang
9. DEMA UIN Malang “Kartanegara”, HMPS Teknik Informatika “Encoder”, yang telah menjadi bagian dari penulis mengembangkan kemampuan leadership dan keorganisasian.
10. Komunitas Otomasi dan Robotika (Ontaki), Komunitas Web Developer (Weboender), Komunitas Data Science Enthusiast (DSE), yang telah menjadi bagian dari penulis mendapatkan pengetahuan dan pengalaman dalam dunia pemrograman.
11. Ayahanda Himawan Deby Damura, Ibunda Lilik Sulistyowati, yang tak pernah lelah untuk terus berdoa dan mengusahakan segalanya untuk penulis meskipun dengan keringat dan air mata.
12. Kak Bima, Kak Angga, Kak Irianti, Kak Shinta, Adik Rahma, yang telah menjadi saudara dan keluarga penulis, memberikan kekuatan dan dukungan kepada penulis.
13. Teman-teman grup *Let's Get Rich Girl*, grup *4 Girls "Maicarawr"*, dan grup *OTW S.Kom*, yang kebersamaan penulis selama masa studi juga proses penulisan skripsi, serta tidak lupa memberikan banyak kehangatan, semangat, motivasi, dan kebahagiaan untuk penulis.

14. Sahabat terdekat penulis, yang selalu memberikan dukungan, nasehat, dan kebaikan kepada penulis, serta menjadi tempat pulang saat penulis dalam keadaan apapun.
15. Pihak yang telah terlibat dalam penulisan skripsi ini, yang telah memberikan waktu, dukungan, dan kesabaran untuk membantu penulis menyelesaikan skripsi ini hingga akhir.
16. Seorang perempuan dengan penuh mimpi, yang telah berjuang menggapai mimpinya walau dengan tekanan, keringat, dan air mata, yang telah bertahan walau dengan banyaknya badai yang menerjang, yang telah berusaha untuk selalu menghargai walau banyaknya orang yang menghakimi. Penulis mengucapkan terima kasih untuk diri sendiri, Deshinta Setyawan Putri. Kamu terlahir untuk hal-hal besar dan hebat.

Malang, 26 Juni 2026

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR	viii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xv
ABSTRAK	xviii
ABSTRACT	xix
مستخلص البحث.....	xx
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Penelitian	6
1.4 Tujuan Penelitian.....	8
1.5 Manfaat Penelitian	8
BAB II STUDI PUSTAKA	10
2.1 Penelitian Terkait.....	10
2.2 Konsep Game Adaptif.....	15
2.3 Capaian Pemain sebagai Parameter Adaptasi.....	17
2.3.1 Indikator Kuantitatif Performa Pemain.....	18
2.3.2 Capaian Pemain terhadap <i>Gameplay</i> dan Dampak Keputusan	19
2.3.3 Faktor Capaian Pemain.....	20
2.4 Decision Support System (DSS)	21
2.5 TOPSIS.....	23
2.6 Firefly Algorithm	26
2.7 Firefly-TOPSIS dalam Game Adaptif.....	30
BAB III DESAIN DAN IMPLEMENTASI.....	33
3.1 Perancangan Game “ <i>The Night Dreamers</i> ”.....	33
3.1.1 Alur Cerita “ <i>The Night Dreamers</i> ”	33
3.1.2 Rancangan Antarmuka.....	35
3.1.3 Peran Alternatif Sepatu dalam <i>Gameplay</i>	39
3.1.4 Storyboard Game “ <i>The Night Dreamers</i> ”.....	40
3.2 Finite State Machine (FSM).....	42
3.3 Arsitektur Sistem Adaptif <i>Firefly</i> -TOPSIS.....	44
3.3.1 Blok Diagram Level 0.....	44
3.3.2 Blok Diagram Level 1.....	45
3.4 Alur Sistem Adaptif <i>Firefly</i> -TOPSIS	46
3.4.1 Mekanisme Transisi dan Perpindahan Level	48
3.5 Penentuan Kriteria Sepatu.....	50

3.6 Penentuan Alternatif Sepatu	52
3.7 Konversi Spesifikasi Teknis Sepatu ke Skor Kriteria	53
3.8 Penilaian Alternatif Sepatu terhadap Kriteria.....	56
3.9 Optimasi Bobot Kriteria.....	60
3.9.1 Representasi Solusi <i>Firefly</i>	60
3.9.2 Parameter <i>Firefly Algorithm</i>	60
3.9.3 Inisialisasi Populasi <i>Firefly</i>	61
3.9.4 Perhitungan Faktor Capaian Pemain (Fcp).....	62
3.9.5 Fungsi <i>Fitness</i>	63
3.9.6 Evaluasi <i>Firefly</i> Menggunakan TOPSIS.....	63
3.9.7 Pergerakan <i>Firefly</i>	64
3.9.8 Proses Iterasi dan Penentuan Bobot Optimal.....	64
3.10 Perhitungan Metode TOPSIS	64
3.10.1 Matriks Keputusan.....	65
3.10.2 Normalisasi Matriks Keputusan.....	65
3.10.3 Matriks Ternormalisasi Terbobot	66
3.10.4 Solusi Ideal Positif dan Negatif	67
3.10.5 Perhitungan Jarak terhadap Solusi Ideal.....	68
3.10.6 Perhitungan Nilai Preferensi.....	69
3.10.7 Pemeringkatan Alternatif dan Interpretasi dalam <i>Gameplay</i>	70
3.10.8 Keterkaitan dengan Sistem Adaptif Game.....	70
3.11 Justifikasi Pemilihan Metode	71
3.12 Skenario Pengujian Adaptabilitas	72
3.12.1 Skenario 1: Eksplorasi Aman (Pemula Berhati-hati).....	73
3.12.2 Skenario 2: Eksplorasi Agresif (Pemula Ceroboh).....	74
3.12.3 Skenario 3: Penguasaan <i>Gameplay</i> (Pemain Ahli / Master)	74
3.13 Rencana Pengujian Sistem	76
BAB IV HASIL DAN PEMBAHASAN	79
4.1 Implementasi Sistem	79
4.1.1 Implementasi Game “ <i>The Night Dreamers</i> ”	79
4.1.2 Implementasi Sistem Adaptif <i>Firefly</i> -TOPSIS	84
4.1.3 Implementasi Perpindahan Rekomendasi Antar Level	101
4.2 Hasil Pengujian Sistem Adaptabilitas	102
4.2.1 Percobaan Pertama (Eksplorasi Aman).....	103
4.2.2 Percobaan Kedua (Eksplorasi Agresif).....	107
4.2.3 Percobaan Ketiga (Penguasaan <i>Gameplay</i>).....	111
4.3 Hasil Pengujian <i>Usability</i>	115
4.4 Analisis dan Pembahasan Hasil Penelitian.....	118
4.4.1 Analisis Sintesis Temuan Sistem dan Ketergunaan (<i>Usability</i>).....	118
4.4.2 Justifikasi Keunggulan Integrasi <i>Firefly</i> -TOPSIS	119
4.5 Integrasi Penelitian dengan Nilai-Nilai Islam.....	121
4.5.1 Pemeliharaan Agama (Hifzh al-Din)	121
4.5.2 Pemeliharaan Jiwa (Hifzh al-Nafs).....	122
4.5.3 Pemeliharaan Akal (Hifzh al-‘Aql).....	122
BAB V KESIMPULAN DAN SARAN	123
5.1 Kesimpulan	123

5.2 Saran.....	124
----------------	-----

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 3. 1 Alur Gameplay	35
Gambar 3. 2 Rancangan Main Menu	36
Gambar 3. 3 Rancangan Tutorial Screen.....	37
Gambar 3. 4 Rancangan Gameplay dan Minimap	37
Gambar 3. 5 Rancangan Game Over.....	38
Gambar 3. 6 Rancangan Menu Pilihan Sepatu.....	38
Gambar 3. 7 Rancangan Level Complete.....	39
Gambar 3. 8 Diagram FSM.....	43
Gambar 3. 9 Blok Diagram Level 0	44
Gambar 3. 10 Blok Diagram Level 1	45
Gambar 3. 11 Perhitungan dan Implementasi Firefly-TOPSIS.....	46
Gambar 4. 1 Main Menu	81
Gambar 4. 2 Tutorial Screen	81
Gambar 4. 3 Gameplay dan Minimap.....	82
Gambar 4. 4 Game Over	83
Gambar 4. 5 Shoe Recommendation.....	83
Gambar 4. 6 Game Over	84
Gambar 4. 7 Penilaian Skor SUS	117

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait.....	13
Tabel 2. 2 Hubungan Capaian Pemain dengan Sistem Pengambilan Keputusan...	20
Tabel 3. 1 Story Board.....	40
Tabel 3. 2 Kriteria Sepatu Running.....	51
Tabel 3. 3 Spesifikasi Resmi Produk untuk Kriteria	51
Tabel 3. 4 Alternatif Sepatu.....	52
Tabel 3. 5 Konversi Skor C1	54
Tabel 3. 6 Konversi Skor C2	54
Tabel 3. 7 Konversi Skor C3	54
Tabel 3. 8 Konversi Skor C4	55
Tabel 3. 9 Konversi Skor C5	55
Tabel 3. 10 Konversi Skor C6	55
Tabel 3. 11 Penilaian Alternatif Sepatu.....	59
Tabel 3. 12 Parameter Firefly Algorithm	61
Tabel 3. 13 Inisialisasi Populasi Firefly	62
Tabel 3. 14 Contoh Data Pemain.....	62
Tabel 3. 15 Contoh Normalisasi Data Pemain	62
Tabel 3. 16 Normalisasi Matriks Keputusan	66
Tabel 3. 17 Matriks Ternormasi Terbobot	67
Tabel 3. 18 Solusi Ideal Positif.....	68
Tabel 3. 19 Solusi Ideal Negatif.....	68
Tabel 3. 20 Jarak terhadap Solusi Ideal.....	69
Tabel 3. 21 Nilai Preferensi.....	69
Tabel 3. 22 Pemeringkatan Alternatif.....	70
Tabel 3. 23 Matriks Desain Skenario Pengujian Adaptabilitas Sistem	75
Tabel 3. 24 Pertanyaan System Usability Scale (SUS)	77
Tabel 3. 25 Nilai Jawaban Kuesioner Skala Likert	77
Tabel 4. 1 Nilai Input Kriteria Sepatu	102
Tabel 4. 2 Data Capaian Pemain Percobaan Pertama.....	103
Tabel 4. 3 Normalisasi Data Pemain Percobaan Pertama	103
Tabel 4. 4 Data Fcp dan Best Fitness Percobaan Pertama	104
Tabel 4. 5 Bobot Kriteria Hasil Optimasi Firefly Percobaan Pertama	104
Tabel 4. 6 Normalisasi Matriks Keputusan Percobaan Pertama	105
Tabel 4. 7 Matriks Ternormalisasi Terbobot Percobaan Pertama.....	105
Tabel 4. 8 Solusi Ideal Positif dan Negatif Percobaan Pertama	105
Tabel 4. 9 Jarak terhadap Solusi Ideal Percobaan Pertama.....	106
Tabel 4. 10 Nilai Preferensi dan Ranking Percobaan Pertama.....	106
Tabel 4. 11 Data Capaian Pemain Percobaan Kedua	107
Tabel 4. 12 Normalisasi Data Pemain Percobaan Kedua.....	107
Tabel 4. 13 Data Fcp dan Best Fitness Percobaan Kedua	108
Tabel 4. 14 Bobot Kriteria Hasil Optimasi Firefly Percobaan Kedua.....	108
Tabel 4. 15 Normalisasi Matriks Keputusan Percobaan Kedua	109

Tabel 4. 16 Matriks Ternormalisasi Terbobot Percobaan Kedua.....	109
Tabel 4. 17 Solusi Ideal Positif dan Negatif Percobaan Kedua.....	109
Tabel 4. 18 Jarak terhadap Solusi Ideal Percobaan Kedua.....	110
Tabel 4. 19 Nilai Preferensi dan Ranking Percobaan Kedua	110
Tabel 4. 20 Data Capaian Pemain Percobaan Ketiga.....	111
Tabel 4. 21 Normalisasi Pemain Percobaan Ketiga	111
Tabel 4. 22 Data Fcp dan Best Fitness Percobaan Ketiga.....	112
Tabel 4. 23 Bobot Kriteria Hasil Optimasi Firefly Percobaan Ketiga	112
Tabel 4. 24 Normalisasi Matriks Keputusan Percobaan Ketiga.....	113
Tabel 4. 25 Matriks Ternormalisasi Terbobot Percobaan Ketiga	113
Tabel 4. 26 Solusi Ideal Positif dan Negatif Percobaan Ketiga	113
Tabel 4. 27 Jarak terhadap Solusi Ideal Percobaan Ketiga	114
Tabel 4. 28 Nilai Preferensi dan Ranking Percobaan Ketiga	114
Tabel 4. 29 Data Kuisioner System Usability Scale.....	115
Tabel 4. 30 Hasil Konversi Nilai Mentah System Usability Scale.....	116
Tabel 4. 31 Skor Akhir System Usability Scale	116

DAFTAR PSEUDOCODE

Pseudocode 4. 1 Data Capaian Pemain.....	85
Pseudocode 4. 2 Inisialisasi Matriks Keputusan.....	86
Pseudocode 4. 3 Normalisasi Data Capaian Pemain (Fcp).....	87
Pseudocode 4. 4 Inisialisasi Populasi Firefly	89
Pseudocode 4. 5 Attractiveness.....	90
Pseudocode 4. 6 Pergerakan Firefly	91
Pseudocode 4. 7 Fungsi Fitness	92
Pseudocode 4. 8 Input Kriteria dan Alternatif.....	93
Pseudocode 4. 9 Normalisasi Matriks Keputusan TOPSIS.....	94
Pseudocode 4. 10 Matriks Ternormalisasi Terbobot	96
Pseudocode 4. 11 Solusi Ideal Positif dan Negatif.....	96
Pseudocode 4. 12 Jarak Euclidean	98
Pseudocode 4. 13 Nilai Preferensi.....	99
Pseudocode 4. 14 Orkestrasi Pipeline (Integrasi Firefly + TOPSIS)	100

ABSTRAK

Putri, Deshinta Setyawan. 2026. **Pemilihan Sepatu Optimal pada Game Adaptif berdasarkan Capaian Pemain dengan Metode *Firefly*-TOPSIS**. Skripsi. Program Studi 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) Shoffin Nahwa Utama, M.T.

Kata Kunci: Game Adaptif, *Firefly Algorithm*, TOPSIS, Sistem Pendukung Keputusan, *The Night Dreamers*.

Perkembangan industri game digital menuntut adanya game adaptif yang mampu menyesuaikan elemen permainan secara dinamis guna menjaga keseimbangan pengalaman bermain (gameplay balance). Penelitian ini mengimplementasikan sistem rekomendasi pemilihan item sepatu optimal pada game prototipe *The Night Dreamers* untuk menguji responsivitas keputusan sistem berbasis performa pengguna. Tujuan penelitian ini adalah membangun dan membuktikan tingkat adaptabilitas mekanis pengambilan keputusan multikriteria secara waktu nyata (real-time) mengikuti fluktuasi capaian pemain. Metode yang digunakan dalam penelitian ini adalah integrasi *Firefly Algorithm* dan TOPSIS. *Firefly Algorithm* diterapkan sebagai metode metaheuristik untuk mengoptimasi bobot kriteria secara dinamis berdasarkan kalkulasi Faktor Capaian Pemain (Fcp), sedangkan metode TOPSIS berfokus mengevaluasi dan memeringkatkan sepuluh alternatif produk sepatu berdasarkan kedekatannya dengan solusi ideal. Pengujian adaptabilitas sistem dilakukan melalui tiga skenario percobaan profil gaya bermain (Eksplorasi Aman, Eksplorasi Agresif, dan Penguasaan Gameplay) dengan melibatkan empat parameter metrik masukan: score, damage taken, play time, dan lives left. Hasil pengujian menunjukkan bahwa mekanisme adaptation engine berhasil mendiferensiasi peringkat alternatif secara kontekstual; sistem merekomendasikan sepatu Nike Structure 25 ($V_i = 0.905$) untuk tipe aman, Adidas Ultraboost 22 ($V_i = 0.988$) untuk tipe agresif, dan kembali memilih Nike Structure 25 ($V_i = 0.982$) pada tipe master. Guna memvalidasi akurasi sistem, dilakukan pengujian komparasi antara output komputasi adaptation engine dengan lembar kalkulasi manual manusia, yang menghasilkan kesesuaian akurasi mutlak sebesar 100% pada penentuan nilai preferensi dan pemeringkatan produk. Kesimpulannya, integrasi metode *Firefly*-TOPSIS terbukti efektif, valid, dan responsif dalam mewujudkan sistem pengambilan keputusan dinamis yang adaptif terhadap karakteristik pemain di dalam game.

ABSTRACT

Putri, Deshinta Setyawan. 2026. **Optimal Shoe Selection in Adaptive Games Based on Player Achievements Using the Firefly-TOPSIS Method.** Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Ir. Fresy Nugroho, M.T, IPM., ASEAN Eng. (II) Shoffin Nahwa Utama, M.T.

Keywords: *Adaptive Game, Firefly Algorithm, TOPSIS, Decision Support System, The Night Dreamers.*

The rapid advancement of the digital gaming industry demands the development of adaptive games capable of dynamically adjusting game elements to maintain gameplay balance. This research implements an optimal shoe item recommendation system within a prototype game titled *The Night Dreamers* to evaluate the responsiveness of system decisions based on player performance. The primary objective of this study is to construct and demonstrate the adaptability of a real-time multi-criteria decision-making system in response to fluctuations in player achievements. The method employed in this research is the integration of the Firefly Algorithm and TOPSIS. The Firefly Algorithm is applied as a metaheuristic method to dynamically optimize criteria weights based on the calculation of the Player Achievement Factor (Fcp), while the TOPSIS method focuses on evaluating and ranking ten alternative shoe products based on their proximity to the ideal solutions. System adaptability testing was conducted across three gameplay profile scenarios (Safe Exploration, Aggressive Exploration, and Gameplay Mastery) utilizing four input metrics: score, damage taken, play time, and lives left. The experimental results indicate that the adaptation engine successfully differentiated alternative rankings contextually; recommending Nike Structure 25 ($V_i = 0.905$) for the safe type, Adidas Ultraboost 22 ($V_i = 0.988$) for the aggressive type, and selecting Nike Structure 25 ($V_i = 0.982$) again for the master type. To validate system accuracy, a comparative test was performed between the adaptation engine's computational output and human manual calculations, yielding an absolute verification rate of 100% in preference value determination and product ranking. In conclusion, the integration of the Firefly-TOPSIS method is proven to be effective, valid, and highly responsive in establishing a dynamic decision-making framework that adapts to player characteristics in-game.

مستخلص البحث

بوتري، ديسهيتا سيتياوان. ٢٠٢٦. اختيار الحذاء الأمثل في الألعاب التكتيكية بناءً على إنجازات اللاعب باستخدام طريقة رسالة بكالوريوس. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم. Firefly-TOPSIS. الإسلامية الحكومية بالانغ. المشرفون: (١) الدكتور المهندس فريسي نوغروهو، الماجستير (٢) شوفين نحوى أوتاما. الماجستير.

The Night Dreamers. نظام دعم القرار، لعبة، TOPSIS، طريقة، Firefly الكلمات المفتاحية: اللعبة التكتيكية، خوارزمية

يتطلب التطور السريع في صناعة الألعاب الرقمية وجود ألعاب تكتيكية قادرة على تعديل عناصر اللعبة ديناميكياً للحفاظ على توازن تجربة اللعب. يطبق هذا البحث نظام توصية لاختيار عنصر الحذاء الأمثل في لعبة نموذجية بعنوان The Night Dreamers على تقييم مدى استجابة قرارات النظام بناءً على أداء المستخدم. الهدف الرئيسي من هذه الدراسة هو بناء وإثبات مدى قدرة نظام اتخاذ القرار متعدد المعايير على التكيف في الوقت الفعلي استجابةً للتقلبات في إنجازات اللاعب. الطريقة المستخدمة في كطريقة ميتاهورية Firefly تم تطبيق خوارزمية TOPSIS وطريقة Firefly Algorithm هذا البحث هي دمج خوارزمية على تقييم وترتيب TOPSIS بينما تركز طريقة (Fcp) لتحسين أوزان المعايير ديناميكياً بناءً على حساب عامل إنجاز اللاعب عشرة أحذية بديلة بناءً على قريها من الحلول المثالية. تم إجراء اختبار قدرة النظام على التكيف عبر ثلاثة سيناريوهات للمفاتيح تعريف، أسلوب اللعب (الاستكشاف الآمن، والاستكشاف العدواني، وإتقان أسلوب اللعب) باستخدام أربعة مقاييس مدخلة: النتيجة والضرر المكتسب، ووقت اللعب، والأرواح المتبقية. تشير النتائج التجريبية إلى أن محرك التكيف نجح في تمييز ترتيب البدائل سياقياً؛ Nike Structure 25 ($V_i = 0.905$) وحذاء Adidas Ultraboost 22 للنوع الآمن، واختار حذاء Nike Structure 25 ($V_i = 0.982$) للنوع العدواني، واختار حذاء ($V_i = 0.988$) مرة أخرى لنوع الماستر. وللتحقق من دقة Nike Structure 25 ($V_i = 0.982$) للنوع العدواني، مما أسفر عن معدل تحقق مطلق النظام، تم إجراء اختبار مقارنة بين المخرجات الحسابية لحرك التكيف والحسابات اليدوية البشرية، مما أسفر عن معدل تحقق مطلق فاعليته وصلاحيته Firefly-TOPSIS بنسبة ١٠٠٪ في تحديد قيمة التفضيل وترتيب المنتجات. في الختام، أثبت دمج طريقة واستجابته العالية في إنشاء إطار عمل ديناميكي لاتخاذ القرار يتكيف مع خصائص اللاعب داخل اللعبة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan kecerdasan buatan dalam beberapa dekade terakhir telah membawa perubahan signifikan pada industri game digital. Game modern tidak lagi bersifat statis, melainkan berkembang menuju sistem yang mampu menyesuaikan diri dengan karakteristik dan performa pemain secara dinamis. Konsep tersebut dikenal sebagai game adaptif, yaitu sistem permainan yang dapat menyesuaikan elemen internalnya, seperti tingkat kesulitan, mekanika permainan, maupun atribut karakter berdasarkan data capaian pemain yang diperoleh selama proses bermain secara real-time (Rahman et al., 2022).

Penerapan sistem adaptif dalam game terbukti mampu meningkatkan player engagement serta menjaga keseimbangan pengalaman bermain. Sistem yang adaptif memungkinkan permainan menyesuaikan tingkat tantangan sesuai dengan kemampuan pemain, sehingga dapat menghindari kondisi permainan yang terlalu mudah maupun terlalu sulit (Hassan et al., 2021). Dalam konteks ini, capaian pemain menjadi komponen utama yang berperan sebagai dasar pengambilan keputusan dalam proses adaptasi.

Sebagaimana firman Allah SWT dalam QS. Al-Baqarah [1]:286:

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

“ ... Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Ia mendapat pahala (dari kebajikan) yang diusahakannya dan mendapat siksa (dari kejahatan) yang dikerjakannya.” QS. Al-Baqarah [1]:286).

Firman Allah *"Allah tidak membebani seseorang melainkan sesuai dengan keianggupannya."*, Artinya, Allah tidak akan membebani seseorang di luar kemampuannya. Ini merupakan kelembutan, kasih sayang, dan kebaikan-Nya terhadap makhluk-Nya [Tafsir Ibnu Katsir Jilid 1].

Dan firman-Nya lebih lanjut, *"Ia mendapat pahala (dari kebajikan) yang diusahakannya."* Yaitu berupa kebaikan yang ia lakukan. *"Dan ia mendapat siksa (dari kejahatan) yang dikerjakannya."* Yaitu berupa keburukan yang ia perbuat [Tafsir Ibnu Katsir Jilid 1].

Merujuk pada tafsir tersebut, penelitian ini memandang bahwa pengembangan sistem adaptif merupakan bagian dari upaya pemain dalam memainkan permainan, sehingga output yang diberikan game selaras dengan apa kemampuan dan capaian pemain pada setiap permainan. Selain itu, prinsip keseimbangan juga dijelaskan dalam QS. Al-Mulk [67]: 3 – 4 yang menjadi landasan konseptual dalam merancang sistem permainan yang mampu menjaga keseimbangan antara kemampuan pemain dan tingkat tantangan yang diberikan.

الَّذِي خَلَقَ سَبْعَ سَمَاوَاتٍ طِبَاقًا مَا تَرَى فِي خَلْقِ الرَّحْمَنِ مِنْ تَفَوتٍ فَارْجِعِ الْبَصَرَ هَلْ تَرَى مِنْ فُتُورٍ
الَّذِي خَلَقَ الْمَوْتَ وَالْحَيَاةَ لِيَبْلُوَكُمْ أَيُّكُمْ أَحْسَنُ عَمَلًا وَهُوَ الْعَزِيزُ الْعَفُورُ ﴿٣﴾

"Yang telah menciptakan tujuh langit berlapis-lapis. Kamu sekali-kali tidak melihat pada ciptaan Rabb Yang Mahapemurah sesuatu yang tidak seimbang. Maka lihatlah berulang-ulang, adakah kamu lihat sesuatu yang tidak seimbang? Kemudian pandanglah sekali lagi niscaya penglihatanmu akan kembali kepadamu dengan tidak menemukan suatu cacat dan penglihatanmu itupun dalam keadaan payah...." (QS. Al-Mulk [67]: 3 – 4).

Dia berfirman, *"Yang menjadikan mati dan hidup."* Ayat ini dijadikan dalil oleh orang-orang yang berpendapat bahwa kematian adalah suatu yang wujud karena ia diciptakan (makhluk). Sedangkan makna ayat itu sendiri bahwa Allah telah mengadakan makhluk ini dari ketiadaan untuk menguji mereka, yakni untuk menguji siapakah diantara mereka yang paling baik amalnya [Tafsir Ibnu Katsir Jilid 8].

Nilai adaptivitas dalam penelitian ini juga sejalan dengan pesan perubahan dalam QS. Ar-Ra'd [13]:11:

لَهُ مُعَقِّبَاتٌ مِّنْ بَيْنِ يَدَيْهِ وَمِنْ خَلْفِهِ يَحْفَظُونَهُ ۚ مِنْ أَمْرِ اللَّهِ إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّىٰ يُعْزِرُوا مَا
بِأَنفُسِهِمْ وَإِذَا أَرَادَ اللَّهُ بِقَوْمٍ سُوءًا فَلَا مَرَدَّ لَهُ ۚ وَمَا لَهُمْ مِّنْ دُونِهِ ۚ مِنْ وَلٍ

“ ... Bagi manusia ada para Malaikat yang selalu mengikutinya bergiliran, di muka dan di belakangnya, mereka menjaganya atas perintah Allah. Sesungguhnya Allah tidak merubah keadaan suatu kaum sehingga mereka merubah keadaan yang ada pada diri mereka sendiri. Dan apabila Allah menghendaki keburukan terhadap suatu kaum, maka tak ada yang dapat menolaknya, dan sekali-kali tak ada pelindung bagi mereka selain Allah.. ... ” (QS. Ar-Ra'd [13]: 11)

"Bagi manusia ada para Malaikat 'yang selalu mengikutinya' bergiliran di muka dan di belakangnya, mereka menjaganya atas perintah Allah," maksudnya, setiap orang mempunyai Malaikat yang bergiliran menjaganya, ada penjaga pada siang hari dan ada penjaga pada malam hari, menjaga mereka dari kejahatan dan kecelakaan. Selain itu ada juga para Malaikat lain yang bergiliran mencatat perbuatannya, baik dan buruk, ada Malaikat yang bertugas malam dan ada yang bertugas siang, ada dua Malaikat di kanan dan di kiri yang mencatat amal perbuatan manusia. Yang di sebelah kanan bertugas mencatat perbuatan baik dan yang di

sebelah kiri bertugas mencatat perbuatan buruk. Masih ada dua Malaikat lain yang menjaga, satu di depan dan satu lagi di belakang [Tafsir Ibnu Katsir Jilid 4].

Ayat tersebut menegaskan bahwa perubahan hanya dapat dicapai melalui usaha dan pembaruan yang berkelanjutan. Prinsip ini selaras dengan konsep sistem adaptif dalam game, di mana keputusan diambil berdasarkan umpan balik performa pemain secara terus-menerus. Dengan demikian, penerapan metode *Firefly-TOPSIS* diharapkan mampu menghasilkan sistem game adaptif yang tidak hanya seimbang secara teknis, tetapi juga bermakna secara konseptual dan aplikatif.

Salah satu elemen permainan yang memiliki pengaruh langsung terhadap performa pemain adalah item yang digunakan oleh karakter, khususnya sepatu. Sepatu dalam game tidak hanya berfungsi sebagai elemen visual, tetapi memiliki implikasi langsung terhadap mekanika permainan, seperti kecepatan pergerakan, stabilitas, daya tahan, dan efisiensi eksplorasi. Pemilihan sepatu yang tidak sesuai dengan kondisi pemain berpotensi menimbulkan ketidakseimbangan permainan, baik dalam bentuk meningkatnya risiko kegagalan maupun menurunnya tantangan yang seharusnya dihadapi pemain. Oleh karena itu, pemilihan sepatu optimal menjadi aspek penting dalam perancangan game adaptif berbasis performa pemain.

Namun demikian, penelitian sebelumnya dalam bidang game adaptif masih didominasi oleh pendekatan penyesuaian tingkat kesulitan atau pola musuh, tanpa memperhatikan optimasi pemilihan item berbasis pengambilan keputusan multikriteria (Meshrif et al., 2023; Mahlmann et al., 2010). Selain itu, mekanisme adaptasi yang digunakan umumnya masih bersifat statis, seperti *rule-based* atau *fuzzy logic* sederhana, sehingga kurang mampu menangani banyak variabel

performa pemain secara simultan. Kondisi ini menunjukkan adanya kesenjangan antara kebutuhan sistem adaptif yang dinamis dengan pendekatan implementasi yang masih terbatas.

Untuk mengatasi permasalahan tersebut, diperlukan suatu pendekatan yang mampu melakukan pengambilan keputusan secara objektif dan adaptif berdasarkan banyak kriteria. Metode *Multi-Criteria Decision Making* (MCDM), khususnya *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), dapat digunakan untuk menentukan alternatif terbaik berdasarkan kedekatannya dengan solusi ideal. Namun, efektivitas metode TOPSIS sangat bergantung pada bobot kriteria yang digunakan, sehingga diperlukan mekanisme optimasi untuk menghasilkan bobot yang adaptif sesuai kondisi pemain.

Dalam penelitian ini, digunakan *Firefly Algorithm* sebagai metode optimasi untuk menentukan bobot kriteria secara dinamis berdasarkan capaian pemain dan kondisi lingkungan permainan. *Firefly Algorithm* merupakan algoritma metaheuristik yang memiliki kemampuan eksplorasi yang baik dalam menemukan solusi optimal pada permasalahan non-linier dan multi-variabel. Bobot yang dihasilkan dari proses optimasi tersebut kemudian digunakan dalam metode TOPSIS untuk melakukan pemeringkatan alternatif sepatu dan menentukan pilihan yang paling optimal.

Penelitian ini mengusulkan integrasi antara *Firefly Algorithm* dan metode TOPSIS dalam suatu sistem adaptif untuk menentukan pemilihan sepatu optimal pada game berbasis performa pemain. *Firefly Algorithm* digunakan untuk mengoptimalkan bobot kriteria secara adaptif, sedangkan TOPSIS digunakan untuk

menentukan alternatif sepatu terbaik berdasarkan bobot tersebut. Integrasi kedua metode ini membentuk suatu mekanisme pengambilan keputusan yang dinamis, objektif, dan responsif terhadap perubahan kondisi pemain.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem game adaptif, khususnya pada aspek pemilihan item berbasis pengambilan keputusan multikriteria. Selain itu, penelitian ini juga memperluas penerapan metode optimasi metaheuristik dan MCDM dalam domain game berbasis performa pemain, serta menjadi referensi bagi pengembangan sistem adaptif pada bidang lain yang memerlukan pengambilan keputusan dinamis.

1.2 Rumusan Masalah

Bagaimana implementasi metode integrasi *Firefly-TOPSIS* dapat membuktikan tingkat adaptabilitas rekomendasi pemilihan sepatu berdasarkan fluktuasi data capaian pemain secara real-time pada game *The Night Dreamers*?

1.3 Batasan Penelitian

Untuk menjaga fokus penelitian dan menghindari perluasan ruang lingkup yang berlebihan, penelitian ini dibatasi oleh beberapa hal berikut:

1. Penelitian diimplementasikan dalam sebuah game prototipe berjudul *The Night Dreamers* yang dirancang sebagai media pengujian sistem adaptif, bukan sebagai produk game komersial.
2. Sistem adaptif dalam penelitian ini dibatasi pada proses pemilihan item sepatu, dan tidak mencakup adaptasi pada elemen permainan lainnya seperti level, perilaku musuh, maupun alur cerita.

3. Capaian pemain yang digunakan sebagai dasar pengambilan keputusan adaptif meliputi score, damage taken, play time, dan lives left.
4. *Firefly Algorithm* digunakan secara khusus untuk mengoptimalkan bobot kriteria berdasarkan capaian pemain, dan tidak digunakan untuk menentukan alternatif sepatu secara langsung.
5. Metode TOPSIS digunakan untuk melakukan pemeringkatan dan pemilihan alternatif sepatu berdasarkan bobot kriteria yang dihasilkan oleh *Firefly Algorithm*.
6. Proses adaptasi sistem dilakukan setelah pemain menyelesaikan satu level permainan, sehingga perubahan rekomendasi sepatu diterapkan pada level berikutnya.
7. Pada level awal (level 0), pemain tidak menggunakan sepatu khusus. Level ini berfungsi sebagai tahap pengambilan data awal (*baseline*) untuk menghindari bias pada proses adaptasi.
8. Pengujian sistem difokuskan pada analisis perubahan bobot kriteria, hasil pemilihan sepatu, serta dampaknya terhadap keseimbangan *gameplay*, tanpa membahas aspek grafis atau visual permainan secara mendalam.
9. Pengujian adaptabilitas sistem dievaluasi dan divalidasi melalui tiga skenario profil pemain yang telah ditentukan, yaitu: Eksplorasi Aman (pemula berhati-hati), Eksplorasi Agresif (pemula impulsif), dan Penguasaan *Gameplay* (pemain tingkat lanjut/*master*)

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengimplementasikan dan membuktikan tingkat adaptabilitas sistem rekomendasi pemilihan sepatu menggunakan metode integrasi *Firefly Algorithm* dan TOPSIS berdasarkan fluktuasi data capaian pemain secara real-time pada game *The Night Dreamers*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi secara teoritis dalam pengembangan ilmu pengetahuan di bidang pengembangan game dan sistem cerdas, khususnya terkait penerapan metode optimasi metaheuristik dan pengambilan keputusan multikriteria (*Multi-Criteria Decision Making*). Integrasi *Firefly Algorithm* sebagai metode optimasi bobot kriteria dan metode *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) sebagai metode pemilihan alternatif diharapkan dapat memperkaya kajian mengenai sistem pengambilan keputusan adaptif berbasis performa pemain, serta menjadi referensi bagi penelitian selanjutnya dalam mengembangkan model adaptif yang menggabungkan pendekatan optimasi dan pengambilan keputusan dalam lingkungan interaktif.

Secara praktis, penelitian ini diharapkan dapat menjadi acuan dalam perancangan sistem pemilihan item yang adaptif pada game, khususnya dalam menentukan atribut karakter yang sesuai dengan kondisi dan capaian pemain. Sistem yang dikembangkan memungkinkan proses pengambilan keputusan dilakukan secara otomatis dan objektif, sehingga dapat membantu pengembang game dalam menjaga keseimbangan *gameplay* serta meningkatkan kenyamanan

dan keterlibatan pemain. Selain itu, konsep adaptasi berbasis capaian pemain juga berpotensi untuk diimplementasikan pada pengembangan game dengan mekanisme serupa maupun pada sistem interaktif lain yang memerlukan penyesuaian keputusan secara dinamis berdasarkan perilaku pengguna.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian-penelitian sebelumnya menunjukkan perkembangan signifikan dalam pemanfaatan algoritma optimasi metaheuristik dan metode pengambilan keputusan multikriteria (MCDM), khususnya dalam sistem adaptif berbasis performa (Kumar, 2024). Pendekatan seperti *Firefly Algorithm* (FA) dan *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) menjadi pilihan utama karena kemampuannya dalam menangani parameter dinamis serta menghasilkan keputusan yang objektif. Kajian terhadap penelitian terdahulu memberikan landasan teoritis dan metodologis bagi pengembangan model *Firefly-TOPSIS* pada game adaptif yang berfokus pada pemilihan sepatu optimal berdasarkan capaian pemain.

Penelitian pertama dilakukan oleh C.T. & A.K. (2025) dengan judul “*Personalized Instructional Strategy Adaptation Using TOPSIS: A Multi-Criteria Decision-Making Approach for Adaptive Learning System*”. Studi ini mengusulkan penerapan metode TOPSIS untuk menyesuaikan strategi pembelajaran berdasarkan performa peserta didik secara *real-time*. Hasil penelitian menunjukkan bahwa pendekatan *multi-criteria decision making* (MCDM) mampu meningkatkan ketepatan rekomendasi dibanding sistem berbasis aturan statis. Temuan ini menjadi acuan penting bagi penelitian saat ini, karena menunjukkan bahwa metode TOPSIS dapat digunakan untuk menilai alternatif terbaik dalam sistem adaptif berbasis data

dinamis, sebagaimana pemilihan atribut sepatu berdasarkan capaian pemain dalam game adaptif.

Penelitian kedua oleh Wang et al. (2023) yang berjudul “*A Hybrid Firefly Algorithm with Double-Level Learning Strategy*” memfokuskan pada peningkatan kemampuan eksplorasi FA melalui mekanisme pembelajaran dua tingkat. Pendekatan tersebut memungkinkan algoritma beradaptasi terhadap perubahan kondisi lingkungan pencarian sehingga mampu menemukan solusi yang lebih stabil. Dengan mengacu pada penelitian ini, model *Firefly* yang digunakan dapat dikembangkan agar mampu menyesuaikan bobot kriteria secara fleksibel sesuai data masukan yang diperoleh sistem game.

Selanjutnya, Liu et al. (2020) dalam penelitiannya yang berjudul “*Dynamic Adaptive Firefly Algorithm with Global Orientation*” mengembangkan FA adaptif dengan kemampuan menyesuaikan parameter pencarian secara otomatis selama proses iterasi. Model ini dirancang agar lebih efisien dalam menghadapi perubahan kondisi data dan mencegah konvergensi prematur. Penerapan prinsip adaptasi global ini mendukung gagasan bahwa mekanisme optimasi tidak hanya perlu cepat dan akurat, tetapi juga responsif terhadap dinamika sistem. Oleh karena itu, penelitian ini menjadi rujukan penting dalam merancang bagian optimasi bobot pada sistem *Firefly*-TOPSIS yang diusulkan.

Sementara itu, penelitian oleh Koto & Dafitri (2024) berjudul “*Optimalisasi Pemilihan Title untuk Senjata pada Game Point Blank Menggunakan Metode TOPSIS*” menerapkan TOPSIS untuk menentukan kombinasi title senjata terbaik dalam game FPS berdasarkan beberapa kriteria, seperti kecepatan, akurasi, dan

stabilitas. Hasilnya menunjukkan peningkatan performa karakter secara signifikan. Penelitian ini memiliki relevansi langsung dengan penelitian yang dilakukan karena sama-sama beroperasi pada domain game, meskipun dengan objek berbeda. Pendekatan yang digunakan Koto & Dafitri memperkuat argumentasi bahwa TOPSIS efektif dalam menentukan alternatif terbaik berdasarkan atribut permainan yang bervariasi.

Penelitian terakhir yaitu konteks integrasi antara algoritma optimasi dan metode pengambilan keputusan, Micale et al. (2019) melakukan penelitian berjudul *“Sustainable Vehicle Routing Based on Firefly Algorithm and TOPSIS Methodology.”* Penelitian ini menggabungkan FA dan TOPSIS untuk memecahkan permasalahan rute kendaraan berkelanjutan dengan mempertimbangkan efisiensi energi dan dampak lingkungan. FA digunakan untuk menentukan bobot optimal dari berbagai kriteria, sedangkan TOPSIS berfungsi memilih solusi terbaik. Integrasi ini membuktikan efektivitas kombinasi FA-TOPSIS untuk optimasi multi-kriteria, di mana sejalan dengan penelitian ini yang menggunakannya dalam konteks pemilihan sepatu optimal pada sistem adaptif.

Secara keseluruhan, dapat disimpulkan bahwa integrasi *Firefly Algorithm* dan TOPSIS memiliki potensi tinggi dalam membangun sistem adaptif yang mampu menyesuaikan keputusan secara *real-time* berdasarkan parameter performa pemain. Namun, belum banyak penelitian yang menerapkan pendekatan ini pada konteks game adaptif berbasis performa pemain, sehingga penelitian ini berusaha mengisi celah tersebut melalui rancangan sistem yang mengombinasikan *Firefly* untuk optimasi bobot dan TOPSIS untuk pemilihan alternatif sepatu optimal.

Tabel 2. 1 Penelitian Terkait

No	Peneliti & Tahun	Judul Penelitian	Metode yang Digunakan	Fokus Penelitian	Hasil Utama	Keterkaitan dengan Penelitian Ini
1	C.T. & A.K. (2025)	<i>Personalized Instructional Strategy Adaptation Using TOPSIS</i>	TOPSIS (MCDM)	Sistem pembelajaran adaptif	TOPSIS efektif untuk penentuan strategi instruksional berbasis multi-kriteria.	Menjadi dasar penerapan metode MCDM dalam sistem adaptif berbasis performa pemain.
2	Wang et al. (2023)	<i>A Novel Hybrid Firefly Algorithm with Double-Level Learning Strategy</i>	<i>Firefly</i> (double-level learning)	Peningkatan eksplorasi dan stabilitas konvergensi	Meningkatkan konvergensi dan adaptivitas pada permasalahan dinamis.	Mendukung pemilihan <i>Firefly</i> untuk optimasi bobot kriteria adaptif.
3	Liu et al. (2020)	<i>Dynamic Adaptive Firefly Algorithm with Global Orientation</i>	Adaptive <i>Firefly Algorithm</i>	Penyesuaian parameter pencarian secara dinamis	Mengurangi risiko konvergensi prematur dan meningkatkan efisiensi pencarian	Menjadi referensi teknis dalam merancang FA adaptif untuk sistem game.
4	Koto & Dafitri (2024)	Optimalisasi Pemilihan Title untuk Senjata pada <i>Game Point Blank</i> Menggunakan Metode TOPSIS	TOPSIS	Sistem pengambilan keputusan dalam game FPS	TOPSIS mampu menentukan kombinasi atribut optimal pada game	Menjadi acuan penerapan metode MCDM di lingkungan game adaptif
5	Micale et al. (2019)	<i>Sustainable Vehicle Routing Based on Firefly Algorithm and TOPSIS Methodology</i>	<i>Firefly</i> -TOPSIS	Optimasi multi-kriteria pada sistem logistik berkelanjutan	Integrasi FA-TOPSIS efektif dalam menentukan solusi optimal dari banyak variabel	Menjadi referensi langsung untuk integrasi FA-TOPSIS dalam sistem adaptif

Berdasarkan kajian terhadap lima penelitian terdahulu, terlihat bahwa penerapan kombinasi antara algoritma optimasi dan metode pengambilan keputusan multikriteria telah banyak digunakan dalam berbagai konteks, mulai dari

sistem pembelajaran adaptif hingga optimasi operasional. Penelitian oleh C.T. & A.K. (2025) serta Koto & Dafitri (2024) menunjukkan efektivitas TOPSIS dalam menghasilkan keputusan yang objektif berdasarkan sejumlah kriteria yang relevan. Namun, metode tersebut masih bersifat statis karena pembobotan kriterianya tidak memperhitungkan perubahan performansi pengguna yang dapat terjadi secara dinamis selama proses interaksi berlangsung. Dalam konteks game adaptif, hal ini menjadi keterbatasan karena kemampuan pemain dapat berubah seiring waktu, sehingga sistem seharusnya mampu menyesuaikan rekomendasi secara otomatis.

Penelitian oleh Wang et al. (2023) dan Liu et al. (2020) memperlihatkan bahwa *Firefly Algorithm* memiliki potensi besar dalam melakukan optimasi adaptif melalui mekanisme pembelajaran yang menyesuaikan intensitas pencarian solusi berdasarkan kualitas hasil sebelumnya. Kedua studi tersebut menegaskan bahwa *Firefly* mampu meningkatkan efisiensi pencarian solusi dibandingkan metode konvensional. Namun, pendekatan yang diusulkan masih berfokus pada optimasi matematis umum dan belum diterapkan dalam konteks pengambilan keputusan berbasis perilaku pengguna, seperti yang dibutuhkan dalam game adaptif yang memerlukan pembaruan bobot kriteria secara *real-time* berdasarkan capaian pemain.

Sementara itu, penelitian oleh Micale et al. (2019) berhasil mengintegrasikan *Firefly Algorithm* dengan TOPSIS untuk menyelesaikan permasalahan multi-kriteria dalam sistem logistik. Kombinasi ini terbukti efektif dalam menghasilkan keputusan yang optimal melalui sinergi antara optimasi bobot oleh *Firefly* dan proses perankingan oleh TOPSIS. Meskipun demikian, penerapan

integrasi tersebut masih terbatas pada domain industri dan belum dikembangkan dalam ranah interaktif seperti game adaptif yang memerlukan mekanisme evaluasi kinerja pemain sebagai dasar pengambilan keputusan.

Berdasarkan hasil kajian terhadap penelitian-penelitian terdahulu, dapat disimpulkan bahwa sebagian besar pendekatan dalam sistem adaptif masih berfokus pada penyesuaian tingkat kesulitan permainan atau menggunakan metode pengambilan keputusan dengan bobot yang bersifat statis. Selain itu, penerapan metode optimasi dan pengambilan keputusan multikriteria dalam konteks adaptasi berbasis pemilihan item dalam game masih terbatas. Oleh karena itu, penelitian ini mengusulkan suatu pendekatan yang mengintegrasikan *Firefly Algorithm* untuk optimasi bobot kriteria berdasarkan capaian pemain dan metode TOPSIS untuk pemilihan alternatif sepatu secara adaptif, sehingga diharapkan mampu memberikan mekanisme pengambilan keputusan yang lebih dinamis dan objektif.

2.2 Konsep Game Adaptif

Game adaptif merupakan pendekatan dalam pengembangan permainan digital yang memungkinkan sistem untuk menyesuaikan elemen permainan secara dinamis berdasarkan karakteristik dan performa pemain. Penyesuaian tersebut bertujuan untuk menciptakan pengalaman bermain yang seimbang, sehingga pemain tidak merasa permainan terlalu mudah maupun terlalu sulit. Dengan demikian, game adaptif berperan dalam meningkatkan keterlibatan (*engagement*) dan mempertahankan minat pemain selama proses bermain berlangsung.

Sistem adaptif dalam game sering dikaitkan dengan mekanisme *Dynamic Difficulty Adjustment* (DDA), yaitu metode penyesuaian tingkat kesulitan

permainan secara otomatis berdasarkan performa pemain agar pengalaman bermain tetap berada pada tingkat tantangan yang optimal (Hunicke, 2005). Konsep ini sejalan dengan teori *flow* yang dikemukakan oleh Csikszentmihalyi (1990), yang menyatakan bahwa pengalaman optimal terjadi ketika tingkat tantangan seimbang dengan kemampuan individu.

Secara umum, game adaptif memiliki dua komponen utama, yaitu pemodelan pemain (*player modeling*) dan mekanisme adaptasi. *Player modeling* merupakan proses pengumpulan dan analisis data perilaku pemain untuk membentuk representasi karakteristik atau kemampuan pemain dalam sistem (Yannakakis & Togelius, 2018). Data tersebut kemudian dianalisis untuk menentukan bentuk penyesuaian yang sesuai, seperti tingkat kesulitan, pola permainan, maupun atribut karakter. Dalam penelitian ini, capaian pemain yang digunakan meliputi *score*, *damage taken*, *play time*, *lives left*, yang merepresentasikan tingkat keberhasilan, efisiensi, serta ketahanan pemain dalam permainan.

Dalam praktiknya, adaptasi dapat dilakukan pada berbagai aspek permainan, seperti tingkat kesulitan musuh, jumlah sumber daya, tata letak level, maupun atribut item yang digunakan pemain. Sistem adaptif yang efektif harus mampu membaca indikator performa pemain secara kuantitatif dan menerjemahkannya menjadi parameter permainan yang relevan (Andrade et al., 2006). Dengan demikian, adaptasi bukan dilakukan secara acak, melainkan berbasis data yang terukur.

Sebagai alternatif, penelitian ini mengadopsi pendekatan adaptasi berbasis pemilihan item, khususnya sepatu, sebagai bagian dari mekanisme penyesuaian sistem. Pendekatan ini dipilih karena bersifat lebih modular dan tidak mengubah mekanika inti permainan secara langsung, sehingga lebih mudah diintegrasikan dalam sistem game yang sudah ada. Selain itu, pemilihan item memungkinkan sistem memberikan penyesuaian yang tetap berdampak pada performa pemain, seperti kecepatan, ketahanan, atau efisiensi pergerakan, tanpa mengurangi konsistensi *gameplay*.

Dengan demikian, penerapan sistem game adaptif dalam penelitian ini difokuskan pada pengambilan keputusan dalam pemilihan item berdasarkan capaian pemain. Pendekatan ini diharapkan mampu menghasilkan mekanisme adaptasi yang lebih fleksibel, terukur, dan tidak mengganggu struktur utama permainan, sehingga tetap menjaga keseimbangan antara tantangan dan kemampuan pemain.

2.3 Capaian Pemain sebagai Parameter Adaptasi

Dalam sistem game adaptif, performa pemain berperan sebagai sumber data utama dalam proses pengambilan keputusan. Performa pemain dapat didefinisikan sebagai tingkat keberhasilan dan efisiensi pemain dalam menyelesaikan tantangan permainan yang diukur melalui indikator kuantitatif tertentu. Data performa ini digunakan untuk merepresentasikan kemampuan aktual pemain dalam konteks permainan yang sedang berlangsung.

Sistem adaptif membutuhkan representasi formal terhadap perilaku pemain agar proses penyesuaian dapat dilakukan secara objektif dan terukur (Yannakakis

dan Togelius, 2018). Representasi tersebut umumnya diperoleh melalui metrik *gameplay* (*gameplay metrics*), yaitu variabel numerik yang menggambarkan interaksi pemain dengan sistem permainan. Metrik ini memungkinkan sistem melakukan evaluasi berbasis data, bukan berdasarkan asumsi atau aturan tetap.

Konsep metrik *gameplay* menjadi penting karena adaptasi yang efektif harus didasarkan pada indikator yang relevan terhadap tujuan permainan. Drachen et al. (2013) menjelaskan bahwa metrik dalam game dapat digunakan untuk memahami pola performa, kesulitan yang dialami pemain, serta potensi ketidakseimbangan permainan. Dengan demikian, pemilihan indikator performa harus mempertimbangkan keterkaitannya dengan mekanika inti permainan.

2.3.1 Indikator Kuantitatif Performa Pemain

Dalam penelitian ini, capaian pemain yang digunakan sebagai parameter adaptasi meliputi score, damage taken, play time, dan lives left. Pemilihan indikator tersebut didasarkan pada pertimbangan bahwa keempat variabel tersebut secara langsung merepresentasikan tingkat keberhasilan, ketahanan, efisiensi, dan daya tahan pemain dalam permainan.

1. *Score*

Score digunakan untuk mengukur tingkat keberhasilan pemain dalam menyelesaikan misi. Dalam konteks adaptasi, nilai skor dapat merepresentasikan tingkat kompetensi pemain secara keseluruhan. Skor sering digunakan dalam analisis performa karena bersifat agregatif dan mencerminkan hasil kumulatif dari berbagai tindakan pemain (Hunicke, 2005).

2. *Damage Taken*

Damage taken menunjukkan jumlah kerusakan yang diterima pemain selama permainan berlangsung. Dalam sistem adaptif, indikator ini penting karena memberikan informasi mengenai tekanan yang dialami pemain. Jika *damage* terlalu tinggi, sistem dapat mempertimbangkan penyesuaian yang membantu meningkatkan daya tahan atau mobilitas karakter.

3. *Play Time*

Play time merepresentasikan durasi yang dibutuhkan pemain untuk menyelesaikan suatu level atau misi. Menurut Drachen et al. (2013), waktu penyelesaian merupakan salah satu metrik perilaku yang efektif untuk mengidentifikasi pola kesulitan permainan. Oleh karena itu, *play time* digunakan sebagai indikator efisiensi pemain dalam menyelesaikan tantangan.

4. *Lives Left*

Lives left menggambarkan jumlah nyawa yang tersisa setelah menyelesaikan suatu level. Variabel ini relevan dalam sistem adaptif karena berkaitan langsung dengan keseimbangan permainan. Ketika pemain hampir kehabisan nyawa, sistem dapat mempertimbangkan keputusan yang membantu menjaga kelangsungan permainan.

2.3.2 Capaian Pemain terhadap *Gameplay* dan Dampak Keputusan

Untuk memperjelas hubungan antara capaian pemain dan sistem pengambilan keputusan, berikut disajikan tabel pemetaan konseptual:

Tabel 2. 2 Hubungan Capaian Pemain dengan Sistem Pengambilan Keputusan

No	Capaian	Makna <i>Gameplay</i>	Dampak terhadap Keputusan Adaptif
1	<i>Score</i>	Tingkat keberhasilan keseluruhan	Menentukan tingkat tantangan sepatu yang dipilih
2	<i>Damage Taken</i>	Tingkat kesulitan bertahan	Mengarah pada pemilihan sepatu dengan atribut pertahanan lebih tinggi
3	<i>Play Time</i>	Efisiensi penyelesaian level	Mengarah pada sepatu dengan peningkatan mobilitas
4	<i>Lives Left</i>	Stabilitas performa pemain	Menentukan kebutuhan peningkatan daya tahan

Tabel tersebut menunjukkan bahwa setiap indikator performa tidak berdiri sendiri, melainkan memiliki implikasi langsung terhadap keputusan adaptif yang diambil sistem. Melalui pendekatan ini, hubungan antara capaian pemain dan kriteria pemilihan sepatu menjadi jelas dan terstruktur. Sistem tidak melakukan penyesuaian secara acak, melainkan berdasarkan interpretasi terukur terhadap performa pemain. Dengan demikian, penggunaan keempat indikator tersebut memiliki dasar konseptual dan metodologis yang mendukung implementasi sistem adaptif berbasis *Firefly*-TOPSIS pada penelitian ini.

2.3.3 Faktor Capaian Pemain

Untuk merepresentasikan kondisi performa pemain secara menyeluruh dalam satu nilai tunggal, keempat meliputi *score*, *damage taken*, *play time*, dan *lives left*, perlu diagregasikan ke dalam suatu formulasi gabungan. Hal ini diperlukan karena algoritma *Firefly* membutuhkan nilai acuan yang merepresentasikan kualitas performa pemain secara holistik untuk digunakan dalam proses evaluasi solusi, bukan empat nilai indikator yang berdiri sendiri-sendiri.

Berdasarkan pertimbangan tersebut, penelitian ini merumuskan sebuah Faktor Capaian Pemain (Fcp) sebagai fungsi linear terbobot dari keempat indikator yang telah dinormalisasi. Pendekatan linear *weighted sum* dipilih karena sederhana,

mudah diinterpretasikan, dan umum digunakan dalam representasi performa pemain pada sistem adaptif berbasis data (Yannakakis & Togelius, 2018). Setiap indikator diberi bobot kontribusi (α) yang menunjukkan tingkat kepentingan relatifnya terhadap representasi performa secara keseluruhan. Khusus untuk indikator *damage taken*, digunakan bentuk $(1 - D)$ karena indikator ini bersifat *cost*, sehingga nilai *damage* yang rendah harus berkontribusi positif terhadap nilai Fcp. Formulasi Fcp dirumuskan sebagai berikut:

$$F_{cp} = (\alpha_1 \cdot S) + (\alpha_2 \cdot (1 - D)) + (\alpha_3 \cdot T) + (\alpha_4 \cdot L) \quad (2.1)$$

Dengan S adalah nilai *score* ternormalisasi, D adalah nilai *damage taken* ternormalisasi, T adalah nilai *play time* ternormalisasi, dan L adalah nilai *lives left* ternormalisasi, sedangkan α_1 , α_2 , α_3 , dan α_4 merupakan bobot kontribusi masing-masing indikator dengan total bobot sama dengan 1.

Nilai Fcp yang dihasilkan berada pada rentang 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan performa pemain yang lebih baik. Nilai Fcp inilah yang selanjutnya digunakan sebagai salah satu komponen dalam fungsi *fitness* pada algoritma *Firefly Algorithm*.

2.4 Decision Support System (DSS)

Sistem Pendukung Keputusan (*Decision Support System*) merupakan sistem berbasis komputer yang digunakan untuk membantu proses pengambilan keputusan dalam situasi yang bersifat semi-terstruktur maupun tidak terstruktur. DSS dirancang untuk mengolah data dan informasi menjadi alternatif keputusan yang dapat digunakan oleh pengambil keputusan dalam menentukan pilihan yang paling

sesuai (Turban et al., 2011). Dengan memanfaatkan berbagai metode analisis, DSS mampu meningkatkan kualitas keputusan melalui pendekatan yang lebih sistematis dan objektif.

Secara umum, DSS terdiri dari beberapa komponen utama, yaitu manajemen data (*data management*), manajemen model (*model management*), dan antarmuka pengguna (*user interface*) (Power, 2002). Komponen manajemen data berfungsi untuk mengelola data yang digunakan dalam proses pengambilan keputusan, sedangkan manajemen model digunakan untuk mengolah data tersebut dengan menggunakan metode atau algoritma tertentu. Sementara itu, antarmuka pengguna memungkinkan interaksi antara sistem dengan pengguna dalam proses pengambilan keputusan.

Dalam perkembangannya, DSS banyak diimplementasikan dalam berbagai bidang, termasuk dalam sistem interaktif seperti game. Pada konteks game adaptif, DSS dapat digunakan untuk menentukan keputusan yang berkaitan dengan penyesuaian elemen permainan berdasarkan kondisi pemain. Keputusan yang dihasilkan oleh sistem tidak lagi bergantung pada aturan statis, melainkan berdasarkan analisis data capaian pemain yang diperoleh selama permainan berlangsung.

Dalam penelitian ini, DSS diimplementasikan sebagai bagian dari mekanisme sistem adaptif yang berfungsi untuk menentukan pemilihan item berupa sepatu secara dinamis. Data capaian pemain digunakan sebagai input dalam sistem, kemudian diolah menggunakan metode optimasi dan pengambilan keputusan untuk menghasilkan alternatif terbaik. Dengan demikian, DSS tidak hanya berperan

sebagai alat bantu analisis, tetapi juga menjadi bagian integral dari sistem permainan yang secara langsung memengaruhi pengalaman bermain pemain

2.5 TOPSIS

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) merupakan salah satu pendekatan dalam *Multi-Criteria Decision Making* (MCDM) yang digunakan untuk menentukan alternatif terbaik berdasarkan kedekatannya dengan solusi ideal positif dan jaraknya dari solusi ideal negatif. Metode ini diperkenalkan oleh Hwang dan Yoon pada tahun 1981 dan banyak digunakan dalam berbagai permasalahan pengambilan keputusan yang melibatkan banyak kriteria (Hwang & Yoon, 1981).

Konsep dasar TOPSIS adalah bahwa alternatif terbaik merupakan alternatif yang memiliki jarak terdekat dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Solusi ideal positif merupakan kombinasi dari nilai terbaik untuk setiap kriteria, sedangkan solusi ideal negatif merupakan kombinasi dari nilai terburuk untuk setiap kriteria.

Dengan pendekatan ini, TOPSIS mampu memberikan hasil pemeringkatan alternatif secara objektif berdasarkan nilai preferensi yang dihitung. Prosedur dalam metode TOPSIS dijelaskan dengan enam langkah utama beserta perhitungan matematikanya sebagai berikut:

1. Pembentukan Matriks Keputusan

Langkah awal adalah menyusun matriks keputusan (Decision Matrix) berdasarkan sejumlah alternatif dan kriteria yang dinilai. Rumusnya ditulis sebagai berikut:

$$X = [x_{ij}]_{m \times n} \quad (2.2)$$

dimana x_{ij} adalah nilai alternatif ke-i terhadap kriteria ke-j, m adalah jumlah alternatif, dan n adalah jumlah kriteria. Setiap nilai x_{ij} bisa berupa hasil pengukuran.

2. Normalisasi Matriks Keputusan

Proses ini dilakukan untuk menyeragamkan skala antar kriteria agar tidak ada perbedaan skala pengukuran yang memengaruhi hasil keputusan.

Rumus normalisasi adalah sebagai berikut:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.3)$$

dimana r_{ij} merupakan nilai normalisasi dari alternatif ke-i terhadap kriteria ke-j. Proses ini mengubah seluruh nilai menjadi rentang 0–1 agar sebanding antar kriteria. Langkah ini menghasilkan matriks normalisasi $R = [r_{ij}]$

3. Pembentukan Matriks Ternormalisasi Berbobot

Langkah berikutnya adalah mengalikan setiap nilai normalisasi dengan bobot kriteria yang diperoleh dari hasil optimasi *Firefly Algorithm* (FA).

Rumusnya pembentukannya adalah sebagai berikut:

$$v_{ij} = w_j \times r_{ij} \quad (2.4)$$

dimana v_{ij} merupakan nilai matriks ternormalisasi berbobot, sedangkan w_j merupakan bobot untuk kriteria ke-j (hasil dari FA). Tahap ini menghasilkan

matriks V , yang menunjukkan kontribusi setiap alternatif terhadap kriteria dengan mempertimbangkan tingkat kepentingannya.

4. Menentukan Solusi Ideal Positif dan Negatif

TOPSIS menentukan dua titik referensi, yaitu solusi ideal positif A^+ (terbaik) dan solusi ideal negatif A^- (terburuk). Rumus perhitungannya sebagai berikut:

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\} \text{ dan } A^- = \{v_1^-, v_2^-, \dots, v_n^-\} \quad (2.5)$$

dengan

$$v_j^+ = \max_i v_{ij} \text{ (untuk kriteria benefit)} \quad (2.6)$$

$$v_j^- = \min_i v_{ij} \text{ (untuk kriteria cost)} \quad (2.7)$$

dimana A^+ adalah himpunan nilai terbaik untuk setiap kriteria, sedangkan A^- adalah himpunan nilai terburuk untuk setiap kriteria. Solusi ideal positif mencerminkan alternatif dengan performa tertinggi, sedangkan solusi ideal negatif menggambarkan kondisi paling tidak diinginkan.

5. Menghitung Jarak Setiap Alternatif terhadap Solusi Ideal

Langkah ini mengukur jarak Euclidean dari setiap alternatif terhadap kedua solusi (positif dan negatif). Rumus perhitungannya sebagai berikut:

$$D_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (2.8)$$

$$D_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (2.9)$$

dengan D_i^+ adalah jarak alternatif ke-i terhadap solusi ideal positif dan D_i^- adalah jarak alternatif ke-i terhadap solusi ideal negatif. Jarak ini menunjukkan seberapa dekat setiap alternatif dengan kondisi ideal dan seberapa jauh dari kondisi terburuk.

6. Menghitung Nilai Preferensi

Tahap terakhir adalah menentukan nilai preferensi (koefisien kedekatan) untuk setiap alternatif, yang menunjukkan seberapa dekat alternatif tersebut dengan solusi ideal positif. Rumus nilai preferensi sebagai berikut:

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (2.10)$$

dengan C_i sebagai nilai preferensi alternatif ke-i ($0 \leq C_i \leq 1$). Semakin tinggi nilai C_i , semakin baik alternatif tersebut. Nilai C_i inilah yang digunakan untuk menentukan peringkat akhir dari semua alternatif (misalnya, rekomendasi sepatu optimal bagi pemain).

2.6 Firefly Algorithm

Firefly Algorithm (FA) merupakan salah satu teknik metaheuristik berbasis perilaku sosial kunang-kunang, yang diperkenalkan oleh Xin-She Yang tahun 2009. Prinsip dasar algoritma ini adalah menarik solusi menuju solusi lain yang lebih terang (lebih baik) melalui mekanisme cahaya sebagai analogi kualitas solusi. Dalam literatur, FA dianggap unggul untuk masalah optimasi non-linier dan multi-dimensi karena kemampuannya dalam mengeksplorasi ruang solusi dan relatif mudah diimplementasikan. Penelitian Frontiers menyatakan bahwa FA bukan

hanya mudah diimplementasikan, tetapi juga mampu mengatasi masalah konvergensi cepat ke lokal optimum (Frontiers: Yang et al. (2022)).

Pada sistem game yang dikembangkan dalam penelitian ini, adaptasi gameplay dilakukan melalui mekanisme pemilihan item atau sepatu yang sesuai dengan kondisi pemain. Keputusan tersebut dihitung menggunakan metode TOPSIS. Namun, akurasi keputusan sangat dipengaruhi oleh bobot masing-masing kriteria. Berikut adalah langkah-langkah utama FA beserta perhitungan matematika yang dipakai di setiap tahap:

1. Inisialisasi Populasi *Firefly*

Pada tahap awal, sejumlah *firefly* diinisialisasi secara acak dalam ruang pencarian. Setiap *firefly* merepresentasikan solusi potensial dengan posisi awal x_i , di mana nilai x_i terdistribusi secara acak dalam batas minimum dan maksimum variabel. Rumus inisialisasi dinyatakan sebagai berikut:

$$x_i = x_{min} + rand(0,1) \times (x_{max} - x_{min}) \quad (2.11)$$

dimana x_i adalah posisi *firefly* ke-i (solusi ke-i), x_{min} dan x_{max} adalah batas minimum dan maksimum pencarian, serta $rand(0,1)$ adalah bilangan acak uniform antara 0 dan 1. Tahap ini bertujuan menghasilkan populasi awal dengan distribusi acak yang seragam agar algoritma tidak terjebak pada solusi lokal sejak awal proses.

2. Evaluasi Fungsi Objektif

Setiap *firefly* kemudian dievaluasi menggunakan fungsi objektif atau fungsi *fitness* untuk menentukan tingkat kecerahan (intensitas cahaya). Nilai fungsi

objektif ini menunjukkan seberapa baik solusi tersebut. Rumus untuk menentukan intensitas cahaya adalah:

$$I_i = f(x_i) \quad (2.12)$$

dimana I_i merupakan intensitas cahaya (brightness) dari *firefly* ke-I, sedangkan $f(x_i)$ nilai fungsi objektif untuk solusi x_i . Semakin tinggi nilai I_i , semakin terang *firefly* tersebut, sehingga *firefly* lain akan tertarik untuk bergerak ke arah posisinya.

3. Perhitungan Ketertarikan (*Attractiveness*)

Ketertarikan antar-*firefly* bergantung pada perbedaan intensitas cahaya dan jarak antar mereka. *Firefly* yang lebih gelap akan bergerak menuju *firefly* yang lebih terang. Daya tarik ini berkurang seiring jarak meningkat karena penyebaran cahaya melemah. Rumus ini digunakan untuk menghitung besarnya ketertarikan antara *firefly* ke-i terhadap *firefly* ke-j. Rumus perhitungan ketertarikan dinyatakan sebagai:

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2} \quad (2.13)$$

dimana β_{ij} adalah daya tarik *firefly* i terhadap *firefly* j, β_0 adalah konstanta daya tarik maksimum pada jarak nol, γ : koefisien pelemahan cahaya (light absorption coefficient), dan r_{ij} adalah jarak Euclidean antara *firefly* i dan j.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2.14)$$

Parameter γ mengontrol tingkat reduksi cahaya, di mana semakin besar nilainya maka semakin cepat intensitas cahaya berkurang seiring jarak.

4. Pergerakan *Firefly*

Jika satu *firefly* lebih gelap dibandingkan dengan lainnya, maka ia akan bergerak menuju *firefly* yang lebih terang. Pergerakan ini terjadi karena adanya kombinasi antara ketertarikan dan faktor acak, yang memungkinkan eksplorasi dan eksploitasi ruang solusi secara seimbang. Rumus pergerakan *firefly* adalah sebagai berikut:

$$x_i^{t+1} = x_i^t + \beta_{ij}(x_j^t - x_i^t) + \alpha(\text{rand} - 0.5) \quad (2.15)$$

dimana rumus ini digunakan untuk memperbarui posisi *firefly* pada iterasi berikutnya. x_i^{t+1} adalah posisi *firefly* i pada iterasi ke- $(t + 1)$, x_i^t , x_j^t adalah posisi *firefly* i dan j pada iterasi t , $\beta_{ij}(x_j^t - x_i^t)$ adalah gerakan ke arah *firefly* yang lebih terang, $\alpha(\text{rand} - 0.5)$ adalah komponen acak untuk menjaga keberagaman, serta α adalah faktor skala acak (biasanya menurun tiap iterasi untuk konvergensi). Tahapan ini memungkinkan *firefly* untuk mengeksplorasi area baru dan mencegah algoritma terjebak dalam solusi lokal.

5. Pembaruan Intensitas dan Iterasi

Setelah semua posisi *firefly* diperbarui, intensitas cahaya dihitung ulang berdasarkan fungsi objektif baru. Langkah ini dilakukan secara berulang hingga mencapai jumlah iterasi maksimum (MaxIter) atau konvergensi solusi terbaik. Kriteria berhenti dapat diformulasikan sebagai:

$$|f_{\text{best}}^{t+1} - f_{\text{best}}^t| < \varepsilon \quad (2.16)$$

dimana rumus ini digunakan untuk menentukan apakah algoritma telah konvergen, dengan f_{best}^t adalah nilai fungsi terbaik pada iterasi ke- t , sedangkan ε adalah batas toleransi perubahan nilai fungsi terbaik. Jika selisih perubahan sudah lebih kecil dari ε , maka proses dianggap stabil dan dihentikan.

6. Penentuan Solusi Terbaik

Setelah iterasi berakhir, *firefly* dengan intensitas cahaya tertinggi dianggap sebagai solusi terbaik. Dalam konteks penelitian ini, hasil tersebut merepresentasikan bobot kriteria optimal yang akan digunakan oleh metode TOPSIS untuk menentukan alternatif sepatu terbaik dalam game adaptif berbasis performa pemain.

2.7 Firefly-TOPSIS dalam Game Adaptif

Integrasi antara algoritma optimasi metaheuristik dan metode pengambilan keputusan multikriteria menjadi salah satu pendekatan yang banyak digunakan untuk mendukung sistem adaptif modern. Dalam konteks game adaptif, kombinasi *Firefly Algorithm* dan TOPSIS menawarkan mekanisme yang efisien untuk menilai dan menyesuaikan keputusan berdasarkan perubahan kondisi pemain. Pendekatan ini memanfaatkan kemampuan FA dalam melakukan pencarian solusi optimal melalui mekanisme perilaku alami kunang-kunang (*fireflies*), serta keunggulan TOPSIS dalam menentukan alternatif terbaik dari sekumpulan pilihan berdasarkan jarak terhadap solusi ideal positif dan negatif (Zhou et al., 2021). Dengan kombinasi tersebut, sistem mampu mengadaptasi keputusan secara dinamis sesuai perubahan variabel input, menjadikan game lebih responsif dan kontekstual terhadap pemain.

Secara umum, algoritma *Firefly* berfungsi sebagai mesin optimasi yang digunakan untuk menentukan bobot terbaik bagi setiap kriteria keputusan. Setiap kunang-kunang merepresentasikan solusi potensial dengan tingkat kecerahan yang sebanding dengan kualitas solusi tersebut. Proses optimasi dilakukan dengan mengarahkan kunang-kunang yang kurang terang menuju yang lebih terang menggunakan fungsi objektif tertentu (Yang, 2010). Dalam penelitian ini, fungsi objektif tersebut berkaitan dengan performa pemain, sehingga hasil optimasi *Firefly* menghasilkan bobot adaptif yang berubah secara dinamis mengikuti capaian pemain. Pendekatan ini membuat sistem lebih tanggap terhadap variasi data input dibandingkan pembobotan statis yang umumnya digunakan pada metode konvensional (Wang et al., 2023).

Setelah proses optimasi bobot selesai dilakukan oleh *Firefly*, metode TOPSIS digunakan untuk menentukan alternatif terbaik. TOPSIS bekerja dengan mendefinisikan dua titik acuan, yaitu solusi ideal positif (A^+) yang menggambarkan kondisi terbaik dan solusi ideal negatif (A^-) yang merepresentasikan kondisi terburuk. Jarak antara setiap alternatif terhadap kedua titik tersebut dihitung menggunakan metode *Euclidean*, dan alternatif dengan jarak paling dekat ke A^+ serta paling jauh dari A^- dipilih sebagai solusi optimal (Hwang & Yoon, 1981). Dalam konteks penelitian ini, alternatif yang dimaksud berupa pilihan sepatu dalam game adaptif, yang memiliki atribut berbeda seperti kecepatan, kenyamanan, dan daya tahan. Melalui TOPSIS, sistem dapat menentukan secara objektif sepatu mana

yang paling sesuai dengan capaian pemain terkini, sehingga pengalaman bermain menjadi lebih realistis dan personal.

Kombinasi *Firefly*-TOPSIS kemudian menjadi inti dari proses adaptation engine pada sistem game adaptif. Proses ini membentuk suatu loop adaptif, yang dimulai dari pengumpulan data performa pemain, dilanjutkan dengan optimasi bobot melalui *Firefly*, kemudian evaluasi alternatif oleh TOPSIS, hingga menghasilkan rekomendasi sepatu optimal secara otomatis. Dengan siklus tersebut, sistem mampu memperbarui keputusan secara terus-menerus mengikuti kondisi terbaru pemain, menjaga keseimbangan antara tingkat tantangan dan kenyamanan, serta meningkatkan immersion dalam permainan (Rahman et al., 2023).

Integrasi *Firefly*-TOPSIS juga memiliki kontribusi teoretis penting bagi pengembangan sistem pengambilan keputusan berbasis optimasi cerdas (intelligent decision support systems) di mana berhasil memadukan kecerdasan metaheuristik dengan logika multikriteria. Dengan demikian, penelitian ini tidak hanya menghadirkan solusi teknis untuk pemilihan sepatu optimal dalam game adaptif, tetapi juga memperluas penerapan konsep optimasi adaptif pada sistem interaktif yang responsif terhadap perubahan lingkungan dan perilaku pengguna secara waktu nyata.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Perancangan Game “*The Night Dreamers*”

The Night Dreamers merupakan permainan digital yang menggabungkan unsur petualangan dengan pendidikan spiritual. Game ini dirancang untuk membantu pemain sesuai target pengguna agar tidak hanya memahami, tetapi juga menghayati dan mengamalkan ibadah *Shalat Tahajud* dalam kehidupan sehari-hari. Dalam perancangan *The Night Dreamers*, ada empat komponen utama yang perlu diperhatikan, yaitu alur cerita yang logis dan koheren, mekanika permainan yang sederhana namun menantang, antarmuka pengguna yang fungsional dan nyaman, serta integrasi pencapaian pemain dengan sistem permainan yang bermakna. Setiap keberhasilan dalam game tidak hanya berdampak pada skor, tetapi juga merepresentasikan pemahaman spiritual, khususnya terkait pelaksanaan *Shalat Tahajud*. Dengan pendekatan terstruktur ini diharapkan menjadi media pembelajaran sekaligus hiburan yang efektif.

3.1.1 Alur Cerita “*The Night Dreamers*”

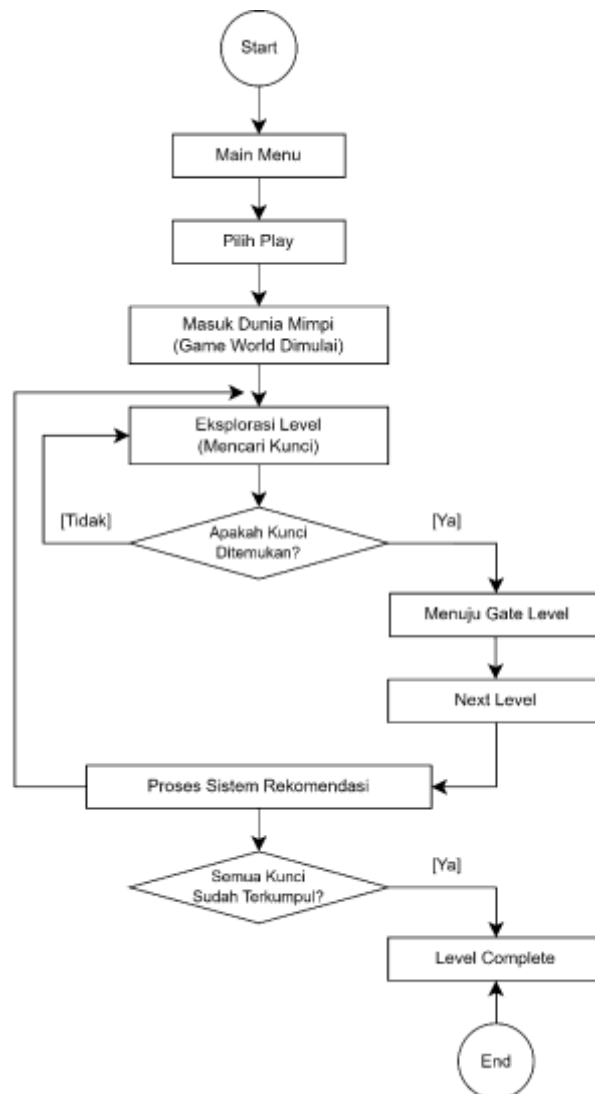
Sebagai permainan interaktif, *The Night Dreamers* dirancang dengan tujuan utama menggabungkan unsur naratif Islami dalam sebuah lingkungan permainan yang terbuka. Tokoh utama digambarkan sebagai seorang pria yang terperangkap di dalam dunia mimpi. Untuk dapat keluar dari dunia tersebut, ia harus mengumpulkan tiga kunci utama yang ada di setiap levelnya. Setiap kunci berfungsi sebagai syarat untuk membuka level berikutnya.

Pada setiap kenaikan level, tantangan menjadi lebih kompleks karena pemain harus mencari kunci dengan struktur maps yang lebih panjang dan kompleks, sehingga setiap sesi permainan memberikan pengalaman dan tantangan yang berbeda. Inti dari permainan ini adalah keberhasilan pemain dalam mengumpulkan seluruh kunci yang tersebar di setiap level. Setelah seluruh kunci berhasil diperoleh, karakter utama akan keluar dari dunia mimpi dan digambarkan melaksanakan *Shalat Tahajud* sebagai simbol kesadaran spiritual yang menjadi pesan moral utama dari permainan *The Night Dreamers*.

Dalam proses pengumpulan kunci, perjalanan pemain tidak berjalan tanpa rintangan. Dunia mimpi ini dijaga oleh entitas musuh (*Nightmares*) yang merepresentasikan gangguan spiritual. Pemain dapat menghindari entitas tersebut atau melakukan serangan balik (*attack*) untuk mempertahankan diri. Jika pemain terkena serangan, karakter akan menerima *damage* yang akan mengurangi sisa nyawa (*lives*). Damage yang didapatkan akan berbeda setiap levelnya, di mana setiap kenaikan level diimbangi dengan damage serangan musuh yang bertambah. Interaksi dengan entitas '*nightmares*' ini dirancang tidak hanya sebagai elemen tantangan *gameplay*, tetapi juga sebagai variabel input yang vital bagi sistem adaptif dalam mengevaluasi efisiensi pertahanan pemain.

Dua di antara empat parameter capaian pemain yang digunakan oleh sistem, yaitu *Damage Taken* dan *Lives Left*, secara langsung dipengaruhi oleh intensitas dan frekuensi pertemuan pemain dengan entitas ini selama satu sesi level berlangsung. Untuk memperjelas alur permainan, maka disusun blok diagram alur

gameplay yang menggambarkan urutan proses interaksi antara pemain dan sistem permainan secara terstruktur.



Gambar 3. 1 Alur Gameplay

3.1.2 Rancangan Antarmuka

Agar interaksi antara pemain dan sistem permainan dapat berlangsung secara efektif, diperlukan perancangan antarmuka (*User Interface*) yang intuitif, informatif, dan mudah dipahami. Dalam game *The Night Dreamers*, desain antarmuka tidak hanya berfungsi sebagai tampilan visual, tetapi juga sebagai media

interaksi dengan alur permainan serta sistem adaptif yang dikembangkan. Selain itu, beberapa antarmuka juga berperan sebagai titik interaksi antara pemain dan sistem rekomendasi berbasis *Firefly Algorithm* dan metode TOPSIS.

1. *Main Menu*

Halaman *Main Menu* muncul ketika permainan dijalankan di mana terdapat beberapa elemen utama seperti tombol *Play*, *Tutorial*, *Option*, dan *Quit* dengan latar belakang visual yang mencerminkan suasana permainan. Tampilan ini muncul secara otomatis saat game dijalankan dan berfungsi sebagai pusat navigasi awal sebelum pemain memasuki gameplay utama.



Gambar 3. 2 Rancangan *Main Menu*

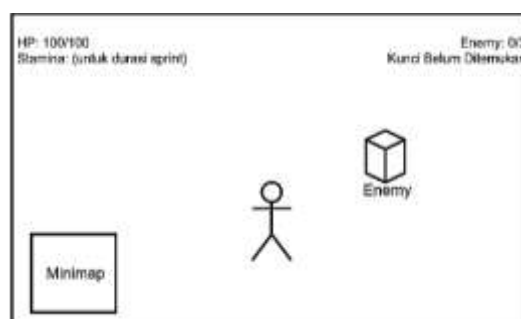
2. *Tutorial Screen*

Halaman *Tutorial* berisi panduan dasar mengenai mekanisme permainan, seperti penggunaan *mouse click left* untuk menyerang enemy, tombol W-A-S-D untuk pergerakan *player*, mouse untuk mengatur arah pandang, tombol *space* untuk pemain melompat, dan tombol *left shift* untuk pemain berlari dalam permainan. Tampilan ini muncul ketika pemain memilih menu *Tutorial* atau sebelum permainan dimulai, dengan tujuan membantu pemain memahami kontrol dasar.

Gambar 3. 3 Rancangan *Tutorial Screen*

3. *Gameplay* dan *Minimap*

Tampilan *gameplay* adalah antarmuka utama selama permainan berlangsung, di mana pemain dapat melihat lingkungan permainan serta elemen minimap yang membantu navigasi. Tampilan ini muncul setelah pemain memulai permainan dari *main menu*. Pemain eksplorasi bergerak, melompat, menyerang *nightmares*, dan mendapatkan kunci untuk ke level selanjutnya. Sistem mulai mengumpulkan data performa pemain meliputi *score*, *damage taken*, *play time*, dan *lives left* yang digunakan sebagai input dalam sistem adaptif pada tahap selanjutnya.

Gambar 3. 4 Rancangan *Gameplay* dan *Minimap*

4. *Game Over*

Game Over merupakan elemen antarmuka yang muncul ketika jumlah nyawa pemain sudah habis. Tampilan ini berupa panel yang menampilkan

siswa nyawa yang sudah habis, di mana juga menampilkan tombol restart game dan tombol back to main menu.



Gambar 3. 5 Rancangan *Game Over*

5. Menu Pilihan Sepatu Karakter

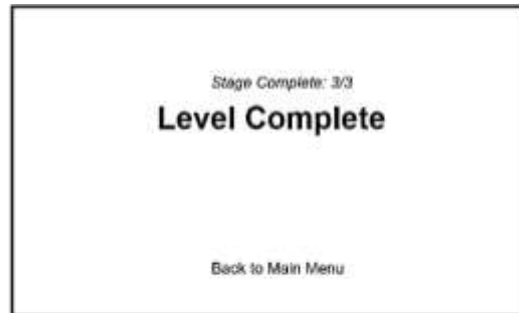
Pada tahap ini, sistem melakukan beberapa proses yang ditampilkan dalam bentuk rekomendasi sepatu yang dapat dipilih oleh pemain. Setelah pemain menyelesaikan tahap eksplorasi dalam setiap level dan sistem mengumpulkan data performa pemain, pemain menerima rekomendasi sepatu dari sistem dan dapat memilih sepatu yang telah direkomendasikan maupun sepatu yang berbeda. Pemilihan sepatu pada tahap ini akan mempengaruhi performa pemain pada level berikutnya, sehingga berperan dalam mendukung strategi permainan secara adaptif.



Gambar 3. 6 Rancangan Menu Pilihan Sepatu

6. *Level Complete*

Gambar 3.7 menampilkan layar *level complete* yang muncul ketika pemain berhasil menyelesaikan seluruh tahapan dalam satu level, ditandai dengan pengumpulan semua kunci dan melewati gerbang menuju stage berikutnya.



Gambar 3. 7 Rancangan Level Complete

3.1.3 Peran Alternatif Sepatu dalam *Gameplay*

Dalam permainan *The Night Dreamers*, alternatif sepatu memiliki peran sebagai komponen pendukung dalam sistem rekomendasi adaptif. Setiap alternatif sepatu memiliki karakteristik yang berbeda berdasarkan spesifikasi teknis, yang kemudian digunakan sebagai dasar dalam proses pengambilan keputusan menggunakan metode TOPSIS yang telah dioptimasi dengan *Firefly Algorithm*. Keberadaan alternatif sepatu memungkinkan sistem untuk memberikan rekomendasi yang sesuai dengan kondisi performa pemain selama permainan berlangsung. Rekomendasi ini bertujuan untuk membantu pemain dalam memilih atribut yang paling optimal sehingga dapat meningkatkan peluang keberhasilan dalam menyelesaikan tantangan pada setiap level.

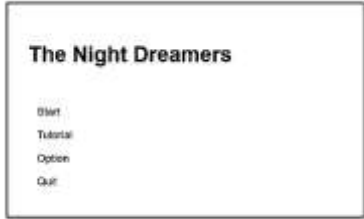
Apabila alternatif sepatu dan sistem rekomendasi ini dihilangkan, maka secara umum mekanika dasar permainan tetap dapat berjalan sebagaimana mestinya. Namun demikian, pemain tidak akan memperoleh dukungan sistem

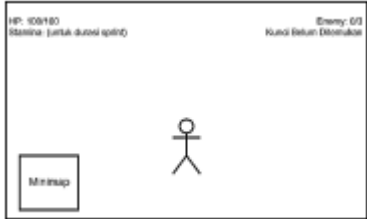
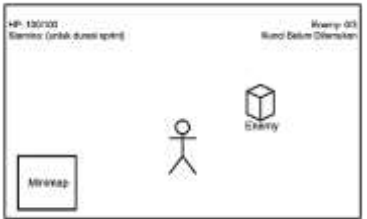
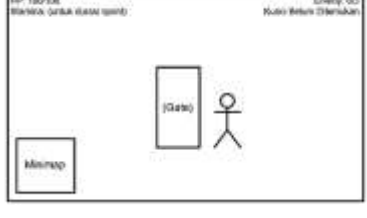
dalam menentukan pilihan atribut yang sesuai dengan kondisi permainan mereka. Hal ini berpotensi mengurangi efektivitas strategi pemain dan menurunkan peluang keberhasilan dalam menyelesaikan permainan secara optimal. Dengan demikian, keberadaan alternatif sepatu dan sistem rekomendasi adaptif memberikan nilai tambah dalam meningkatkan kualitas pengalaman bermain serta mendukung tujuan penelitian dalam mengimplementasikan sistem adaptif berbasis *Firefly*-TOPSIS.

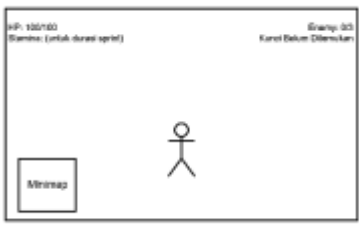
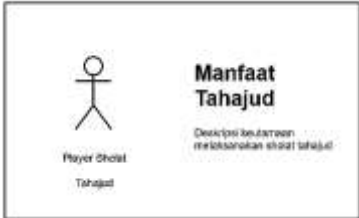
3.1.4 Storyboard Game “*The Night Dreamers*”

Dalam penelitian ini, storyboard berfungsi untuk memperjelas hubungan antara elemen cerita, mekanika permainan, serta interaksi pemain dengan sistem, sehingga alur pengalaman bermain (*player experience*) dapat dipahami secara komprehensif. Storyboard game *The Night Dreamers* disusun dalam beberapa scene utama sebagai berikut:

Tabel 3. 1 Story Board

Scene	Visual	Deskripsi	Interaksi	Output Sistem
1		Main Menu (Tampilan awal game dengan pilihan Play, Settings, dan Tutorial)	Pemain memilih menu	Sistem mengarahkan ke scene berikutnya
2		Tutorial Screen (Tampilan panduan kontrol permainan)	Mouse click, WASD, space, left shift	Pemain memahami mekanik

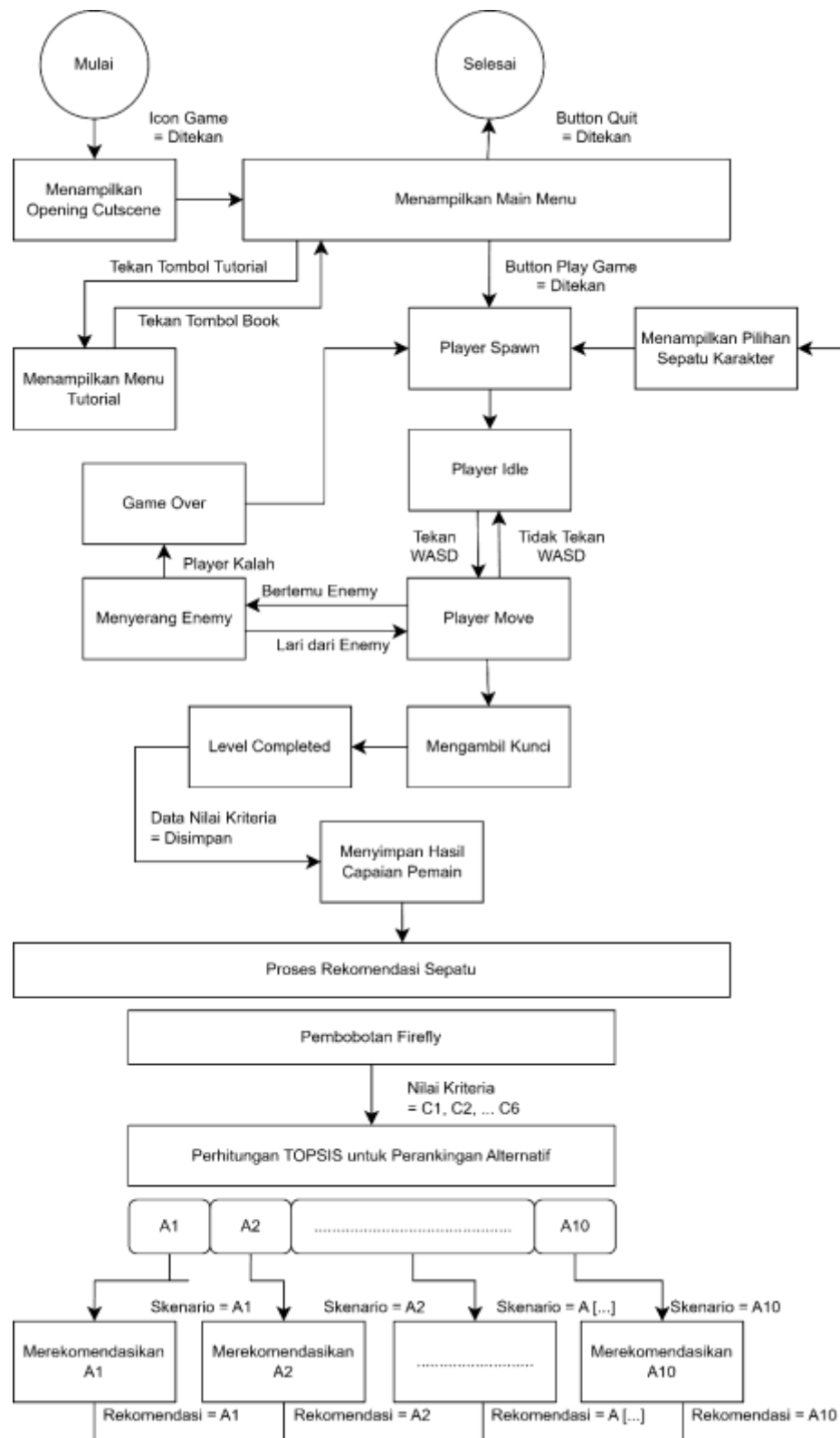
Scene	Visual	Deskripsi	Interaksi	Output Sistem
4		Eksplorasi Level (Area permainan)	Bergerak, lompat, dan interaksi dengan lingkungan	Sistem mulai mencatat skor, damage, waktu, dan sisa nyawa
4		Musuh (Nightmares) usuh yang muncul saat eksplorasi dan menyerang pemain	Menghindari atau menyerang balik (<i>attack</i>)	Sistem mencatat <i>Damage Taken</i> dan <i>Lives Left</i> sebagai input sistem adaptif
5		Mendapatkan Kunci (Objek kunci ditemukan oleh pemain)	Menghampiri kunci	Sistem memicu event pengambilan kunci
6		Game Over (Tampilan ketika nyawa pemain habis)	Pemain dapat memilih <i>button restart</i> atau <i>back to main menu</i>	Sistem menyampaikan informasi nyawa telah habis sebagai bagian dari <i>gameplay</i>
7		Transisi Level Perpindahan ke level berikutnya	Pemain masuk gate untuk melanjutkan permainan	Meningkatkan kompleksitas permainan
8		Rekomendasi Sepatu (Tampilan pemilihan sepatu dengan opsi dan rekomendasi)	Pemain memilih sepatu atau mengikuti rekomendasi sistem	Menjalankan <i>Firefly Algorithm</i> dan TOPSIS

Scene	Visual	Deskripsi	Interaksi	Output Sistem
9		Gameplay Loop (Pengulangan proses eksplorasi hingga semua kunci didapat)	Pemain terus bermain hingga menyelesaikan 3 level	Sistem terus memperbarui rekomendasi secara adaptif
10		Ending Scene (Karakter keluar dari dunia mimpi lalu Shalat Tahajud)	Pemain menyelesaikan permainan	Tujuan edukasi tercapai, menampilkan pesan moral dan akhir permainan

Tabel 3.1 menegaskan bahwa sistem rekomendasi berbasis Firefly-TOPSIS tidak berdiri sendiri, melainkan terintegrasi secara langsung dalam mekanika permainan. Dengan adanya storyboard ini, perancangan game dapat dipahami dari perspektif pengalaman pengguna (user experience), sehingga memperkuat aspek multimedia dalam penelitian ini.

3.2 Finite State Machine (FSM)

Perancangan *Finite State Machine* pada penelitian ini bertujuan untuk memvisualisasikan mekanisme kerja sistem dan menjelaskan bagaimana sistem rekomendasi pada game *The Night Dreamers* mengimplementasikan algoritma *Firefly* dan TOPSIS dalam proses pemilihan alternatif terbaik sebagai rekomendasi sepatu terbaik untuk pemain.



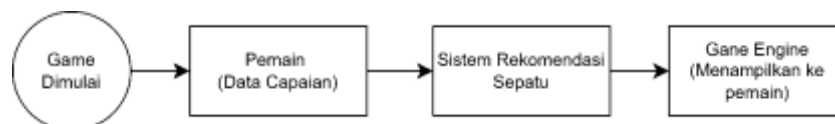
Gambar 3. 8 Diagram FSM

3.3 Arsitektur Sistem Adaptif *Firefly*-TOPSIS

Arsitektur sistem dibangun untuk menggambarkan pembagian struktur komponen internal serta aliran data yang berinteraksi di dalam lingkungan rekomendasi. Sistem adaptif dalam penelitian ini dirancang untuk menghasilkan rekomendasi sepatu berdasarkan capaian performa pemain. Arsitektur sistem menggambarkan komponen utama yang terlibat serta hubungan antar komponen dalam proses pengolahan data.

3.3.1 Blok Diagram Level 0

Pada Gambar 3.9, sistem direpresentasikan sebagai satu proses utama yang menerima input berupa data capaian pemain dari game engine dan menghasilkan output berupa rekomendasi sepatu yang dikembalikan ke sistem permainan untuk ditampilkan kepada pemain. Blok diagram level 0 menggambarkan hubungan antara sistem rekomendasi sepatu dengan entitas eksternal yang terlibat. Diagram ini menunjukkan batasan sistem serta aliran data utama yang masuk dan keluar dari sistem.



Gambar 3. 9 Blok Diagram Level 0

Dalam konteks penelitian ini, pemain berperan sebagai sumber data melalui aktivitas permainan, sedangkan game engine bertindak sebagai penghubung yang mengirimkan data capaian pemain ke sistem. Data yang masuk ke dalam sistem berupa parameter performa pemain, seperti skor, damage, waktu bermain, dan sisa nyawa. Sistem kemudian mengolah data tersebut menggunakan metode *Firefly*-

TOPSIS untuk menghasilkan rekomendasi sepatu. Hasil rekomendasi yang dihasilkan oleh sistem dikirimkan kembali ke game engine dan ditampilkan kepada pemain dalam bentuk pilihan sepatu yang sesuai dengan kondisi permainan.

3.3.2 Blok Diagram Level 1

Pada Gambar 3.10, proses sistem meliputi normalisasi data, perhitungan faktor capaian pemain (Fcp), optimasi bobot menggunakan *Firefly Algorithm*, serta pemeringkatan alternatif menggunakan metode TOPSIS hingga menghasilkan rekomendasi sepatu. Blok diagram level 1 menggambarkan proses internal yang terjadi di dalam sistem rekomendasi secara lebih rinci. Diagram ini merupakan pengembangan dari level 0 yang menunjukkan bagaimana data diproses hingga menghasilkan output.



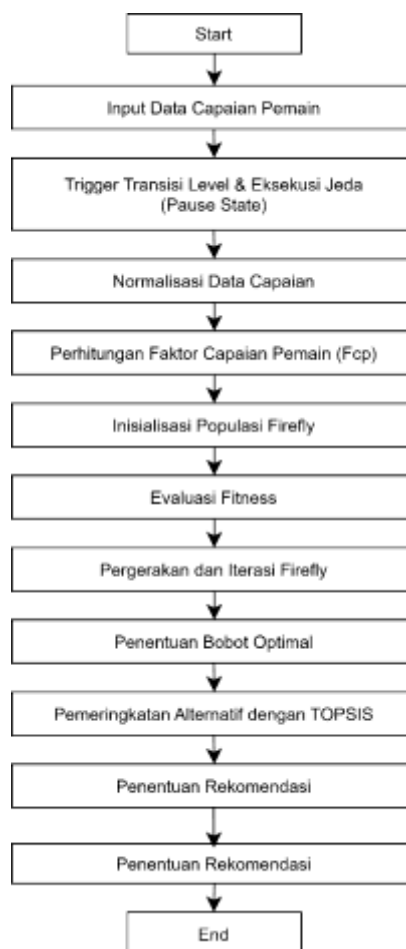
Gambar 3. 10 Blok Diagram Level 1

Pada tahap pertama, sistem menerima data capaian pemain yang kemudian dilakukan normalisasi untuk menyamakan skala antar parameter. Selanjutnya, data yang telah dinormalisasi digunakan dalam proses perhitungan faktor capaian pemain (Fcp). Kemudian optimasi bobot kriteria menggunakan *Firefly Algorithm*. Proses ini menghasilkan bobot yang adaptif berdasarkan kondisi performa pemain. Bobot yang diperoleh kemudian digunakan dalam metode TOPSIS untuk melakukan pemeringkatan alternatif sepatu. Hasil dari proses TOPSIS berupa nilai preferensi setiap alternatif yang digunakan untuk menentukan sepatu terbaik.

Alternatif dengan nilai tertinggi dipilih sebagai rekomendasi dan dikirimkan ke sistem permainan.

3.4 Alur Sistem Adaptif *Firefly*-TOPSIS

Alur sistem adaptif menggambarkan tahapan eksekusi dan urutan integrasi *Firefly Algorithm* dan TOPSIS ketika terjadi transisi level. Sistem adaptif yang dikembangkan dalam penelitian ini bekerja dengan memanfaatkan data capaian pemain sebagai input utama yang kemudian diolah melalui dua tahapan utama, yaitu optimasi bobot kriteria dan pemeringkatan alternatif. Secara umum, alur sistem dapat digambarkan sebagai berikut:



Gambar 3. 11 Perhitungan dan Implementasi *Firefly*-TOPSIS

Sistem adaptif dalam penelitian ini bekerja dengan memanfaatkan data capaian pemain sebagai input utama yang diperoleh dari game engine selama permainan berlangsung. Data tersebut meliputi skor permainan, damage yang diterima, waktu bermain, serta sisa nyawa pemain. Namun, data tersebut tidak langsung diproses begitu diperoleh. Sistem dirancang untuk menunggu momen transisi level sebelum memulai kalkulasi.

Ketika pemain berhasil menyelesaikan misi dan menyentuh gerbang keluar level, *event listener* pada Unity secara otomatis mendeteksi perpindahan tersebut dan mengaktifkan *pause state* di mana seluruh mekanika permainan dihentikan sementara untuk memberikan ruang bagi sistem mengeksekusi pipeline rekomendasi. Pada kondisi inilah seluruh data capaian yang telah terakumulasi dikirimkan ke *adaptation engine* untuk diproses lebih lanjut. Karena masing-masing parameter memiliki skala yang berbeda, maka dilakukan proses normalisasi untuk mengubah seluruh nilai ke dalam rentang 0 hingga 1, sehingga setiap parameter memiliki kontribusi yang seimbang dalam proses perhitungan selanjutnya.

Setelah proses normalisasi, sistem menghitung faktor capaian pemain (Fcp) yang merepresentasikan kondisi performa pemain secara keseluruhan. Nilai Fcp ini digunakan sebagai bagian dari fungsi *fitness* pada algoritma *Firefly* untuk menyesuaikan proses optimasi dengan kondisi aktual pemain. Semakin tinggi nilai Fcp, maka semakin baik performa pemain, sehingga sistem akan menghasilkan rekomendasi yang lebih sesuai dengan kondisi tersebut. Tahap ini menjadi penghubung antara data *gameplay* dan proses optimasi dalam sistem.

Tahap berikutnya adalah optimasi bobot kriteria menggunakan *Firefly Algorithm*. Pada tahap ini, setiap *firefly* merepresentasikan kandidat solusi berupa kombinasi bobot kriteria. Proses dimulai dengan inisialisasi populasi, dilanjutkan dengan evaluasi setiap solusi menggunakan fungsi *fitness*, kemudian pergerakan *firefly* menuju solusi dengan nilai *fitness* yang lebih tinggi. Tahap ini menghasilkan bobot kriteria optimal yang telah disesuaikan dengan kondisi performa pemain. Bobot optimal yang diperoleh kemudian digunakan dalam proses perhitungan metode TOPSIS untuk melakukan pemeringkatan alternatif sepatu.

Metode ini mengevaluasi setiap alternatif berdasarkan kedekatannya terhadap solusi ideal, sehingga diperoleh nilai preferensi yang digunakan untuk menentukan sepatu terbaik. Hasil akhir dari proses ini berupa rekomendasi sepatu yang dikirimkan kembali ke game engine dan ditampilkan kepada pemain melalui antarmuka permainan. Sistem ini bersifat adaptif karena bobot kriteria yang digunakan tidak bersifat tetap, melainkan berubah secara dinamis mengikuti performa pemain. Integrasi antara *Firefly Algorithm* dan metode TOPSIS memastikan bahwa rekomendasi yang dihasilkan tidak hanya optimal secara matematis, tetapi juga relevan dengan kondisi permainan pemain.

3.4.1 Mekanisme Transisi dan Perpindahan Level

Sistem adaptif dalam penelitian ini tidak bekerja secara terus-menerus sepanjang permainan. Alih-alih melakukan kalkulasi tanpa henti yang berpotensi membebani runtime, sistem dirancang untuk bekerja secara sekuensial yang hanya aktif ketika suatu *event* tertentu terpicu pada akhir setiap tahapan permainan.

Pendekatan berbasis *state machine* ini dipilih karena lebih efisien secara komputasi sekaligus lebih mudah dikendalikan secara logis.

Proses transisi antar level berlangsung melalui empat tahapan yang saling berurutan, sebagai berikut:

1. Tahap *Baseline* (Level 0)

Permainan diawali dengan Level 0, yakni *Baseline Level*, di mana karakter pemain bergerak dalam kondisi standar tanpa atribut tambahan dari sepatu khusus apapun. Level ini sengaja dirancang bukan sebagai tantangan, melainkan sebagai ruang pengukuran kemampuan murni pemain sebelum sistem memberikan rekomendasi. Selama Level 0 berjalan, komponen *Game Manager* secara langsung merekam empat metrik utama: akumulasi skor, total kerusakan yang diterima (*damage taken*), durasi waktu permainan (*play time*), dan jumlah nyawa yang tersisa (*lives left*).

2. Tahap Interupsi Transisi (*Trigger State*)

Begitu pemain berhasil menyelesaikan misi dan menyentuh gerbang keluar level, sebuah *event listener* pada Unity langsung mendeteksi momen tersebut. Deteksi ini menjadi sinyal bagi sistem untuk mengaktifkan *pause state*, seluruh mekanika permainan dihentikan sementara guna memberikan ruang bagi algoritma untuk bekerja tanpa gangguan.

3. Tahap Pemanggilan *Pipeline* (*Execution State*)

Di dalam kondisi *pause state* proses inti berlangsung di mana *Game Manager* meneruskan seluruh data metrik yang telah terkumpul ke dalam *adaptation engine*, kemudian memanggil fungsi

`RecommendationSystem.Instance.GenerateRecommendation()`. Fungsi ini menjalankan rangkaian pipeline secara berurutan, yaitu normalisasi data performa pemain menggunakan metode *Min-Max*, penghitungan bobot Faktor Capaian Pemain (*Fcp*), optimasi bobot kriteria melalui *Firefly Algorithm*, lalu pemeringkatan pilihan sepatu dengan metode TOPSIS.

4. Tahap Injeksi Parameter (*Injection State*)

Hasil akhir dari kalkulasi TOPSIS berupa ID sepatu dengan nilai preferensi tertinggi V_i yang langsung dikirim kembali ke sistem permainan. Atribut teknis sepatu terpilih, seperti peningkatan kecepatan gerak atau efisiensi pengurangan *damage*, segera di-*inject* ke dalam *Player Controller*. Saat *pause state* dicabut dan Level 1 dimulai, pemain sudah otomatis menggunakan kemampuan baru yang telah disesuaikan secara personal berdasarkan performa mereka di level sebelumnya.

3.5 Penentuan Kriteria Sepatu

Kriteria merupakan faktor-faktor penilaian yang digunakan untuk mengevaluasi berbagai alternatif yang tersedia. Penentuan kriteria dalam penelitian ini tidak dilakukan secara subjektif, melainkan mengacu pada karakteristik teknis sepatu running yang umum digunakan dalam industri olahraga, khususnya pada kategori *road running shoes*. Faktor utama yang digunakan dalam evaluasi performa sepatu lari secara langsung mempengaruhi kenyamanan, efisiensi gerak, dan keamanan pengguna saat berlari.

Tabel 3.2 berikut menunjukkan penentuan kriteria pada penelitian ini yang dibuat berdasarkan karakteristik teknis sepatu running asli yang umum digunakan pada kategori road running shoes:

Tabel 3. 2 Kriteria Sepatu Running

No	Kode	Kriteria	Jenis	Keterangan
1	C ₁	Berat Sepatu	Cost	Berat sepatu diukur dalam satuan gram berdasarkan spesifikasi resmi. Semakin ringan sepatu, semakin tinggi mobilitas karakter dalam game.
2	C ₂	Cushioning Level	Benefit	Ditentukan berdasarkan teknologi midsole yang digunakan, seperti Zoom Air, Boost, atau Fresh Foam. Cushioning berpengaruh terhadap pengurangan damage dalam <i>gameplay</i> .
3	C ₃	Responsivitas (<i>Energy Return</i>)	Benefit	Menggambarkan kemampuan sepatu dalam mengembalikan energi saat pijakan. Teknologi midsole dengan energy return tinggi akan meningkatkan performa pergerakan karakter.
4	C ₄	Traction (<i>Outsole Grip</i>)	Benefit	Ditentukan berdasarkan material dan pola outsole. Grip yang baik meningkatkan stabilitas pergerakan dalam eksplorasi medan normal.
5	C ₅	Stability Support	Benefit	Mengacu pada struktur pendukung seperti heel counter dan arch support. Stabilitas membantu mengurangi risiko kehilangan keseimbangan karakter.
6	C ₆	Durability Material	Benefit	Berdasarkan material upper dan outsole. Ketahanan material mempengaruhi daya tahan sepatu dalam penggunaan jangka panjang selama permainan.

Berbeda dengan pendekatan subjektif, seluruh kriteria dalam penelitian ini diturunkan dari spesifikasi teknik resmi produk sehingga meningkatkan objektivitas sistem pengambilan keputusan sebagai berikut:

Tabel 3. 3 Spesifikasi Resmi Produk untuk Kriteria

No	Kode	Kriteria	Dampak di Dunia Nyata	Adaptasi ke <i>Gameplay</i>
1	C ₁	Berat	Mempengaruhi kecepatan	Mempengaruhi movement speed
2	C ₂	Cushioning	Menyerap benturan	Mengurangi damage
3	C ₃	Responsiveness	Efisiensi energi	Meningkatkan akselerasi
4	C ₄	Traction	Mencegah slip	Stabilitas saat bergerak
5	C ₅	Stability	Menjaga keseimbangan	Mengurangi kesalahan gerakan
6	C ₆	Durability	Ketahanan sepatu	Konsistensi performa

3.6 Penentuan Alternatif Sepatu

Alternatif dalam penelitian ini merupakan produk *shoes running* nyata yang memiliki spesifikasi teknis resmi dari setiap brand yang digunakan. Pemilihan sepatu difokuskan pada kategori medan normal (*road running*) sesuai dengan desain lingkungan game yang tidak melibatkan medan ekstrem. Alternatif sepatu yang digunakan dalam penelitian ini berjumlah 10 produk (A_1 - A_{10}) yang ditampilkan pada tabel 3.4 berikut:

Tabel 3. 4 Alternatif Sepatu

No	Alt	Keterangan	Spesifikasi Teknis
1	A ₁	Nike Air Zoom Pegasus 40	Berat: sekitar 288 g (men size 9) Midsole: Nike React Foam + Zoom Air Unit Outsole: Rubber waffle traction pattern Upper: Engineered mesh Fungsi: daily running shoe dengan cushioning responsif
2	A ₂	Adidas Ultraboost 22	Berat: sekitar 333 g Midsole: BOOST foam Outsole: Stretchweb + Continental Rubber Upper: Primeknit textile Drop: sekitar 10 mm
3	A ₃	New Balance Fresh Foam 1080v13	Berat: sekitar 262 g Midsole: Fresh Foam X Upper: engineered mesh Kategori: premium daily running shoe
4	A ₄	ASICS Gel-Nimbus 25	Berat: sekitar 290 g Midsole: FF Blast+ Eco + GEL technology Outsole: AHAR rubber Upper: engineered knit upper
5	A ₅	Nike Structure 25	Berat: ± 290 g Midsole: Cushlon + Zoom Air Outsole: Durable rubber Upper: Engineered mesh Kategori: stability running
6	A ₆	Adidas Adizero SL	Berat: ± 243 g Midsole: Lightstrike EVA Outsole: Rubber Upper: mesh ringan
7	A ₇	ASIC Gel-Cumulus 25	Berat: ± 280 g Midsole: FF Blast+ Outsole: AHAR rubber Upper: engineered mesh
8	A ₈	Puma Velocity Nitro 2	Berat: ± 266 g Midsole: Nitro foam Outsole: PUMAGRIP Upper: engineered mesh
9	A ₉	Brooks Ghost 15	Berat: ± 286 g

No	Alt	Keterangan	Spesifikasi Teknis
			Midsole: DNA Loft v2 Outsole: Segmented crash pad Upper: mesh
10	A ₁₀	Saucony Ride 16	Berat: ± 282 g Midsole: PWRRUN foam Outsole: XT-900 rubber Upper: engineered mesh

Pemilihan alternatif berbasis produk nyata bertujuan untuk meningkatkan objektivitas sistem dan memastikan bahwa setiap nilai kriteria diturunkan langsung dari spesifikasi teknis yang terverifikasi.

3.7 Konversi Spesifikasi Teknis Sepatu ke Skor Kriteria

Spesifikasi teknis sepatu dikonversi ke dalam skala penilaian 1-5 untuk keperluan perhitungan TOPSIS. Proses konversi dilakukan berdasarkan rentang nilai kuantitatif maupun klasifikasi teknologi yang digunakan pada masing-masing sepatu. Skor yang lebih tinggi menunjukkan performa yang lebih baik terhadap kriteria yang bersifat *benefit*, sedangkan untuk kriteria *cost* nilai yang lebih rendah menunjukkan kondisi yang lebih baik. Konversi ini dilakukan untuk memastikan bahwa seluruh spesifikasi teknis sepatu dapat dinormalisasi ke dalam skala yang sama sehingga dapat diproses dalam matriks keputusan metode TOPSIS.

1. Berat Sepatu (C_1)

Nilai berat sepatu diperoleh dari spesifikasi teknis resmi masing-masing produk. Rentang nilai tersebut disusun berdasarkan distribusi umum berat sepatu running modern untuk kategori sepatu lari harian (*daily running shoes*). Sepatu yang lebih ringan memungkinkan pengguna untuk bergerak lebih cepat.

Tabel 3. 5 Konversi Skor C1

No	Berat Sepatu (gram)	Skor
1	< 270	5
2	270 – 290	4
3	290 – 310	3
4	310 – 330	2
5	> 330	1

2. Cushioning Level (C₂)

Cushioning level ditentukan berdasarkan teknologi midsole yang digunakan pada sepatu. Teknologi *midsole* yang lebih maju umumnya memiliki kemampuan penyerapan benturan (*shock absorption*) dan kenyamanan yang lebih tinggi.

Tabel 3. 6 Konversi Skor C2

No	Berat Sepatu (gram)	Skor
1	Teknologi premium cushioning tinggi (Zoom Air, Fresh Foam X)	5
2	Teknologi cushioning tinggi (Boost, Gel, React, DNA Loft)	4
3	Foam responsif standar (FlyteFoam, EVA responsif)	3
4	Foam dasar (EVA standar)	2
5	Cushioning minimal	1

3. Responsiveness / Energy Return (C₃)

Responsiveness menggambarkan kemampuan *midsole* dalam mengembalikan energi saat langkah di mana energy return yang lebih baik memungkinkan pergerakan yang lebih efisien. Penilaian dilakukan berdasarkan jenis teknologi *midsole* serta deskripsi performa yang tercantum pada spesifikasi produk.

Tabel 3. 7 Konversi Skor C3

No	Berat Sepatu (gram)	Skor
1	Energy return sangat tinggi	5
2	Energy return tinggi	4
3	Energy return sedang	3
4	Energy return rendah	2
5	Hampir tidak memiliki energy return	1

4. Traction / Outsole Grip (C₄)

Traction ditentukan berdasarkan material *outsole* dan pola tapak yang digunakan. Material *outsole* yang lebih kuat dan pola tapak yang lebih optimal memberikan daya cengkeram yang lebih baik pada permukaan tanah.

Tabel 3. 8 Konversi Skor C₄

No	Berat Sepatu (gram)	Skor
1	Rubber premium dengan grip tinggi (Continental Rubber, Contagrip)	5
2	Rubber berkualitas tinggi dengan traction kuat	4
3	Rubber standar dengan traction sedang	3
4	Rubber dasar dengan traction terbatas	2
5	Outsole dengan grip rendah	1

5. Stability Support (C₅)

Stability support ditentukan berdasarkan keberadaan struktur pendukung seperti *heel counter*, *arch support*, dan desain platform sepatu. Struktur stabilitas yang baik membantu menjaga keseimbangan saat bergerak.

Tabel 3. 9 Konversi Skor C₅

No	Berat Sepatu (gram)	Skor
1	Stabilitas sangat tinggi dengan struktur pendukung lengkap	5
2	Stabilitas tinggi dengan heel counter dan arch support	4
3	Stabilitas sedang	3
4	Stabilitas rendah	2
5	Hampir tidak memiliki dukungan stabilitas	1

6. Durability Material

Durability material ditentukan berdasarkan kualitas material upper dan outsole yang digunakan sepatu. Material dengan ketahanan tinggi memungkinkan sepatu digunakan dalam durasi permainan yang lebih lama.

Tabel 3. 10 Konversi Skor C₆

No	Berat Sepatu (gram)	Skor
1	Material premium dengan ketahanan sangat tinggi	5
2	Material berkualitas tinggi	4
3	Material standar dengan ketahanan cukup	3

No	Berat Sepatu (gram)	Skor
4	Material dengan ketahanan rendah	2
5	Material sangat tipis atau mudah rusak	1

3.8 Penilaian Alternatif Sepatu terhadap Kriteria

Setelah proses konversi spesifikasi teknis sepatu ke dalam skala penilaian 1 – 5 dilakukan pada setiap kriteria, langkah selanjutnya adalah menyusun tabel penilaian alternatif sepatu. Tabel ini berfungsi sebagai matriks keputusan awal yang akan digunakan dalam proses perhitungan metode TOPSIS. Skala skor 1 – 5 pada kriteria C1 (Berat Sepatu) memiliki makna yang berbeda dengan kriteria benefit. Karena C1 bersifat *cost*, skor yang lebih tinggi menunjukkan sepatu yang lebih ringan (lebih baik). Namun demikian, dalam proses normalisasi TOPSIS, skor 5 pada C1 menghasilkan nilai normalisasi yang lebih besar dibandingkan skor 3. Oleh karena itu, penentuan solusi ideal untuk C1 menggunakan logika *cost*: A^+ = nilai normalisasi terbobot terkecil (sepatu paling berat, skor terendah) dan A^- = nilai normalisasi terbobot terbesar (sepatu paling ringan, skor tertinggi).

Dengan demikian, sepatu yang paling ringan (skor C1 = 5) akan memiliki nilai normalisasi terbesar dan ditetapkan sebagai kondisi terburuk (A^-) untuk kriteria *cost*, sedangkan sepatu yang lebih berat secara relatif (skor C1 lebih rendah, nilai normalisasi lebih kecil) justru lebih dekat ke A^+ . Hal ini konsisten dengan definisi TOPSIS bahwa untuk kriteria *cost*, nilai yang lebih kecil adalah lebih baik. Perbedaan ini diimplementasikan secara eksplisit dalam kode melalui flag `isBenefit[0] = false` pada komponen TOPSIS.

1. Nike Air Zoom (A₁)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	288 g	4	masuk rentang 270–290 g
C ₂	Cushioning	React foam + Zoom Air	5	teknologi cushioning premium
C ₃	Responsiveness	Zoom Air energy return	4	responsif tapi bukan maksimal
C ₄	Traction	rubber waffle outsole	4	grip baik untuk road running
C ₅	Stability	padded heel + secure fit	4	stabilitas cukup baik
C ₆	Durability	engineered mesh + rubber outsole	4	material cukup tahan lama

2. Adidas Ultraboost 22 (A₂)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	333 g	3	berada di kisaran berat >310 g
C ₂	Cushioning	Boost midsole	4	cushioning tinggi, foam padat
C ₃	Responsiveness	Boost energy return	4	energy return baik dan stabil
C ₄	Traction	Continental Rubber outsole	5	grip sangat kuat
C ₅	Stability	Linear Energy Push system	4	sistem stabilisasi tambahan
C ₆	Durability	Primeknit + rubber tebal	5	outsole sangat tahan lama

3. New Balance Fresh Foam X 1080v13 (A₃)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	262 g	5	< 270 g
C ₂	Cushioning	Fresh Foam X	5	cushioning premium
C ₃	Responsiveness	Fresh Foam energy return	4	responsif tetapi fokus comfort
C ₄	Traction	rubber outsole	4	grip baik
C ₅	Stability	balances platform	4	stabilitas baik
C ₆	Durability	reinforced mesh	4	material tahan lama

4. ASICS Gel-Nimbus 25 (A₄)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	290 g	3	masuk rentang 290–310 g
C ₂	Cushioning	GEL + FF Blast+	4	cushioning tinggi
C ₃	Responsiveness	FF Blast+ foam	4	energy return baik
C ₄	Traction	AHAR rubber outsole	4	grip stabil
C ₅	Stability	wide platform + heel counter	5	stabilitas sangat baik
C ₆	Durability	AHAR outsole durability	5	material sangat tahan lama

5. Nike Structure 25 (A_5)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	290 g	3	masuk rentang 290–310 g
C ₂	Cushioning	Cushlon + Zoom Air	5	cushioning premium
C ₃	Responsiveness	Zoom Air	4	responsif tinggi
C ₄	Traction	Rubber Outsole	4	grip baik
C ₅	Stability	Stability system	5	sepatu stability
C ₆	Durability	Durable rubber + mesh	4	cukup tahan lama

6. Adidas Adizero SL (A_6)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	243 g	5	< 270 g
C ₂	Cushioning	Lightstrike EVA	3	bukan premium cushioning
C ₃	Responsiveness	Responsive EVA	4	cukup responsif
C ₄	Traction	Rubber outsole	3	standar
C ₅	Stability	Minim stability	3	tidak fokus stability
C ₆	Durability	Lightweight mesh	3	durability sedang

7. ASICS Gel-Cumulus 25 (A_7)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	± 280 g	4	270g – 290g
C ₂	Cushioning	FF Blast+	5	cushioning premium
C ₃	Responsiveness	FF Blast+	4	responsif baik
C ₄	Traction	AHAR rubber	4	grip kuat
C ₅	Stability	Neutral support	4	stabil
C ₆	Durability	Durable outsole	4	tahan lama

8. Puma Velocity Nitro 2 (A_8)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	± 266 g	5	< 270g
C ₂	Cushioning	Nitro foam	5	cushioning premium
C ₃	Responsiveness	Nitro energy return	5	sangat responsif
C ₄	Traction	PUMAGRIP	5	grip sangat tinggi
C ₅	Stability	Neutral	4	stabil cukup
C ₆	Durability	Rubber outsole kuat	4	tahan lama

9. Brooks Ghost 15 (A₉)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	± 286 g	4	270–290
C ₂	Cushioning	DNA Loft v2	4	cushioning tinggi
C ₃	Responsiveness	energy return	4	responsif stabil
C ₄	Traction	segmented outsole	4	grip baik
C ₅	Stability	balanced stability	4	stabil
C ₆	Durability	durable outsole	5	sangat tahan lama

10. Saucony Ride 16 (A₁₀)

No	Kriteria	Spesifikasi	Skor	Alasan
C ₁	Berat	± 282 g	4	270g – 290g
C ₂	Cushioning	PWRRUN foam	4	Cushioning tinggi
C ₃	Responsiveness	responsive foam	4	Responsif baik
C ₄	Traction	XT-900 rubber	5	Grip tinggi
C ₅	Stability	neutral support	4	Stabil
C ₆	Durability	durable outsole	5	Sangat tahan lama

Nilai pada tabel penilaian diperoleh dengan cara mencocokkan spesifikasi teknis masing-masing sepatu. Setiap alternatif kemudian diberikan skor pada masing-masing kriteria berdasarkan hasil konversi tersebut. Berdasarkan proses konversi spesifikasi teknis terhadap enam kriteria yang telah ditentukan, diperoleh tabel penilaian alternatif sepatu sebagai berikut.

Tabel 3. 11 Penilaian Alternatif Sepatu

No	Alternatif	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
1	A ₁	4	5	4	4	4	4
2	A ₂	3	4	4	5	4	5
3	A ₃	5	5	4	4	4	4
4	A ₄	3	4	4	4	5	5
5	A ₅	3	5	4	4	5	4
6	A ₆	5	3	4	3	3	3
7	A ₇	4	5	4	4	4	4
8	A ₈	5	5	5	5	4	4
9	A ₉	4	4	4	4	4	5
10	A ₁₀	4	4	4	5	4	5

3.9 Optimasi Bobot Kriteria

Dalam penelitian ini, *Firefly Algorithm* digunakan untuk melakukan optimasi bobot kriteria berdasarkan capaian pemain. Setiap *firefly* merepresentasikan kandidat solusi berupa kombinasi bobot kriteria yang akan dievaluasi menggunakan fungsi *fitness*. Evaluasi kualitas solusi dilakukan dengan memanfaatkan nilai preferensi dari metode TOPSIS.

3.9.1 Representasi Solusi *Firefly*

Dalam penelitian ini, setiap *firefly* direpresentasikan sebagai vektor bobot kriteria w_1 - w_6 untuk masing-masing kriteria sepatu. Nilai setiap bobot berada pada rentang 0 hingga 1, dan memenuhi kondisi normalisasi. Untuk mencegah bias akibat konsentrasi bobot pada satu kriteria dominan, setiap komponen bobot dibatasi pada rentang minimum 0,0 dan maksimum 1,0 sebelum proses normalisasi dilakukan. Pembatasan ini diterapkan pada tahap inisialisasi populasi *firefly* melalui parameter $WeightMin = 0,0$ dan $WeightMax = 1,0$, sehingga setiap *firefly* dibangkitkan dalam rentang yang terdefinisi sebelum dinormalisasi agar jumlah total bobot sama dengan satu. Dengan constraint ini, distribusi bobot antar kriteria tetap proporsional dan tidak ada satu kriteria pun yang secara sepihak mendominasi keputusan rekomendasi.

3.9.2 Parameter *Firefly Algorithm*

Dalam penelitian ini, parameter *Firefly Algorithm* ditentukan untuk menyesuaikan dengan karakteristik data dan kebutuhan sistem. Parameter yang digunakan ditunjukkan oleh tabel 3.12 berikut:

Tabel 3. 12 Parameter *Firefly Algorithm*

Parameter	Simbol	Nilai	Keterangan
Jumlah <i>firefly</i>	n	10	Jumlah solusi dalam populasi
Iterasi maksimum	t_max	20	Batas jumlah iterasi
Daya tarik awal	β_0	1	Tingkat ketertarikan awal antar <i>firefly</i>
Koefisien penyerapan cahaya	γ	1	Mengontrol penurunan daya tarik
Parameter random	α	0.2	Faktor eksplorasi

Pemilihan parameter didasarkan pada penyesuaian terhadap skala permasalahan serta untuk menghasilkan proses konvergensi yang stabil dalam sistem. Berikut adalah penjelasan dari lima parameter *Firefly Algorithm* yang digunakan untuk menyesuaikan dengan karakteristik data dan kebutuhan sistem:

1. Jumlah *Firefly* (n) yang digunakan merepresentasikan jumlah solusi kandidat dalam setiap iterasi.
2. Iterasi Maksimum (t_max) sebanyak 20 kali dilakukan untuk memastikan proses optimasi mencapai kondisi stabil.
3. Daya Tarik Awal (β_0) ditetapkan sebesar 1 untuk menjaga keseimbangan antara eksploitasi dan konvergensi solusi.
4. Koefisien Penyerapan Cahaya (γ) ditetapkan sebesar 1 untuk mengatur penurunan intensitas daya tarik antar *firefly* secara proporsional terhadap jarak.
5. Parameter *Random* (α) sebesar 0.2 digunakan untuk memberikan variasi pergerakan *firefly* agar tidak terjebak pada solusi lokal.

3.9.3 Inisialisasi Populasi *Firefly*

Populasi awal *firefly* dibangkitkan secara acak dengan jumlah tertentu. Setiap *firefly* memiliki nilai bobot yang berbeda sebagai kandidat solusi awal. Parameter yang digunakan dalam algoritma meliputi jumlah *firefly* (n), koefisien

daya tarik (β), parameter penyerapan cahaya (γ), dan faktor randomisasi (α). Salah satu *firefly* dapat direpresentasikan sebagai berikut:

Tabel 3. 13 Inisialisasi Populasi *Firefly*

No	Kriteria	Bobot
1	C_1	0.15
2	C_2	0.20
3	C_3	0.18
4	C_4	0.17
5	C_5	0.15
6	C_6	0.15

3.9.4 Perhitungan Faktor Capaian Pemain (Fcp)

Faktor capaian pemain (Fcp) digunakan untuk merepresentasikan performa pemain dalam permainan yang akan mempengaruhi proses evaluasi solusi. Nilai Fcp dihitung berdasarkan parameter capaian pemain yang telah dinormalisasi. Parameter yang digunakan meliputi *score* (S), *damage taken* (D), *play time* (T), *lives left* (L). Berdasarkan persamaan 2.1 yang telah dijabarkan pada Sub-bab 2.3.3, berikut adalah implementasi perhitungannya menggunakan sampel data capaian pemain:

Tabel 3. 14 Contoh Data Pemain

No	Parameter	Nilai
1	<i>Score</i> (S)	80
2	<i>Damage</i> (D)	30
3	<i>Play time</i> (T)	60
4	<i>Lives left</i> (L)	3

Maka, dari asumsi data capaian pemain tersebut, nilai dinormalisasi sehingga menghasilkan nilai normalisasi sebagai berikut:

Tabel 3. 15 Contoh Normalisasi Data Pemain

No	Parameter	Nilai Normalisasi
1	S	0.8
2	D	$0.3 \rightarrow (1-D) = 0.7$
3	T	0.6
4	L	0.6

Dengan asumsi bobot parameter:

$$\alpha_1 = 0.3, \alpha_2 = 0.3, \alpha_3 = 0.2, \alpha_4 = 0.2$$

Hitung F_{cp} :

$$F_{cp} = (0.3 \times 0.8) + (0.3 \times 0.7) + (0.2 \times 0.6) + (0.2 \times 0.6)$$

$$F_{cp} = 0.24 + 0.21 + 0.12 + 0.12$$

$$F_{cp} = 0.69$$

Nilai F_{cp} sebesar 0.69 ini selanjutnya digunakan sebagai salah satu komponen dalam perhitungan fungsi fitness pada Sub-bab 3.9.5.

3.9.5 Fungsi *Fitness*

Fungsi *fitness* digunakan untuk mengevaluasi kualitas setiap *firefly*. Dalam penelitian ini, nilai *fitness* ditentukan berdasarkan hasil perhitungan metode TOPSIS yang dikombinasikan dengan faktor capaian pemain. Parameter yang digunakan yaitu nilai preferensi tertinggi hasil evaluasi menggunakan TOPSIS (V_{max}) dan faktor capaian pemain (F_{cp}). Nilai V_{max} diperoleh dari hasil perhitungan metode TOPSIS menggunakan bobot yang direpresentasikan oleh *firefly*. Semakin tinggi nilai *fitness*, semakin baik kualitas bobot yang dihasilkan.

Sebagai contoh perhitungan:

$$Fitness = 0.77 \times 0.69 = 0.5313$$

3.9.6 Evaluasi *Firefly* Menggunakan TOPSIS

Setiap *firefly* dievaluasi dengan cara menggunakan bobot yang dimilikinya dalam perhitungan metode TOPSIS untuk memperoleh nilai preferensi. Pada tahap ini, metode TOPSIS tidak digunakan untuk menghasilkan rekomendasi akhir,

melainkan sebagai alat evaluasi dalam fungsi *fitness* untuk menilai kualitas bobot. Proses perhitungan TOPSIS secara rinci disajikan pada Subbab 3.10.

3.9.7 Pergerakan *Firefly*

Pergerakan *firefly* dilakukan berdasarkan nilai *fitness*. *Firefly* dengan nilai *fitness* lebih rendah akan bergerak menuju *firefly* dengan nilai *fitness* lebih tinggi. Parameter yang digunakan posisi *firefly* ke- i (x_i), posisi *firefly* dengan *fitness* lebih tinggi (x_j), daya tarik (β), dan faktor randomisasi(α).

3.9.8 Proses Iterasi dan Penentuan Bobot Optimal

Proses evaluasi dan pergerakan *firefly* dilakukan secara iteratif hingga mencapai kondisi berhenti, seperti jumlah iterasi maksimum atau konvergensi nilai *fitness*. *Firefly* dengan nilai *fitness* tertinggi dipilih sebagai solusi terbaik yang merepresentasikan bobot kriteria optimal. Bobot ini selanjutnya digunakan dalam metode TOPSIS (Subbab 3.10) untuk menghasilkan rekomendasi sepatu sebagai output sistem.

3.10 Perhitungan Metode TOPSIS

Metode TOPSIS dalam penelitian ini digunakan untuk melakukan pemeringkatan terhadap alternatif sepatu yang berjumlah 10 produk berdasarkan tingkat kedekatannya terhadap solusi ideal. Perhitungan dilakukan berdasarkan bobot kriteria hasil optimasi *Firefly Algorithm* yang telah diperoleh pada tahap sebelumnya, sehingga nilai preferensi yang dihasilkan bersifat dinamis dan adaptif terhadap kondisi pemain. Metode ini dipilih karena mampu memberikan keputusan

yang objektif dengan mempertimbangkan jarak terhadap solusi ideal positif dan solusi ideal negatif.

3.10.1 Matriks Keputusan

Matriks keputusan disusun berdasarkan hasil penilaian alternatif sepatu yang telah dijelaskan pada Subbab 3.8. Setiap alternatif memiliki nilai pada masing-masing kriteria dalam skala 1–5 yang digunakan sebagai input dalam proses perhitungan metode TOPSIS.

Perhitungan pada tahap selanjutnya dilakukan terhadap seluruh elemen dalam matriks keputusan. Namun, untuk mempermudah pemahaman, contoh perhitungan hanya ditampilkan pada sebagian alternatif.

3.10.2 Normalisasi Matriks Keputusan

Normalisasi dilakukan untuk menyamakan skala antar kriteria menggunakan metode *vector normalization*. Sebagai contoh, normalisasi untuk kriteria C_1 (berat sepatu) pada alternatif pertama dihitung sebagai berikut:

$$\begin{aligned} &= \sqrt{(4^2 + 3^2 + 5^2 + 3^2 + 3^2 + 5^2 + 4^2 + 5^2 + 4^2 + 4^2)} \\ &= \sqrt{(16 + 9 + 25 + 9 + 9 + 25 + 16 + 25 + 16 + 16)} \\ &= \sqrt{(166)} \\ &= 12.884 \end{aligned}$$

Kemudian dihitung normalisasi untuk contoh C_1 :

$$r_{11} = 4 / 12.884 = 0.310$$

$$r_{21} = 3 / 12.884 = 0.233$$

Proses normalisasi dilakukan terhadap seluruh alternatif untuk setiap kriteria, sehingga menghasilkan matriks ternormalisasi yang memiliki skala yang seragam. Hasil normalisasi matriks dapat dilihat pada tabel 3.16 berikut:

Tabel 3. 16 Normalisasi Matriks Keputusan

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.310	0.355	0.308	0.298	0.306	0.291
2	A ₂	0.233	0.284	0.308	0.373	0.306	0.364
3	A ₃	0.388	0.355	0.308	0.298	0.306	0.291
4	A ₄	0.233	0.284	0.308	0.298	0.382	0.364
5	A ₅	0.233	0.355	0.308	0.298	0.382	0.291
6	A ₆	0.388	0.213	0.308	0.224	0.229	0.218
7	A ₇	0.310	0.355	0.308	0.298	0.306	0.391
8	A ₈	0.388	0.355	0.385	0.373	0.306	0.291
9	A ₉	0.310	0.284	0.308	0.298	0.306	0.364
10	A ₁₀	0.310	0.284	0.308	0.373	0.306	0.364

Hasil normalisasi menunjukkan bahwa seluruh nilai pada matriks keputusan telah dikonversi ke dalam skala yang seragam, sehingga setiap kriteria dapat dibandingkan secara proporsional. Nilai normalisasi ini selanjutnya digunakan dalam proses pembobotan dengan bobot kriteria hasil optimasi *Firefly Algorithm*.

3.10.3 Matriks Ternormalisasi Terbobot

Matriks ternormalisasi kemudian dikalikan dengan bobot kriteria yang diperoleh dari hasil optimasi menggunakan *Firefly Algorithm*. Bobot kriteria yang digunakan merupakan salah satu hasil optimasi *Firefly Algorithm* yang telah disesuaikan dengan capaian pemain, seperti disajikan pada tabel 3.13 Sub-bab 3.9 Sub-sub-bab 3.9.3.

Contoh perhitungan untuk A₁ pada C₁:

$$Y_{11} = 0.15 \times 0.310 = 0.0465$$

Contoh perhitungan untuk A1 pada C2:

$$Y_{12} = 0.20 \times 0.348 = 0.0696$$

Setelah dilakukan perhitungan matriks ternormalisasi yang dikalikan dengan bobor kriteria, maka akan menghasilkan matriks ternormalisasi terbobot yang ditunjukkan oleh tabel 3.17 berikut:

Tabel 3. 17 Matriks Ternormalisasi Terbobot

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.047	0.071	0.055	0.051	0.046	0.044
2	A ₂	0.035	0.057	0.055	0.063	0.046	0.055
3	A ₃	0.058	0.071	0.055	0.051	0.046	0.044
4	A ₄	0.035	0.057	0.055	0.051	0.057	0.055
5	A ₅	0.035	0.071	0.055	0.051	0.057	0.044
6	A ₆	0.058	0.043	0.055	0.038	0.034	0.033
7	A ₇	0.047	0.071	0.055	0.051	0.046	0.044
8	A ₈	0.058	0.071	0.069	0.063	0.046	0.044
9	A ₉	0.047	0.057	0.055	0.051	0.046	0.055
10	A ₁₀	0.047	0.057	0.055	0.063	0.046	0.055

Matriks ternormalisasi terbobot menunjukkan nilai setiap alternatif setelah mempertimbangkan bobot kriteria hasil optimasi *Firefly Algorithm*. Nilai ini mencerminkan kontribusi masing-masing kriteria terhadap setiap alternatif dalam proses pengambilan keputusan.

3.10.4 Solusi Ideal Positif dan Negatif

Dalam penelitian ini, kriteria dibagi menjadi *Cost* (C₁) dan *Benefit* (C₂, C₃, C₄, C₅, C₆). Solusi ideal positif (A⁺) ditentukan di mana nilai minimum untuk kriteria *cost* dan nilai maksimum untuk kriteria *benefit*. Sedangkan solusi ideal negatif (A⁻) ditentukan di mana nilai maksimum untuk kriteria *cost* dan nilai minimum untuk kriteria *benefit*.

Penentuan solusi ideal dilakukan berdasarkan seluruh alternatif yang tersedia, sehingga nilai maksimum dan minimum pada setiap kriteria dipengaruhi oleh keseluruhan data alternatif. Setelah itu, diambil data dari tabel matriks terbobot sehingga memperoleh hasil solusi ideal positif dan negatif sebagai berikut:

Tabel 3. 18 Solusi Ideal Positif

No	Kriteria	Nilai	Alasan
1	C ₁	0.0349	Nilai Terkecil
2	C ₂	0.0711	Terbesar
3	C ₃	0.0692	Terbesar
4	C ₄	0.0634	Terbesar
5	C ₅	0.0574	Terbesar
6	C ₆	0.0546	Terbesar

Tabel 3. 19 Solusi Ideal Negatif

No	Kriteria	Nilai	Alasan
1	C ₁	0.0582	Nilai Terbesar
2	C ₂	0.0426	Terkecil
3	C ₃	0.0554	Terkecil
4	C ₄	0.0380	Terkecil
5	C ₅	0.0344	Terkecil
6	C ₆	0.0327	Terkecil

Solusi ideal positif (A^+) merepresentasikan kondisi terbaik dari setiap kriteria, yaitu nilai minimum untuk kriteria cost dan nilai maksimum untuk kriteria *benefit*. Sebaliknya, solusi ideal negatif (A^-) merepresentasikan kondisi terburuk dari setiap kriteria. Penentuan solusi ideal ini dilakukan berdasarkan seluruh alternatif yang tersedia, sehingga hasilnya mencerminkan kondisi optimal relatif dalam dataset yang digunakan.

3.10.5 Perhitungan Jarak terhadap Solusi Ideal

Jarak setiap alternatif terhadap solusi ideal dihitung menggunakan metode Euclidean. Perhitungan jarak dilakukan untuk setiap alternatif terhadap solusi ideal

positif dan negatif, sehingga diperoleh nilai kedekatan relatif masing-masing alternatif.

Contoh perhitungan untuk alternatif pertama (A_1):

$$D1+ = \sqrt{(y11 - y1+)^2 + \dots + (y16 - y6+)^2}$$

Sehingga memperoleh hasil perhitungan jarak solusi ideal positif dan jarak solusi ideal negatif sebagai berikut:

Tabel 3. 20 Jarak terhadap Solusi Ideal

No	Alternatif	D+	D-
1	A ₁	0.0272	0.0369
2	A ₂	0.0229	0.0448
3	A ₃	0.0339	0.0350
4	A ₄	0.0235	0.0438
5	A ₅	0.0217	0.0465
6	A ₆	0.0566	0.0000
7	A ₇	0.0272	0.0369
8	A ₈	0.0282	0.0436
9	A ₉	0.0287	0.0333
10	A ₁₀	0.0257	0.0400

3.10.6 Perhitungan Nilai Preferensi

Nilai preferensi dihitung untuk menentukan tingkat kedekatan alternatif terhadap solusi ideal. Nilai V_i berada pada rentang 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan bahwa alternatif tersebut lebih baik. Dalam penelitian ini, nilai preferensi memiliki fungsi sebagai dasar pemeringkatan alternatif sepatu dan input dalam fungsi fitness pada *Firefly Algorithm*. Nilai Preferensi ditunjukkan pada tabel 3.21 berikut:

Tabel 3. 21 Nilai Preferensi

No	Alternatif	D+	D-	V_i
1	A ₁	0.0272	0.0369	0.5757
2	A ₂	0.0229	0.0448	0.6617
3	A ₃	0.0339	0.0350	0.5080
4	A ₄	0.0235	0.0438	0.6508
5	A ₅	0.0217	0.0465	0.6818

No	Alternatif	D+	D-	Vi
6	A ₆	0.0566	0.0000	0.0000
7	A ₇	0.0272	0.0369	0.5757
8	A ₈	0.0282	0.0436	0.6072
9	A ₉	0.0287	0.0333	0.5371
10	A ₁₀	0.0257	0.0400	0.6088

3.10.7 Pemeringkatan Alternatif dan Interpretasi dalam *Gameplay*

Berdasarkan nilai preferensi yang diperoleh, dilakukan pemeringkatan alternatif sepatu. Alternatif dengan nilai preferensi tertinggi dipilih sebagai sepatu yang paling direkomendasikan oleh sistem. Hasil pemeringkatan alternatif sepatu tidak hanya digunakan sebagai output numerik, tetapi diinterpretasikan dalam konteks gameplay. Sepatu dengan nilai preferensi tertinggi direkomendasikan kepada pemain karena memiliki kombinasi atribut yang paling sesuai dengan kondisi performa pemain saat itu. Dengan demikian, sistem tidak hanya memberikan keputusan matematis, tetapi juga mendukung strategi pemain dalam permainan. Hasil pemeringkatan alternatif ditunjukkan pada tabel berikut:

Tabel 3. 22 Pemeringkatan Alternatif

Rank	Alternatif	Vi
1	A ₅	0.6818
2	A ₂	0.6617
3	A ₄	0.6508
4	A ₁₀	0.6088
5	A ₈	0.6072
6	A ₁	0.5757
7	A ₇	0.5757
8	A ₉	0.5371
9	A ₃	0.5080
10	A ₆	0.0000

3.10.8 Keterkaitan dengan Sistem Adaptif Game

Dalam penelitian ini, metode TOPSIS tidak berdiri sendiri, melainkan terintegrasi dengan *Firefly Algorithm*. Bobot kriteria yang digunakan dalam

TOPSIS merupakan hasil optimasi yang telah mempertimbangkan capaian pemain, seperti skor, kerusakan, waktu bermain, dan sisa nyawa. Dengan demikian, nilai preferensi yang dihasilkan oleh TOPSIS akan berubah sesuai dengan kondisi pemain, sehingga sistem dapat memberikan rekomendasi sepatu yang bersifat adaptif. Integrasi ini menjadikan metode TOPSIS tidak hanya berfungsi sebagai alat pengambilan keputusan, tetapi juga sebagai bagian dari mekanisme adaptasi sistem dalam menyesuaikan rekomendasi terhadap performa pemain secara dinamis.

3.11 Justifikasi Pemilihan Metode

Pemilihan metode dalam penelitian ini didasarkan pada kebutuhan sistem untuk menghasilkan keputusan yang optimal, adaptif, dan mampu menangani banyak kriteria secara dinamis. Oleh karena itu, digunakan kombinasi *Firefly Algorithm* dan metode TOPSIS yang saling melengkapi dalam proses pengambilan keputusan.

Firefly Algorithm dipilih karena memiliki kemampuan yang baik dalam menyelesaikan permasalahan optimasi nonlinier dan multidimensi. Dalam konteks penelitian ini, optimasi dilakukan terhadap bobot kriteria yang tidak bersifat tetap, melainkan berubah sesuai dengan kondisi performa pemain. Karakteristik *firefly* yang berbasis metaheuristik memungkinkan proses eksplorasi ruang solusi secara global, sehingga tidak mudah terjebak pada solusi lokal. Selain itu, mekanisme pergerakan berbasis intensitas cahaya (*fitness*) memungkinkan algoritma ini secara efektif menemukan kombinasi bobot kriteria yang paling optimal berdasarkan kondisi yang dinamis.

Di sisi lain, metode TOPSIS dipilih karena kemampuannya dalam menyelesaikan masalah pengambilan keputusan multikriteria secara objektif. TOPSIS bekerja dengan mempertimbangkan kedekatan setiap alternatif terhadap solusi ideal positif dan solusi ideal negatif, sehingga mampu memberikan hasil pemeringkatan yang lebih representatif. Metode ini juga memiliki keunggulan dalam hal kesederhanaan perhitungan serta kemampuannya dalam mengakomodasi berbagai jenis kriteria, baik yang *bersifat* benefit maupun *cost*, sehingga sesuai untuk digunakan dalam proses evaluasi alternatif sepatu.

Integrasi antara Firefly Algorithm dan TOPSIS dalam penelitian ini dilakukan dengan memanfaatkan keunggulan masing-masing metode. Firefly Algorithm digunakan untuk mengoptimasi bobot kriteria secara adaptif, sedangkan TOPSIS digunakan untuk melakukan pemeringkatan alternatif berdasarkan bobot yang telah dioptimasi tersebut. Dengan kombinasi ini, sistem tidak hanya mampu menghasilkan keputusan yang optimal secara matematis, tetapi juga adaptif terhadap kondisi performa pemain yang berubah-ubah selama permainan berlangsung.

Dengan demikian, pemilihan kombinasi kedua metode ini dinilai tepat untuk mendukung pengembangan sistem adaptif dalam game, karena mampu mengintegrasikan proses optimasi dan pengambilan keputusan secara efektif dalam satu kerangka sistem yang terstruktur.

3.12 Skenario Pengujian Adaptabilitas

Pengujian adaptabilitas sistem pada game *The Night Dreamers* dirancang khusus untuk memvalidasi *Firefly Algorithm* dan TOPSIS terhadap berbagai

karakteristik pengguna. Pengujian ini menggunakan tiga skenario mock-up performa kuantitatif yang merepresentasikan variasi karakteristik gaya bermain serta tingkat risiko cedera yang dihadapi oleh karakter di dalam permainan. Setiap skenario menggunakan kombinasi parameter capaian nyata pemain, meliputi *score*, *damage taken*, *play time*, dan *lives left* sebagai variabel masukan (*input*) utama bagi *adaptation engine*. Melalui simulasi variasi data tersebut, sistem diuji untuk melihat apakah proses pembaruan vektor bobot kriteria oleh *Firefly Algorithm* dan kalkulasi koefisien kedekatan oleh TOPSIS dapat bergeser secara logis dan responsif demi mendukung kebutuhan taktis pemain pada level berikutnya.

Berikut adalah rincian desain dari ketiga skenario *mock-up* yang merepresentasikan tingkat kemahiran gaya bermain yang berbeda:

3.12.1 Skenario 1: Eksplorasi Aman (Pemula Berhati-hati)

- a. Karakteristik: Pemain baru yang bermain secara pasif dan sangat berhati-hati. Pemain cenderung berjalan pelan, menghindari konfrontasi dengan entitas Nightmares, dan lebih fokus pada observasi peta.
- b. Parameter Capaian (*Input*): Durasi penyelesaian (*play time*) cukup lama, damage yang diterima sangat rendah, dan skor yang dikumpulkan relatif standar/rendah.
- c. Ekspetasi Hasil Sistem (*Output*): Karena pergerakan yang lambat namun aman, *adaptation engine* dituntut untuk menjaga kenyamanan dasar. Algoritma diekspektasikan merespons dengan memprioritaskan kriteria kenyamanan atau *Cushioning* (C_2) dan stabilitas (C_5) guna memberikan keamanan melangkah bagi pemain pemula.

3.12.2 Skenario 2: Eksplorasi Agresif (Pemula Ceroboh)

- a. Karakteristik: Pemain pemula yang bermain impulsif dan terburu-buru. Pemain sering melakukan *random jumping* (melompat tanpa arah) dan *random running* tanpa memperhitungkan posisi musuh atau rintangan.
- b. Parameter Capaian (*Input*): Akibat kecerobohan tersebut, damage yang diterima menjadi sangat tinggi, skor rendah karena banyak manuver yang gagal, dan durasi penyelesaian (*play time*) menjadi lama karena seringnya terjadi repetisi kegagalan.
- c. Ekspetasi Hasil Sistem (*Output*): Sistem harus merespons risiko tingkat cedera yang masif ini dengan pertahanan. Algoritma diekspektasikan akan mendistribusikan bobot terbesar pada kriteria *Durability* (C_6) untuk mempertahankan daya tahan item dari kerusakan beruntun, serta *Cushioning* (C_2) untuk meredam benturan akibat lompatan ceroboh yang dilakukan pemain.

3.12.3 Skenario 3: Penguasaan Gameplay (Pemain Ahli / Master)

- a. Karakteristik: Pemain tingkat lanjut yang telah memahami seluruh mekanika permainan, lokasi kunci, dan pola pergerakan musuh. Pemain bergerak sangat taktis dan efisien.
- b. Parameter Capaian (*Input*): Score maksimal, tingkat kerusakan (*damage*) sangat rendah hingga nol, jumlah nyawa (*lives*) utuh, dan durasi (*play time*) sangat cepat (konsep speedrunning).
- c. Ekspetasi Hasil Sistem (*Output*): Pemain dengan mekanika sempurna tidak lagi membutuhkan proteksi. Sistem diekspektasikan merespons hal ini

dengan merekomendasikan sepatu yang memiliki kriteria Berat sangat ringan (C_1) dan *Responsiveness* tertinggi (C_3). Rekomendasi sepatu ini difokuskan sepenuhnya untuk melipatgandakan kelincahan dan kecepatan karakter (*boosting agility*).

Untuk memberikan pemetaan rancangan dari ketiga skenario pengujian adaptabilitas di atas, disajikan bentuk matriks terstruktur pada Tabel 3.23 berikut:

Tabel 3. 23 Matriks Desain Skenario Pengujian Adaptabilitas Sistem

No	Skenario Pengujian	Karakteristik Gaya Bermain	Simulasi Parameter Masukan (<i>Input</i>)	Prioritas Kriteria Kunci (<i>Firefly Algorithm</i>)	Ekspetasi Karakteristik Rekomendasi (TOPSIS)
1	Skenario 1 (Eksplorasi Aman)	Pemain pemula yang bermain pasif dan sangat berhati-hati. Cenderung berjalan pelan, menghindari konfrontasi dengan musuh, dan fokus pada observasi peta.	<i>Score</i> : Standar <i>Damage</i> : Sangat Rendah <i>Play Time</i> : Cukup Lama <i>Lives Left</i> : Maksimal	Fokus pada kriteria kenyamanan dan keamanan dasar, yaitu <i>Cushioning</i> (C_2) dan <i>Stability</i> (C_5).	Merekomendasikan sepatu dengan tingkat peredaman dan stabilitas tinggi guna memberikan keamanan serta kenyamanan melangkah bagi pemain pemula.
2	Skenario 2 (Eksplorasi Agresif)	Pemain pemula yang impulsif dan terburu-buru. Sering melakukan <i>random jumping</i> dan <i>random running</i> tanpa memperhitungkan posisi musuh atau rintangan.	<i>Score</i> : Rendah <i>Damage</i> : Sangat Tinggi <i>Play Time</i> : Lama <i>Lives Left</i> : Minimal	Distribusi bobot terbesar pada kriteria ketahanan material (<i>Durability</i> / C_6) dan peredaman benturan (<i>Cushioning</i> / C_2).	Merekomendasikan sepatu dengan daya tahan fisik tertinggi untuk menoleransi kerusakan beruntun, serta <i>cushioning</i> kuat untuk meredam lompatan ceroboh.
3	Skenario 3 (Penguasaan <i>Game-play</i>)	Pemain tingkat lanjut (<i>Master</i>) yang bermain taktis, efisien, serta sangat memahami mekanika permainan, tata letak kunci, dan pola musuh (<i>speedrunner</i>).	<i>Score</i> : Maksimal <i>Damage</i> : Sangat Rendah <i>Play Time</i> : Cepat <i>Lives Left</i> : Maksimal	Dominasi penuh pada kriteria <i>Berat Sepatu</i> (C_1) yang sangat ringan dan tingkat <i>Responsiveness</i> (C_3) tertinggi.	Merekomendasikan sepatu yang sepenuhnya mengesampingkan fungsi proteksi, murni difokuskan untuk melipatgandakan kelincahan (<i>boosting agility</i>) dan akselerasi karakter.

3.13 Rencana Pengujian Sistem

Pengujian dalam penelitian ini bertujuan untuk menilai sejauh mana tingkat kemudahan penggunaan (*usability*) dan kenyamanan pemain dalam berinteraksi dengan game adaptif yang dikembangkan. Metode yang digunakan dalam pengujian ini adalah *System Usability Scale* (SUS), karena metode ini sederhana, cepat, dan mampu memberikan hasil pengukuran yang terstandarisasi terhadap pengalaman pengguna.

Pengujian dilakukan untuk memperoleh gambaran umum mengenai efektivitas dan kepuasan pengguna selama memainkan game. Sebanyak 15 partisipan dilibatkan dalam proses pengujian, yang terdiri dari pelajar atau mahasiswa dengan kriteria memiliki pengalaman dasar dalam menggunakan game berbasis teknologi serta *game developer* sebagai ahli dalam bidang multimedia dan game. Sebelum proses pengujian dimulai, disediakan tutorial singkat agar seluruh partisipan memiliki pemahaman yang sama terhadap aturan dan alur permainan.

Selama sesi pengujian, partisipan diminta untuk memainkan game sampai level selesai, dengan fokus pada interaksi adaptif sistem, seperti pemilihan sepatu berdasarkan capaian pemain. Tujuan utama pengujian ini adalah mengevaluasi pengalaman bermain secara keseluruhan, yang mencakup kemudahan navigasi antarmuka, kejelasan instruksi, respon adaptif sistem, serta kenyamanan pemain selama berinteraksi dengan fitur-fitur permainan.

Setelah sesi bermain selesai, partisipan diminta untuk mengisi kuesioner SUS yang terdiri dari 10 pertanyaan, masing-masing dengan skala penilaian 1 hingga 5, di mana 1 berarti “sangat tidak setuju” dan 5 berarti “sangat setuju”.

Pertanyaan dalam kuesioner mencakup aspek-aspek seperti kemudahan, tingkat kompleksitas permainan, daya tarik fitur, dan potensi penggunaan game di masa mendatang. Berikut adalah daftar pertanyaan yang diajukan kepada pengguna dan nilai jawabannya:

Tabel 3. 24 Pertanyaan System Usability Scale (SUS)

No	Pertanyaan
1.	Saya berpikir game ini menarik untuk dimainkan.
2.	Saya merasa antarmuka game ini terlalu rumit.
3.	Saya merasa game ini mudah digunakan.
4.	Saya membutuhkan bantuan orang lain agar dapat memainkan game ini.
5.	Fitur-fitur dalam game ini berjalan dengan baik dan logis.
6.	Anda terlalu banyak inkonsistensi dalam game ini.
7.	Saya yakin sebagian besar orang akan cepat mempelajari cara memainkan game ini.
8.	Saya merasa game ini terlalu membingungkan.
9.	Saya merasa percaya diri saat memainkan game.
10.	Saya harus belajar banyak hal sebelum dapat memainkan game ini.

Tabel 3. 25 Nilai Jawaban Kuesioner Skala Likert

No	Jawaban	Nilai
1.	Sangat Tidak Setuju	1
2.	Tidak Setuju	2
3.	Netral	3
4.	Setuju	4
5.	Sangat Setuju	5

Interpretasi skor SUS dilakukan dengan mengacu pada rentang nilai berikut: skor di bawah 50 menunjukkan tingkat kegunaan yang rendah, skor antara 51 hingga 70 menunjukkan kegunaan yang cukup, dan skor di atas 71 menunjukkan tingkat kegunaan yang baik hingga sangat baik (Susila & Arsa, 2023). Data yang diperoleh dari hasil pengujian kemudian dianalisis untuk memperoleh rata-rata skor SUS dari seluruh partisipan, yang akan memberikan gambaran mengenai efektivitas game adaptif ini dalam memberikan pengalaman bermain yang interaktif dan nyaman bagi pengguna.

Analisis ini juga bertujuan untuk menilai sejauh mana mekanisme adaptasi sistem dan fitur rekomendasi sepatu dapat mendukung proses pembelajaran dan motivasi pemain selama bermain. Selain itu, umpan balik dari para partisipan akan dimanfaatkan untuk mengidentifikasi kelemahan atau aspek yang perlu disempurnakan, sehingga hasil pengembangan game diharapkan menjadi lebih optimal dan sesuai dengan kebutuhan pemain.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Bab ini menyajikan hasil implementasi sistem adaptif *Firefly*-TOPSIS pada game prototipe *The Night Dreamers* beserta analisis terhadap hasil pengujian yang telah dilakukan. Implementasi dilakukan secara menyeluruh mencakup perancangan game, integrasi sistem adaptif ke dalam mesin permainan, serta pelaksanaan uji coba dengan tiga skenario capaian pemain yang berbeda. Seluruh pembahasan disajikan secara sistematis dan merujuk pada kerangka desain yang telah ditetapkan pada Bab III.

4.1.1 Implementasi Game “*The Night Dreamers*”

Game *The Night Dreamers* diimplementasikan menggunakan Unity sebagai mesin permainan utama. Game ini dirancang sebagai prototipe penelitian yang menggabungkan elemen eksplorasi dunia mimpi dengan muatan edukatif berbasis nilai-nilai spiritual Islami. Pengembangan game melibatkan perancangan level yang terdiri dari environment bertema gelap dan misterius, sistem pergerakan karakter berbasis kontrol keyboard dan mouse, serta mekanika pengambilan kunci sebagai inti dari progres permainan.

Pada implementasinya, game terdiri dari tiga level utama yang masing-masing memiliki tingkat kompleksitas berbeda. Setiap level dirancang dengan tata letak objek dan distribusi kunci yang diacak secara prosedural agar setiap sesi permainan memberikan tantangan yang variatif. Sistem pencatatan data performa

pemain diintegrasikan langsung ke dalam mekanisme permainan sehingga data skor, damage yang diterima, waktu bermain, dan jumlah nyawa tersisa dapat direkam secara *real-time* selama sesi bermain berlangsung. Data inilah yang menjadi fondasi utama bagi sistem adaptif untuk melakukan evaluasi dan menghasilkan rekomendasi sepatu pada akhir setiap level.

Antarmuka pengguna diimplementasikan secara modular, mencakup layar utama (*main menu*), layar tutorial, tampilan *gameplay* dengan minimap, tampilan *Paper-Hint* saat kunci ditemukan, serta menu pilihan sepatu yang menjadi titik interaksi utama antara pemain dan sistem rekomendasi. Seluruh elemen antarmuka dirancang agar intuitif dan informatif, sehingga pemain dapat memahami konteks rekomendasi yang diberikan oleh sistem tanpa harus memahami mekanisme teknis di baliknya.

a. *Main Menu*

Main menu berfungsi sebagai gerbang utama player untuk memulai game. Tampilan pada gambar 4.1 di bawah menampilkan empat tombol utama, Pertama start untuk memulai permainan baru, kedua *tutorial* untuk melihat panduan dasar *gameplay*, ketiga *option* untuk mengatur suara dalam game, resolution dan lainnya, dan terakhir tombol exit kegunaanya untuk keluar dari game. Desain visual main menu menggunakan latar belakang yang mencerminkan game *The Night Dreamers* di mana pemain berada dalam dunia mimpi yang bercorak gelap dan sunyi.

Gambar 4. 1 *Main Menu*

b. *Tutorial Screen*

Gambar 4.2 di bawah menunjukkan halaman *tutorial* berisi panduan dasar mengenai mekanisme permainan, seperti penggunaan *mouse click left* untuk menyerang *enemy*, tombol W-A-S-D untuk pergerakan *player*, mouse untuk mengatur arah pandang, tombol *space* untuk pemain melompat, dan tombol *left shift* untuk pemain berlari dalam permainan. Desain visual *tutorial screen* dibuat untuk memudahkan pemain untuk mengetahui mekanisme cara bermain dalam game *The Night Dreamers*.

Gambar 4. 2 *Tutorial Screen*

c. *Gameplay dan Minimap*

Pada gambar 4.3 di bawah, pemain mulai melakukan eksplorasi. Entitas musuh muncul selama eksplorasi dan menyerang karakter pemain. Setelah pemain berhasil mengalahkan semua musuh, kunci akan muncul untuk syarat memasuki gerbang menuju ke level selanjutnya. Selama permainan berlangsung, sistem juga mulai mengumpulkan data performa pemain berupa *score*, *damage taken*, *time*, dan *lives left* yang digunakan sebagai *input* dalam sistem rekomendasi.



Gambar 4. 3 *Gameplay dan Minimap*

d. *Game Over*

Gambar 4.4 di bawah menampilkan layar *game over* yang muncul ketika nyawa pemain habis. Pada kondisi ini, sistem menghentikan seluruh proses pelacakan data performa secara otomatis dan menampilkan pesan naratif. Layar ini menyediakan tombol untuk mengulang permainan dari awal (*restart*) dan tombol untuk kembali ke main menu.

Gambar 4. 4 *Game Over*

e. *Shoe Recommendation*

Pada tahap ini, sistem melakukan beberapa proses yang mana hasil dari proses tersebut ditampilkan dalam bentuk rekomendasi sepatu yang dapat dipilih oleh pemain. Proses rekomendasi dilakukan berdasarkan capaian pemain dari stage sebelumnya. Pemilihan sepatu pada tahap ini akan mempengaruhi performa pemain pada level berikutnya, sehingga berperan dalam mendukung strategi permainan secara adaptif.

Gambar 4. 5 *Shoe Recommendation*

f. Level Complete

Gambar 4.6 di bawah menampilkan layar level complete yang muncul ketika pemain berhasil menyelesaikan seluruh level. Pada tahap ini, sistem merekap seluruh data capaian pemain yang telah dikumpulkan sepanjang level tersebut. Layar ini juga menampilkan ringkasan statistik performa pemain sebagai umpan balik langsung atas keberhasilan pemain.



Gambar 4. 6 Game Over

4.1.2 Implementasi Sistem Adaptif Firefly-TOPSIS

Sistem adaptif diimplementasikan sebagai modul terpisah yang berjalan di dalam lingkungan Unity menggunakan bahasa pemrograman C#. Modul ini terdiri dari tiga komponen utama, yaitu komponen pengumpulan dan normalisasi data capaian pemain, komponen optimasi bobot kriteria menggunakan *Firefly Algorithm*, serta komponen pemeringkatan alternatif menggunakan metode TOPSIS. Ketiga komponen tersebut bekerja secara sekuensial dan dipanggil secara otomatis pada saat pemain menyelesaikan satu level permainan.

1. Capaian Pemain

Tahap pertama dalam rantai sistem adaptif adalah menghentikan proses pelacakan dan mengambil data performa pemain yang telah terakumulasi selama satu sesi level. Proses ini diimplementasikan melalui fungsi `StopAndGetResult()` yang dipanggil secara otomatis ketika pemain menyelesaikan sebuah level.

Pseudocode 4.1 Data Capaian Pemain

```

FUNCTION StopAndGetResult()
    // Hentikan pelacakan
    SET _isTracking TO false

    // Jika tidak ada data performa, catat error dan kembalikan null
    IF _currentPerformance IS null THEN
        PRINT error "[PlayerPerformanceTracker] No performance data!"
        RETURN null
    END IF

    // Perbarui data terakhir dari variabel yang sedang dilacak
    SET _currentPerformance.score TO currentScore
    SET _currentPerformance.damageTaken TO currentDamage
    SET _currentPerformance.playTime TO currentPlayTime
    SET _currentPerformance.livesLeft TO currentLives

    // Hitung nilai Fcp (Financial Cost of Performance) dari performa
    saat ini
    COMPUTE fcp = _currentPerformance.CalculateFcp()

    // Kembalikan objek performa yang telah diperbarui
    RETURN _currentPerformance
END FUNCTION

```

Pada fungsi tersebut, langkah awal yang dilakukan adalah menonaktifkan *flag_is Tracking* agar sistem tidak lagi merekam data baru setelah level dinyatakan selesai. Selanjutnya, sistem memeriksa apakah objek `_currentPerformance` telah terisi. Apabila objek tersebut kosong atau bernilai *null*, sistem mencatat pesan *error* dan menghentikan eksekusi dengan mengembalikan nilai *null*. Jika data tersedia, sistem memperbarui keempat atribut utama pada objek performa, yaitu *score*, *damageTaken*, *playTime*, dan *livesLeft*, dengan nilai terkini dari variabel-variabel

yang telah direkam selama permainan berlangsung. Sebelum data dikembalikan, fungsi `CalculateFcp()` dipanggil untuk menghitung nilai Faktor Capaian Pemain (*Fcp*) yang merepresentasikan kualitas performa pemain secara agregat. Nilai *Fcp* inilah yang kemudian menjadi masukan utama bagi *Firefly Algorithm* pada tahap berikutnya.

Sebelum capaian pemain dihitung, sistem menginisialisasi pustaka sepatu sebagai matriks keputusan awal (*Decision Matrix*). Proses ini dijalankan pada metode `InitializedData()` melalui pustaka `ShoeDatabase.Instance`.

Pseudocode 4.2 Inisialisasi Matriks Keputusan

```
PROCEDURE InitializeData()
    // Ambil semua data sepatu dari database
    SET _allShoes TO ShoeDatabase.Instance.GetAllShoes()

    // Ambil matriks keputusan dari database
    SET _decisionMatrix TO ShoeDatabase.Instance.GetDecisionMatrix()

    // Buat array untuk menyimpan ID alternatif berdasarkan jumlah
    // sepatu
    SET _alternativeIds TO array baru dengan ukuran _allShoes.Count

    // Looping untuk mengisi setiap ID sepatu ke dalam array
    FOR i DARI 0 KE (_allShoes.Count - 1) DO
        SET _alternativeIds[i] TO _allShoes[i].shoeId
    END FOR
END PROCEDURE
```

Prosedur `InitializeData()` mengambil daftar lengkap sepatu melalui `ShoeDatabase.Instance.GetAllShoes()` dan menyimpan matriks keputusan awal melalui `ShoeDatabase.Instance.GetDecisionMatrix()`. Matriks keputusan ini berisi nilai setiap alternatif sepatu terhadap keenam kriteria yang telah ditetapkan, sebagaimana yang tercantum pada Tabel 4.1. Selanjutnya, sistem membuat array `_alternativeIds` berukuran sejumlah sepatu yang tersedia, kemudian mengisi setiap elemennya dengan ID unik masing-masing sepatu melalui iterasi *for loop*.

Matriks `_decisionMatrix` ini bersifat statis untuk menjaga objektivitas evaluasi spesifikasi perangkat, sebelum akhirnya dibobotkan secara dinamis oleh capaian pemain.

Setelah data capaian pemain diperoleh dan matriks keputusan diinisialisasi, seluruh nilai parameter performa perlu diseragamkan terlebih dahulu ke dalam rentang 0 hingga 1 agar perbedaan satuan dan skala antar parameter tidak mendistorsi hasil perhitungan. Proses ini dijalankan melalui prosedur `Normalize()` dan fungsi `CalculateFcp()`.

Pseudocode 4.3 Normalisasi Data Capaian Pemain (Fcp)

```

PROCEDURE Normalize()
    // Score: lebih tinggi lebih baik → normalisasi langsung
    IF maxScore > 0 THEN
        normalizedScore = CLAMP(score / maxScore, 0, 1)
    ELSE
        normalizedScore = 0
    END IF

    // Damage: lebih rendah lebih baik → 1 - (damage/maxDamage)
    IF maxDamage > 0 THEN
        normalizedDamage = 1 - CLAMP(damageTaken / maxDamage, 0, 1)
    ELSE
        normalizedDamage = 1
    END IF

    // PlayTime: lebih rendah lebih baik → 1 - (playTime/maxPlayTime)
    IF maxPlayTime > 0 THEN
        normalizedPlayTime = 1 - CLAMP(playTime / maxPlayTime, 0, 1)
    ELSE
        normalizedPlayTime = 0
    END IF

    // LivesLeft: lebih tinggi lebih baik → normalisasi langsung
    IF maxLives > 0 THEN
        normalizedLives = CLAMP(livesLeft / maxLives, 0, 1)
    ELSE
        normalizedLives = 0
    END IF
END PROCEDURE

FUNCTION CalculateFcp() RETURNS float
    CALL Normalize()

    // Hitung bobot linear
    fcp = (alpha1 * normalizedScore)

```

```

    + (alpha2 * normalizedDamage)
    + (alpha3 * normalizedPlayTime)
    + (alpha4 * normalizedLives)

    // Jamin nilai dalam rentang [0,1]
    fcp = CLAMP(fcp, 0, 1)

    RETURN fcp
END FUNCTION

```

Prosedur `Normalize()` menerapkan dua pendekatan normalisasi yang berbeda tergantung pada sifat setiap parameter. Untuk parameter yang bersifat *benefit* (semakin tinggi semakin baik), seperti `score` dan `livesLeft`, normalisasi dilakukan secara langsung dengan membagi nilai aktual terhadap nilai maksimum referensi. Sebaliknya, untuk parameter yang bersifat *cost* (semakin rendah semakin baik), seperti `damageTaken` dan `playTime`, normalisasi dilakukan dengan formula inversif, yakni satu dikurangi hasil pembagian nilai aktual dengan nilai maksimumnya. Setelah keempat nilai normalisasi diperoleh, fungsi `CalculateFcp()` menggabungkannya menjadi satu nilai skalar menggunakan formula rata-rata berbobot: $Fcp = (\alpha_1 \times S_norm) + (\alpha_2 \times D_norm) + (\alpha_3 \times T_norm) + (\alpha_4 \times L_norm)$, dengan bobot parameter $\alpha_1 = 0.3$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, dan $\alpha_4 = 0.2$ sebagaimana ditetapkan pada Bab III (Tabel 3.15). Nilai akhir Fcp dijamin berada dalam rentang $[0,1]$ melalui fungsi *clamp*.

Pengumpulan data dilakukan melalui *event listener* yang terpasang pada objek-objek kritis dalam game, seperti objek musuh, mekanisme damage, timer level, dan sistem nyawa. Setiap interaksi yang relevan dicatat dalam sebuah variabel global yang dikelola oleh *game manager*. Ketika level selesai, data yang terkumpul dikirimkan ke modul sistem adaptif untuk diproses lebih lanjut. Proses normalisasi

dilakukan menggunakan metode *min-max normalization* untuk mengubah setiap nilai parameter ke dalam rentang 0 hingga 1, sehingga perbedaan skala antar parameter tidak mempengaruhi hasil perhitungan.

2. *Firefly*

Setelah nilai *Fcp* dan matriks keputusan tersedia, tahap berikutnya adalah membentuk populasi awal *Firefly Algorithm*. Setiap individu *firefly* merepresentasikan satu kandidat solusi berupa vektor bobot kriteria, dan populasi awal dibentuk melalui fungsi `GenerateRandomWeights()`.

Pseudocode 4.4 Inisialisasi Populasi Firefly

```

FUNCTION GenerateRandomWeights() RETURNS array of float
  // Tentukan jumlah kriteria
  SET n TO ShoeCriteria.TOTAL_CRITERIA

  // Buat array bobot dengan panjang n
  SET weights TO array baru berukuran n
  SET total TO 0.0

  // Loop untuk menghasilkan bobot acak untuk setiap kriteria
  FOR j FROM 0 TO n-1 DO
    SET weights[j] TO Random.Range(_params.WeightMin,
    _params.WeightMax)
    ADD weights[j] TO total
  END FOR

  // Normalisasi agar total bobot sama dengan 1
  IF total > 0 THEN
    FOR j FROM 0 TO n-1 DO
      SET weights[j] TO weights[j] / total
    END FOR
  END IF

  RETURN weights
END FUNCTION

```

Fungsi ini menentukan dimensi vektor berdasarkan jumlah total kriteria (`ShoeCriteria.TOTAL_CRITERIA`), yaitu enam kriteria dalam penelitian ini. Setiap elemen bobot diisi dengan nilai acak dalam rentang `[WeightMin, WeightMax]` menggunakan `Random.Range()`, sambil mengakumulasikan jumlah total bobot.

Setelah seluruh elemen terisi, normalisasi dilakukan dengan membagi setiap elemen dengan total keseluruhan, sehingga jumlah semua bobot tepat sama dengan satu. Kondisi ini memenuhi syarat normalisasi bobot dalam TOPSIS. Proses ini diulang sebanyak jumlah populasi yang ditetapkan, yaitu 10 *firefly*, untuk membentuk populasi awal yang beragam sebelum iterasi optimasi dimulai.

Setelah populasi awal terbentuk, sistem perlu menghitung seberapa kuat satu *firefly* tertarik untuk bergerak mendekati *firefly* lain yang lebih terang. Mekanisme ini diimplementasikan melalui fungsi `CalculateAttractiveness()`.

Pseudocode 4.5 Attractiveness

```
FUNCTION CalculateAttractiveness(distance) RETURNS float
    // Menghitung nilai daya tarik berdasarkan jarak
    // Beta0 adalah konstanta awal, Gamma adalah koefisien peluruhan
    RETURN _params.Beta0 * EXP(-_params.Gamma * distance * distance)
END FUNCTION
```

Fungsi ini mengimplementasikan persamaan daya tarik berbasis jarak *Euclidean* menggunakan formula eksponensial *Gaussian* sesuai Persamaan 2.12, yaitu $\beta(r) = \beta_0 \times e^{(-\gamma \times r^2)}$. Dalam implementasi ini, `Beta0` adalah konstanta daya tarik awal yang ditetapkan sebesar 1, dan `Gamma` adalah koefisien penyerapan cahaya sebesar 1 yang mengatur seberapa cepat daya tarik melemah seiring bertambahnya jarak r antara dua *firefly*. Semakin jauh jarak antara dua *firefly*, nilai eksponensial semakin mendekati nol sehingga daya tarik pun melemah. Mekanisme ini memastikan hanya *firefly* yang relatif berdekatan dalam ruang solusi yang saling mempengaruhi secara signifikan.

Setelah nilai daya tarik antar dua *firefly* diperoleh, sistem menjalankan pembaruan posisi *firefly* yang lebih redup agar bergerak menuju *firefly* yang lebih terang. Proses ini diimplementasikan melalui prosedur `MoveFirefly()`.

Pseudocode 4.6 Pergerakan *Firefly*

```

PROCEDURE MoveFirefly(firefly, snapI, snapJ)
  // Hitung jarak Euclidean antara dua kunang-kunang
  SET r_ij TO EuclideanDistance(snapI.Position, snapJ.Position)

  // Hitung daya tarik berdasarkan jarak
  SET beta_ij TO CalculateAttractiveness(r_ij)  // Persamaan 2.12

  // Inisialisasi array untuk bobot baru
  SET newWeights TO array baru berukuran n

  // Untuk setiap dimensi (kriteria)
  FOR k FROM 0 TO n-1 DO
    // Hitung komponen gerakan akibat daya tarik
    SET movement TO beta_ij * (snapJ.Position[k] -
snapI.Position[k])

    // Hitung faktor acak (random walk)
    SET randomFactor TO _params.Alpha * (Random.value - 0.5)

    // Hitung posisi baru berdasarkan Persamaan 2.14
    SET newWeights[k] TO snapI.Position[k] + movement +
randomFactor
  END FOR

  // Normalisasi ulang bobot agar total = 1
  SET firefly.Position TO NormalizeWeights(newWeights)
END PROCEDURE

```

Prosedur ini pertama-tama menghitung jarak *Euclidean* r_{ij} antara posisi dua *firefly*, kemudian menghitung daya tarik β_{ij} berdasarkan jarak tersebut menggunakan Persamaan 2.12. Selanjutnya, untuk setiap dimensi (kriteria), posisi baru dihitung berdasarkan Persamaan 2.14 yang terdiri dari tiga komponen: posisi awal *firefly* itu sendiri, komponen gerakan akibat daya tarik (β_{ij}) dikalikan selisih posisi antara *firefly* sumber dan target, dan faktor acak (*random walk*) yang dihitung sebagai $\text{Alpha} \times (\text{Random.value} - 0.5)$. Ketiga komponen ini dijumlahkan untuk menghasilkan posisi baru pada setiap dimensi, yang kemudian dinormalisasi ulang melalui `NormalizeWeights()` agar tetap memenuhi syarat jumlah bobot sama dengan satu.

Firefly Algorithm diimplementasikan dengan parameter yang telah ditentukan pada Bab III (Tabel 3.12 Parameter *Firefly Algorithm*), yaitu jumlah *firefly* sebanyak 10 individu, iterasi maksimum sebanyak 20 iterasi, daya tarik awal (β_0) sebesar 1, koefisien penyerapan cahaya (γ) sebesar 1, dan faktor randomisasi (α) sebesar 0.2. Setiap *firefly* direpresentasikan sebagai vektor bobot kriteria berdimensi enam yang memenuhi kondisi normalisasi (jumlah total bobot sama dengan satu). Proses iterasi dilakukan secara berulang hingga mencapai jumlah iterasi maksimum, di mana pada setiap iterasi dilakukan evaluasi *fitness* dan pembaruan posisi *firefly* berdasarkan Persamaan 2.6 yang telah dideskripsikan pada Bab II.

Setelah posisi setiap *firefly* diperbarui pada setiap iterasi, sistem perlu mengevaluasi seberapa baik kualitas bobot yang direpresentasikan oleh posisi tersebut. Evaluasi ini dilakukan melalui prosedur `EvaluateFitness()`.

Pseudocode 4. 7 Fungsi Fitness

```

PROCEDURE EvaluateFitness(firefly, fcp)
  // Dapatkan nilai preferensi maksimum dari evaluator TOPSIS
  SET vmax TO _topsisEvaluator.GetMaxPreferenceValue(firefly.Position)

  // Hitung fitness sebagai produk dari vmax dan fcp (Persamaan 3.4)
  SET firefly.Fitness TO vmax * fcp

  // Kecerahan kunang-kunang sama dengan nilai fitness (Persamaan 2.11)
  SET firefly.Brightness TO firefly.Fitness
END PROCEDURE

```

Pada prosedur ini, yang pertama-tama dipanggil adalah `_topsisEvaluator.GetMaxPreferenceValue(firefly.Position)`, dimana ini untuk memperoleh nilai preferensi tertinggi (V_{max}) yang dihasilkan oleh TOPSIS ketika

menggunakan vektor bobot milik *firefly* yang bersangkutan sebagai masukan. Nilai *fitness* kemudian dihitung sebagai produk dari V_{max} dan F_{cp} sesuai Persamaan 3.4, dan nilai ini sekaligus ditetapkan sebagai nilai kecerahan (*brightness*) *firefly* sesuai Persamaan 2.11.

Nilai *fitness* setiap *firefly* dihitung berdasarkan kombinasi nilai preferensi tertinggi yang dihasilkan oleh metode TOPSIS (V_{max}) dan faktor capaian pemain (F_{cp}). Dengan demikian, kualitas bobot yang dihasilkan tidak hanya bergantung pada kemampuan matematis TOPSIS dalam membedakan alternatif, tetapi juga mempertimbangkan kondisi aktual performa pemain. Pendekatan ini memastikan bahwa sistem tidak hanya mengoptimalkan bobot secara matematis, melainkan juga menghasilkan rekomendasi yang relevan dan kontekstual terhadap kondisi pemain.

3. TOPSIS

Setelah bobot optimal diperoleh dari hasil iterasi *Firefly Algorithm*, tahap selanjutnya adalah menjalankan metode TOPSIS untuk memeringkatkan sepuluh alternatif sepatu. Sebelum kalkulasi dijalankan, komponen TOPSIS memerlukan inisialisasi data yang terpisah dari komponen *Firefly*, sebagaimana diimplementasikan pada prosedur `InitializeData()` berikut.

Pseudocode 4. 8 Input Kriteria dan Alternatif

```
PROCEDURE InitializeData()
    // Jika instance database belum ada, buat dan inisialisasi secara manual
    IF ShoeDatabase.Instance IS null THEN
        CREATE GameObject dengan nama "ShoeDatabase"
        TAMBAHKAN komponen ShoeDatabase ke objek tersebut, simpan sebagai db
        PANGGIL DontDestroyOnLoad(db)

        // Panggil metode private InitializeShoes menggunakan refleksi
        method = DAPATKAN metode "InitializeShoes" dari kelas ShoeDatabase (non-public, instance)
        IF method TIDAK null THEN
```

```

        PANGGIL method.Invoke(db, null)
    END IF
END IF

    // Ambil semua data sepatu dan matriks keputusan dari instance
database
    SET _allShoes TO ShoeDatabase.Instance.GetAllShoes()
    SET _decisionMatrix TO ShoeDatabase.Instance.GetDecisionMatrix()

    // Buat array alternatif ID berdasarkan jumlah sepatu
    SET _alternativeIds TO array baru dengan panjang _allShoes.Count

    FOR i DARI 0 KE (_allShoes.Count - 1) DO
        SET _alternativeIds[i] TO _allShoes[i].shoeId
    END FOR
END PROCEDURE

```

Prosedur ini terlebih dahulu memeriksa ketersediaan *instance* ShoeDatabase. Apabila *instance* belum tersedia, sistem akan membuat objek ShoeDatabase baru secara dinamis menggunakan *reflection* untuk memanggil metode privat InitializeShoes(), kemudian memastikan objek tersebut bertahan lintas *scene* melalui DontDestroyOnLoad(). Setelah *instance* dipastikan tersedia, sistem mengambil data sepatu dan matriks keputusan, lalu membangun array *_alternativeIds* berisi ID seluruh sepatu yang menjadi alternatif dalam proses pemeringkatan.

Setelah data alternatif dan matriks keputusan siap, tahap pertama kalkulasi TOPSIS adalah menormalisasi matriks keputusan menggunakan metode *vector normalization*. Proses ini diimplementasikan melalui fungsi Step2_NormalizeMatrix().

Pseudocode 4.9 Normalisasi Matriks Keputusan TOPSIS

```

FUNCTION Step2_NormalizeMatrix() RETURNS matriks 2D float
    // Inisialisasi matriks R dengan ukuran [jumlahAlternatif,
jumlahKriteria]
    SET R TO matriks baru berukuran _numAlternatif x _numKriteria

    // Untuk setiap kolom (kriteria)
    FOR j DARI 0 KE (_numKriteria - 1) DO

```

```

        SET sumSquared TO 0

        // Hitung jumlah kuadrat dari semua nilai pada kolom j
        FOR i DARI 0 KE (_numAlternatif - 1) DO
            ADD (_decisionMatrix[i, j] * _decisionMatrix[i, j]) KE
sumSquared
        END FOR

        // Hitung akar dari sumSquared (norm Euclidean)
        SET denominator TO SQRT(sumSquared)

        // Normalisasi setiap elemen pada kolom j
        FOR i DARI 0 KE (_numAlternatif - 1) DO
            IF denominator > 0 THEN
                SET R[i, j] TO _decisionMatrix[i, j] / denominator
            ELSE
                SET R[i, j] TO 0
            END IF
        END FOR
    END FOR

    RETURN R
END FUNCTION

```

Fungsi ini memproses matriks keputusan kolom per kolom, di mana setiap kolom merepresentasikan satu kriteria. Untuk setiap kolom, sistem menghitung jumlah kuadrat dari seluruh nilai pada kolom tersebut, kemudian menghitung akar kuadratnya sebagai nilai *norm Euclidean* kolom (*denominator*). Setiap elemen pada kolom kemudian dibagi dengan *denominator* tersebut untuk menghasilkan nilai ternormalisasi. Apabila *denominator* bernilai nol, sistem memberikan nilai normalisasi nol untuk mencegah pembagian dengan nol. Hasil dari prosedur ini adalah matriks *R* berukuran [jumlah alternatif $\times$ jumlah kriteria] yang berisi nilai-nilai yang terdistribusi secara proporsional.

Setelah matriks dinormalisasi, tahap berikutnya adalah mengintegrasikan bobot kriteria hasil optimasi *Firefly Algorithm* ke dalam matriks tersebut. Proses ini diimplementasikan melalui fungsi `Step3_ApplyWeights()`.

Pseudocode 4. 10 Matriks Ternormalisasi Terbobot

```

FUNCTION Step3_ApplyWeights(weights) RETURNS matriks 2D float
  // Inisialisasi matriks V dengan ukuran [jumlahAlternatif,
  jumlahKriteria]
  SET V TO matriks baru berukuran _numAlternatif x _numKriteria

  // Untuk setiap alternatif
  FOR i DARI 0 KE (_numAlternatif - 1) DO
    // Untuk setiap kriteria
    FOR j DARI 0 KE (_numKriteria - 1) DO
      // Kalikan bobot dengan nilai matriks ternormalisasi
      SET V[i, j] TO weights[j] * _normalizedMatrix[i, j]
    END FOR
  END FOR

  RETURN V
END FUNCTION

```

Fungsi ini menerima vektor bobot optimal dari Firefly sebagai argumen, kemudian mengalikan setiap elemen matriks ternormalisasi dengan bobot kriteria yang bersesuaian melalui iterasi dua lapis, yaitu iterasi luar menelusuri setiap alternatif, dan iterasi dalam menelusuri setiap kriteria. Pada setiap sel matriks, nilai dinormalisasi $_normalizedMatrix[i, j]$ dikalikan dengan bobot $weights[j]$ untuk menghasilkan nilai terbobot $V[i, j]$. Matriks V yang dihasilkan merepresentasikan kontribusi masing-masing alternatif pada setiap kriteria setelah mempertimbangkan tingkat kepentingan relatif setiap kriteria berdasarkan kondisi aktual pemain.

Setelah matriks ternormalisasi terbobot diperoleh, sistem menentukan dua titik referensi utama dalam TOPSIS, yaitu solusi ideal positif (A^+) dan solusi ideal negatif (A^-). Proses ini diimplementasikan melalui prosedur `Step4_ComputeIdealSolutions()`.

Pseudocode 4. 11 Solusi Ideal Positif dan Negatif

```

PROCEDURE Step4_ComputeIdealSolutions(OUT idealPos, OUT idealNeg)
  // Inisialisasi array solusi ideal positif dan negatif
  SET idealPos TO array baru berukuran _numCriteria
  SET idealNeg TO array baru berukuran _numCriteria

  // Untuk setiap kriteria

```

```

FOR j DARI 0 KE (_numCriteria - 1) DO
  SET minVal KE nilai terbesar (misal:  $+\infty$ )
  SET maxVal KE nilai terkecil (misal:  $-\infty$ )

  // Cari nilai minimum dan maksimum pada kolom matriks terbobot
  FOR i DARI 0 KE (_numAlternatives - 1) DO
    JIKA _weightedMatrix[i, j] < minVal MAKA
      minVal = _weightedMatrix[i, j]
    AKHIR JIKA
    JIKA _weightedMatrix[i, j] > maxVal MAKA
      maxVal = _weightedMatrix[i, j]
    AKHIR JIKA
  END FOR

  // Tentukan ideal berdasarkan tipe kriteria (benefit atau
cost)
  JIKA _isBenefit[j] MAKA
    // Kriteria benefit: semakin besar semakin baik
    idealPos[j] = maxVal
    idealNeg[j] = minVal
  SEBALIKNYA
    // Kriteria cost: semakin kecil semakin baik
    idealPos[j] = minVal
    idealNeg[j] = maxVal
  AKHIR JIKA
END FOR
END PROCEDURE

```

Untuk setiap kriteria, sistem menelusuri seluruh nilai pada kolom matriks terbobot dan mengidentifikasi nilai minimum (*minVal*) serta maksimum (*maxVal*). Penetapan *idealPos* dan *idealNeg* kemudian bergantung pada tipe setiap kriteria yang tersimpan pada array *_isBenefit[]*. Untuk kriteria bersifat *benefit* (semakin besar semakin baik), *idealPos* ditetapkan sebagai nilai maksimum dan *idealNeg* sebagai nilai minimum. Sebaliknya, untuk kriteria bersifat *cost* (semakin kecil semakin baik), seperti C_1 (berat sepatu), *idealPos* ditetapkan sebagai nilai minimum dan *idealNeg* sebagai nilai maksimum.

Setelah solusi ideal positif dan negatif ditentukan, langkah selanjutnya adalah menghitung jarak *Euclidean* setiap alternatif terhadap kedua titik referensi

tersebut. Proses ini diimplementasikan melalui prosedur

Step5_ComputeDistances().

Pseudocode 4. 12 Jarak Euclidean

```

PROCEDURE Step5_ComputeDistances(OUT distPos, OUT distNeg)
  // Inisialisasi array jarak ke solusi ideal positif dan negatif
  SET distPos TO array baru berukuran _numAlternatives
  SET distNeg TO array baru berukuran _numAlternatives

  // Untuk setiap alternatif
  FOR i DARI 0 KE (_numAlternatives - 1) DO
    SET sumPos TO 0
    SET sumNeg TO 0

    // Hitung jumlah kuadrat selisih untuk setiap kriteria
    FOR j DARI 0 KE (_numCriteria - 1) DO
      SET diffPos TO _weightedMatrix[i, j] - _idealPositive[j]
      SET diffNeg TO _weightedMatrix[i, j] - _idealNegative[j]

      ADD (diffPos * diffPos) TO sumPos
      ADD (diffNeg * diffNeg) TO sumNeg
    END FOR

    // Jarak Euclidean = akar kuadrat dari jumlah kuadrat
    SET distPos[i] TO SQRT(sumPos)
    SET distNeg[i] TO SQRT(sumNeg)
  END FOR
END PROCEDURE

```

Untuk setiap alternatif, sistem menginisialisasi dua akumulator (sumPos dan sumNeg) yang kemudian diisi melalui iterasi atas seluruh kriteria. Pada setiap kriteria, selisih antara nilai terbobot alternatif dengan nilai solusi ideal positif dikuadratkan dan dijumlahkan ke sumPos, sementara selisih dengan nilai solusi ideal negatif dikuadratkan dan dijumlahkan ke sumNeg. Setelah semua kriteria diproses, akar kuadrat dari masing-masing akumulator dihitung untuk menghasilkan jarak *Euclidean* D^+ (distPos) dan D^- (distNeg) bagi alternatif tersebut.

Setelah jarak terhadap solusi ideal positif dan negatif diperoleh untuk setiap alternatif, tahap terakhir kalkulasi TOPSIS adalah menghitung nilai preferensi atau

closeness coefficient yang menjadi dasar pemeringkatan. Proses ini diimplementasikan melalui fungsi `Step6_ComputePreferenceValues()`.

Pseudocode 4. 13 Nilai Preferensi

```

FUNCTION Step6_ComputePreferenceValues() RETURNS array of float
  // Inisialisasi array nilai preferensi (closeness coefficient)
  SET ci TO array baru berukuran _numAlternatives

  // Untuk setiap alternatif
  FOR i DARI 0 KE (_numAlternatives - 1) DO
    // Hitung jarak total ke solusi ideal positif dan negatif
    SET total TO _distancePositive[i] + _distanceNegative[i]

    // Nilai preferensi = jarak negatif dibagi total jarak
    IF total > 0 THEN
      SET ci[i] TO _distanceNegative[i] / total
    ELSE
      SET ci[i] TO 0
    END IF
  END FOR

  RETURN ci
END FUNCTION

```

Fungsi ini menghitung nilai preferensi (C_i) bagi setiap alternatif dengan membagi jarak ke solusi ideal negatif (D^-) terhadap total jarak, yaitu penjumlahan D^+ dan D^- : $C_i = D^- / (D^+ + D^-)$. Nilai C_i yang mendekati 1 menandakan bahwa alternatif sangat dekat dengan solusi ideal positif sekaligus sangat jauh dari solusi ideal negatif, sementara nilai C_i mendekati 0 menunjukkan kondisi sebaliknya. Untuk mencegah pembagian dengan nol, sistem menetapkan $C_i = 0$ apabila total jarak bernilai nol. Seluruh nilai C_i yang dihasilkan kemudian diurutkan dari tertinggi ke terendah untuk menghasilkan peringkat akhir alternatif sepatu.

Setelah seluruh komponen, yaitu perhitungan F_{cp} , optimasi bobot *Firefly*, dan pemeringkatan TOPSIS, telah diimplementasikan secara terpisah, ketiganya diintegrasikan ke dalam satu alur kerja tunggal melalui fungsi

GenerateRecommendation() yang berperan sebagai orkestrator utama sistem adaptif.

Pseudocode 4. 14 Orkestrasi Pipeline (Integrasi *Firefly* + TOPSIS)

```

FUNCTION GenerateRecommendation(performance) RETURNS
RecommendationRecord
    // Hitung nilai FCP dari performa pemain (Persamaan 3.3)
    SET fcp TO performance.CalculateFcp()

    // Inisialisasi algoritma Firefly dengan matriks keputusan, tipe
    kriteria, dan parameter
    SET fa TO new FireflyAlgorithm(_decisionMatrix,
    ShoeCriteria.IsBenefit, _fireflyParams)

    // Jalankan optimasi Firefly untuk mendapatkan bobot optimal
    SET weights TO fa.Optimize(fcp)

    // Inisialisasi kalkulator TOPSIS dengan matriks keputusan, tipe
    kriteria, dan bobot
    SET topsis TO new TopsisCalculator(_decisionMatrix,
    ShoeCriteria.IsBenefit, ...)

    // Hitung hasil TOPSIS berdasarkan bobot yang diperoleh
    SET result TO topsis.Calculate(weights)

    // Bangun dan kembalikan record rekomendasi (berdasarkan result,
    fcp, dll.)
    RETURN ConstructRecommendationRecord(result, fcp, weights)
END FUNCTION

```

Proses dimulai dengan memanggil `performance.CalculateFcp()` untuk memperoleh nilai *Fcp* dari data capaian pemain (Persamaan 3.1). Selanjutnya, objek *Firefly Algorithm* diinisialisasi dengan matriks keputusan, informasi tipe kriteria, dan parameter algoritma yang telah ditetapkan, kemudian metode `Optimize(fcp)` dijalankan untuk menghasilkan vektor bobot optimal. Bobot optimal tersebut diserahkan kepada objek *TopsisCalculator* melalui metode `Calculate(weights)` yang menjalankan seluruh enam tahap kalkulasi TOPSIS secara sekuensial. Hasil pemeringkatan, nilai *Fcp*, dan vektor bobot final kemudian

dikemas ke dalam objek `RecommendationRecord` melalui fungsi `ConstructRecommendationRecord()`.

Setelah bobot optimal diperoleh dari *Firefly Algorithm*, metode TOPSIS dijalankan menggunakan bobot tersebut untuk melakukan pemeringkatan terhadap sepuluh alternatif sepatu yang telah didefinisikan. Hasil pemeringkatan berupa urutan alternatif dari nilai preferensi tertinggi hingga terendah, di mana alternatif dengan nilai tertinggi dipilih sebagai rekomendasi yang ditampilkan kepada pemain melalui menu pilihan sepatu. Pemain tetap memiliki kebebasan untuk menerima atau mengabaikan rekomendasi sistem, sehingga pengalaman bermain tetap bersifat interaktif dan tidak deterministik.

4.1.3 Implementasi Perpindahan Rekomendasi Antar Level

Proses adaptasi sistem dirancang untuk berjalan setelah setiap level permainan selesai. Ketika pemain menyelesaikan satu level, data capaian yang telah dikumpulkan selama level tersebut diproses oleh sistem adaptif untuk menghasilkan rekomendasi sepatu bagi level berikutnya. Dengan demikian, setiap perpindahan level menjadi momen evaluasi dan adaptasi yang memungkinkan sistem untuk terus memperbarui rekomendasinya sesuai perkembangan performa pemain.

Pada level awal (level 0), pemain tidak menggunakan sepatu khusus karena level ini berfungsi sebagai tahap pengambilan data baseline. Data yang dikumpulkan pada level 0 digunakan sebagai input pertama bagi sistem adaptif untuk menghasilkan rekomendasi pada level 1. Mekanisme ini dirancang untuk menghindari bias pada tahap awal adaptasi, sehingga sistem memiliki data referensi yang objektif sebelum mulai memberikan rekomendasi. Pendekatan ini selaras

dengan prinsip pengambilan data berbasis pengamatan yang dikemukakan pada desain sistem di Bab III, Sub-bab 3.4 Alur Sistem Adaptif *Firefly*-TOPSIS, Sub-bab 3.4.1 Mekanisme Transisi dan Perpindahan Level Sistem Adaptif.

4.2 Hasil Pengujian Sistem Adaptabilitas

Pengujian sistem dilakukan melalui tiga skenario percobaan yang merepresentasikan pola perilaku pemain yang berbeda-beda. Setiap percobaan menggunakan data capaian pemain yang berbeda sebagai input, sehingga menghasilkan bobot kriteria yang dinamis dan rekomendasi sepatu yang adaptif. Perbedaan antar percobaan dirancang untuk memvalidasi kemampuan sistem dalam merespons perubahan kondisi pemain secara konsisten dan tepat.

Sebelum melakukan perhitungan adaptif, sistem terlebih dahulu memuat data matriks keputusan yang berisi nilai setiap alternatif sepatu terhadap keenam kriteria yang telah ditentukan. Data ini bersifat tetap pada setiap percobaan karena merepresentasikan spesifikasi teknis objektif dari masing-masing produk sepatu, sebagaimana telah dijelaskan pada Bab III (Tabel 3.11). Tabel 4.1 berikut menampilkan nilai input untuk setiap alternatif pada masing-masing kriteria.

Tabel 4. 1 Nilai Input Kriteria Sepatu

No	Alternatif	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
1	A ₁	4	5	4	4	4	4
2	A ₂	3	4	4	5	4	5
3	A ₃	5	5	4	4	4	4
4	A ₄	3	4	4	4	5	5
5	A ₅	3	5	4	4	5	4
6	A ₆	5	3	4	3	3	3
7	A ₇	4	5	4	4	4	4
8	A ₈	5	5	5	5	4	4
9	A ₉	4	4	4	4	4	5
10	A ₁₀	4	4	4	5	4	5

4.2.1 Percobaan Pertama (Eksplorasi Aman)

Skenario percobaan pertama ini menyimulasikan karakteristik gaya bermain "Eksplorasi Aman", yaitu pola bermain pasif yang ditandai dengan gerakan sangat berhati-hati, minimnya konfrontasi langsung dengan entitas musuh (*Nightmares*), dan kecenderungan untuk lebih banyak mengobservasi lingkungan permainan. Data kuantitatif yang digunakan pada pengujian ini diambil dari log performa riil Responden 3 (R3) pada Stage 3. Rekapitulasi metrik capaian pemain tersebut disajikan pada Tabel 4.2.

Tabel 4. 2 Data Capaian Pemain Percobaan Pertama

No	Parameter	Nilai
1	<i>Score (S)</i>	60
2	<i>Damage taken (D)</i>	95
3	<i>Play time (T)</i>	86.040
4	<i>Lives left (L)</i>	3

Untuk mengeliminasi bias akibat perbedaan satuan pengukuran antar parameter, sistem melakukan normalisasi data menggunakan metode min-max. Hasil penyeragaman skala metrik tersebut ditunjukkan pada Tabel 4.3.

Tabel 4. 3 Normalisasi Data Pemain Percobaan Pertama

No	Parameter	Nilai Normalisasi
1	<i>Score (S)</i>	0.600
2	<i>Damage taken (D)</i>	0.050
3	<i>Play time (T)</i>	0.713
4	<i>Lives left (L)</i>	1.000

Data normalisasi tersebut selanjutnya diproses melalui fungsi linear terbobot untuk menghitung nilai Faktor Capaian Pemain (Fcp). Nilai Fcp ini merepresentasikan kualitas performa pemain secara holistik dan berfungsi sebagai komponen penentu nilai *fitness* pada populasi kunang-kunang. Hasil kalkulasi Fcp beserta nilai *Best Fitness* yang dicapai populasi ditunjukkan pada Tabel 4.4.

Tabel 4. 4 Data Fcp dan *Best Fitness* Percobaan Pertama

No	Parameter	Nilai
1	Faktor Capaian Pemain (Fcp)	0.538
2	<i>Best Fitness</i>	0.487

Melalui interaksi daya tarik antar individu kunang-kunang yang berlangsung selama 20 iterasi maksimum, diperoleh konfigurasi bobot kriteria yang adaptif terhadap kondisi performa pemain. Hasil pembobotan ini disajikan pada Tabel 4.5.

Tabel 4. 5 Bobot Kriteria Hasil Optimasi *Firefly* Percobaan Pertama

No	Kriteria	Bobot
1	C ₁ - Berat	0.028
2	C ₂ - Cushioning	0.359
3	C ₃ - Responsiveness	0.114
4	C ₄ - Traction	0.030
5	C ₅ - Stability	0.456
6	C ₆ - Durability	0.014

Berdasarkan Tabel 4.5, algoritma *Firefly* mengalokasikan bobot tertinggi pada kriteria C₅ (*Stability Support*) sebesar 0,456, diikuti oleh C₂ (*Cushioning Level*) sebesar 0,359. Distribusi bobot ini sejalan dengan karakteristik gaya bermain yang terdeteksi, yaitu pemain bergerak relatif lambat namun mampu menjaga jumlah nyawa hingga maksimal, sehingga sistem merespons dengan memprioritaskan kriteria stabilitas dan kenyamanan melangkah untuk menunjang keamanan navigasi pemain berhati-hati.

Vektor bobot optimal dari algoritma *Firefly* kemudian digunakan sebagai pengali pada matriks keputusan yang telah dinormalisasi secara vektor. Hasil normalisasi matriks ditunjukkan pada Tabel 4.6.

Tabel 4. 6 Normalisasi Matriks Keputusan Percobaan Pertama

No	Alternatif	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
1	A ₁	0.310	0.355	0.308	0.298	0.306	0.291
2	A ₂	0.233	0.284	0.308	0.373	0.306	0.364
3	A ₃	0.388	0.355	0.308	0.298	0.306	0.291
4	A ₄	0.233	0.284	0.308	0.298	0.382	0.364
5	A ₅	0.233	0.355	0.308	0.298	0.382	0.291
6	A ₆	0.388	0.213	0.308	0.224	0.229	0.218
7	A ₇	0.310	0.355	0.308	0.298	0.306	0.291
8	A ₈	0.388	0.355	0.385	0.373	0.306	0.291
9	A ₉	0.310	0.284	0.308	0.298	0.306	0.364
10	A ₁₀	0.310	0.284	0.308	0.373	0.306	0.364

Setelah matriks ternormalisasi dikalikan dengan bobot kriteria optimal, diperoleh Matriks Ternormalisasi Terbobot sebagaimana disajikan pada Tabel 4.7.

Tabel 4. 7 Matriks Ternormalisasi Terbobot Percobaan Pertama

No	Alternatif	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
1	A ₁	0.009	0.128	0.035	0.009	0.139	0.004
2	A ₂	0.006	0.102	0.035	0.011	0.139	0.005
3	A ₃	0.011	0.128	0.035	0.009	0.139	0.004
4	A ₄	0.006	0.102	0.035	0.009	0.174	0.005
5	A ₅	0.006	0.128	0.035	0.009	0.174	0.004
6	A ₆	0.011	0.077	0.035	0.007	0.105	0.003
7	A ₇	0.009	0.128	0.035	0.009	0.139	0.004
8	A ₈	0.011	0.128	0.044	0.011	0.139	0.004
9	A ₉	0.009	0.102	0.035	0.009	0.139	0.005
10	A ₁₀	0.009	0.102	0.035	0.011	0.139	0.005

Berdasarkan matriks terbobot tersebut, sistem menentukan koordinat Solusi Ideal Positif (A^+) dan Solusi Ideal Negatif (A^-) dengan memperlakukan C^1 sebagai kriteria biaya (*cost*) dan $C^2 - C^6$ sebagai kriteria keuntungan (*benefit*). Hasil kalkulasi solusi ideal dirangkum pada Tabel 4.8.

Tabel 4. 8 Solusi Ideal Positif dan Negatif Percobaan Pertama

No	Kriteria	A ⁺	A ⁻
1	C ₁	0.006	0.011
2	C ₂	0.128	0.077
3	C ₃	0.044	0.035
4	C ₄	0.011	0.007
5	C ₅	0.174	0.105
6	C ₆	0.005	0.003

Langkah selanjutnya adalah mengukur jarak *Euclidean* masing-masing alternatif sepatu terhadap titik A^+ dan A^- . Hasil pengukuran jarak ditunjukkan pada Tabel 4.9.

Tabel 4. 9 Jarak terhadap Solusi Ideal Percobaan Pertama

No	Alternatif	D+	D-
1	A ₁	0.036	0.062
2	A ₂	0.044	0.044
3	A ₃	0.036	0.062
4	A ₄	0.027	0.074
5	A ₅	0.009	0.087
6	A ₆	0.087	0.000
7	A ₇	0.036	0.062
8	A ₈	0.035	0.063
9	A ₉	0.044	0.043
10	A ₁₀	0.044	0.044

Melalui pembagian proporsional nilai jarak tersebut, diperoleh Nilai Preferensi (V_i) yang menjadi dasar penentuan peringkat akhir rekomendasi. Hasil pemeringkatan disajikan pada Tabel 4.10.

Tabel 4. 10 Nilai Preferensi dan Ranking Percobaan Pertama

No	Alternatif	V _i	Rank
1	A ₁	0.632	4
2	A ₂	0.498	7
3	A ₃	0.631	6
4	A ₄	0.733	2
5	A ₅	0.905	1
6	A ₆	0.000	10
7	A ₇	0.632	5
8	A ₈	0.641	3
9	A ₉	0.495	9
10	A ₁₀	0.497	8

Berdasarkan Tabel 4.10, alternatif A₅ (*Nike Structure 25*) menempati peringkat pertama dengan nilai preferensi tertinggi sebesar 0,905. Hasil ini konsisten dengan spesifikasi teknis produk, di mana *Nike Structure 25* memiliki teknologi midsole *Cushlon + Zoom Air* serta struktur stability system yang dirancang khusus untuk kategori stability running. Karakteristik fisik produk

tersebut tertangkap secara objektif oleh perhitungan TOPSIS terbobot, sehingga rekomendasi yang dihasilkan dapat dikatakan relevan untuk menunjang keamanan langkah pemain pada level berikutnya.

4.2.2 Percobaan Kedua (Eksplorasi Agresif)

Skenario percobaan kedua menyimulasikan pola perilaku pengguna "Eksplorasi Agresif", yang ditandai dengan manuver melompat tanpa arah (*random jumping*), gerakan konfrontasi yang impulsif, serta kesulitan membaca rintangan, sehingga karakter cenderung menerima akumulasi kerusakan fisik (*damage taken*) yang besar. Data kuantitatif pada pengujian ini diambil dari log performa Responden 1 (R1) pada Stage 3. Parameter capaian pemain disajikan pada Tabel 4.11.

Tabel 4. 11 Data Capaian Pemain Percobaan Kedua

No	Parameter	Nilai
1	<i>Score</i> (S)	40
2	<i>Damage taken</i> (D)	120
3	<i>Play time</i> (T)	89.686
4	<i>Lives left</i> (L)	2

Melalui proses normalisasi dengan inversi khusus untuk variabel bertipe cost (*damage taken* dan *play time*), diperoleh nilai ternormalisasi dalam rentang [0,1] sebagaimana ditunjukkan pada Tabel 4.12.

Tabel 4. 12 Normalisasi Data Pemain Percobaan Kedua

No	Parameter	Nilai Normalisasi
1	<i>Score</i> (S)	0.400
2	<i>Damage taken</i> (D)	0.000
3	<i>Play time</i> (T)	0.701
4	<i>Lives left</i> (L)	0.667

Melalui penjumlahan linear terbobot, sistem mengekstraksi parameter normalisasi menjadi nilai Faktor Capaian Pemain (Fcp). Nilai Fcp beserta *Best Fitness* yang dicapai populasi kunang-kunang disajikan pada Tabel 4.13.

Tabel 4. 13 Data Fcp dan *Best Fitness* Percobaan Kedua

No	Parameter	Nilai
1	Faktor Capaian Pemain	0.394
2	<i>Best Fitness</i>	0.389

Nilai Fcp yang relatif rendah (0,394) merefleksikan kondisi pertahanan karakter yang banyak terkuras, di mana hal ini mendorong algoritma *Firefly* melakukan iterasi optimasi untuk menyusun konfigurasi bobot kriteria baru. Hasil pembobotan tersebut disajikan pada Tabel 4.14.

Tabel 4. 14 Bobot Kriteria Hasil Optimasi *Firefly* Percobaan Kedua

No	Kriteria	Bobot
1	C ₁ - Berat	0.010
2	C ₂ - Cushioning	0.010
3	C ₃ - Responsiveness	0.010
4	C ₄ - Traction	0.557
5	C ₅ - Stability	0.010
6	C ₆ - Durability	0.405

Tabel 4.14 menunjukkan pergeseran bobot yang cukup signifikan dibanding percobaan pertama. *Algoritma Firefly* mengalokasikan bobot dominan pada kriteria C₄ (*Traction/Outsole Grip*) sebesar 0,557 dan C₆ (*Durability*) sebesar 0,405. Pergeseran ini mengindikasikan bahwa ketika performa pemain menurun akibat gaya bermain yang ceroboh, sistem cenderung mengalihkan prioritas dari aspek kenyamanan menuju aspek proteksi fisik, yaitu mengutamakan daya cengkeram tapak untuk menoleransi kegagalan manuver lompatan, serta ketahanan material agar tidak mudah mengalami penurunan performa akibat pemakaian yang intensif.

Vektor bobot yang berorientasi pada pertahanan fisik tersebut selanjutnya digunakan sebagai pengali pada matriks keputusan ternormalisasi. Matriks normalisasi disajikan kembali pada Tabel 4.15.

Tabel 4. 15 Normalisasi Matriks Keputusan Percobaan Kedua

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.310	0.355	0.308	0.298	0.306	0.291
2	A ₂	0.233	0.284	0.308	0.373	0.306	0.364
3	A ₃	0.388	0.355	0.308	0.298	0.306	0.291
4	A ₄	0.233	0.284	0.308	0.298	0.382	0.364
5	A ₅	0.233	0.355	0.308	0.298	0.382	0.291
6	A ₆	0.388	0.213	0.308	0.224	0.229	0.218
7	A ₇	0.310	0.355	0.308	0.298	0.306	0.291
8	A ₈	0.388	0.355	0.385	0.373	0.306	0.291
9	A ₉	0.310	0.284	0.308	0.298	0.306	0.364
10	A ₁₀	0.310	0.284	0.308	0.373	0.306	0.364

Hasil perkalian matriks tersebut menghasilkan Matriks Ternormalisasi Terbobot Percobaan Kedua sebagaimana dipaparkan pada Tabel 4.16.

Tabel 4. 16 Matriks Ternormalisasi Terbobot Percobaan Kedua

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.003	0.003	0.003	0.166	0.003	0.118
2	A ₂	0.002	0.003	0.003	0.208	0.003	0.147
3	A ₃	0.004	0.003	0.003	0.166	0.003	0.118
4	A ₄	0.002	0.003	0.003	0.166	0.004	0.147
5	A ₅	0.002	0.003	0.003	0.166	0.004	0.118
6	A ₆	0.004	0.002	0.003	0.125	0.002	0.088
7	A ₇	0.003	0.003	0.003	0.166	0.003	0.118
8	A ₈	0.004	0.003	0.004	0.208	0.003	0.118
9	A ₉	0.003	0.003	0.003	0.166	0.003	0.147
10	A ₁₀	0.003	0.003	0.003	0.208	0.003	0.147

Berdasarkan matriks terbobot tersebut, sistem menentukan koordinat Solusi Ideal Positif (A^+) dan Solusi Ideal Negatif (A^-) yang dirangkum pada Tabel 4.17.

Tabel 4. 17 Solusi Ideal Positif dan Negatif Percobaan Kedua

No	Kriteria	A ⁺	A ⁻
1	C ₁	0.002	0.004
2	C ₂	0.003	0.002
3	C ₃	0.004	0.003
4	C ₄	0.208	0.125
5	C ₅	0.004	0.002
6	C ₆	0.147	0.088

Kalkulasi jarak *Euclidean* dari sepuluh alternatif produk terhadap kedua titik solusi ideal dipaparkan pada Tabel 4.18.

Tabel 4. 18 Jarak terhadap Solusi Ideal Percobaan Kedua

No	Alternatif	D+	D-
1	A ₁	0.051	0.051
2	A ₂	0.001	0.102
3	A ₃	0.051	0.051
4	A ₄	0.042	0.072
5	A ₅	0.051	0.051
6	A ₆	0.102	0.000
7	A ₇	0.051	0.051
8	A ₈	0.029	0.088
9	A ₉	0.042	0.072
10	A ₁₀	0.001	0.102

Melalui pembagian proporsional nilai jarak tersebut, diperoleh Koefisien *Closeness Preferensi* (V_i) beserta urutan peringkat yang tertera pada Tabel 4.19.

Tabel 4. 19 Nilai Preferensi dan Ranking Percobaan Kedua

No	Alternatif	V _i	Rank
1	A ₁	0.500	7
2	A ₂	0.988	1
3	A ₃	0.500	9
4	A ₄	0.634	4
5	A ₅	0.500	6
6	A ₆	0.000	10
7	A ₇	0.500	8
8	A ₈	0.749	3
9	A ₉	0.634	5
10	A ₁₀	0.986	2

Berdasarkan Tabel 4.19, alternatif A₂ (*Adidas Ultraboost 22*) menempati peringkat pertama dengan nilai preferensi sebesar 0,988, diikuti secara ketat oleh A₁₀ (*Saucony Ride 16*) dengan skor 0,986. Hasil rekomendasi ini relevan secara fungsional, mengingat *Adidas Ultraboost 22* memiliki *outsole Stretchweb* yang mengintegrasikan material *Continental Rubber*, memberikan traksi cengkraman yang tinggi (skor C₄ = 5) untuk menoleransi pergerakan acak akibat gaya bermain agresif. Selain itu, material *upper Primeknit* serta konstruksi karet tapak yang padat

memberikan ketahanan fisik (*durability* maksimal, skor $C_6 = 5$), yang berguna untuk meredam degradasi performa item akibat pemakaian yang intensif pada level berikutnya.

4.2.3 Percobaan Ketiga (Penguasaan *Gameplay*)

Skenario percobaan ketiga merepresentasikan karakteristik pemain dengan tingkat kemahiran tertinggi, yaitu kelompok "Penguasaan *Gameplay*". Pemain pada kategori ini bergerak secara taktis, memahami tata letak objek kunci, mampu membaca pola musuh secara efisien, serta menyelesaikan permainan dalam durasi singkat (*speedrunning*) dengan tingkat cedera karakter yang mendekati nol. Data kuantitatif pada pengujian ini diambil dari log performa Responden 1 (R1) pada Stage 1. Parameter capaian pemain disajikan pada Tabel 4.20.

Tabel 4. 20 Data Capaian Pemain Percobaan Ketiga

No	Parameter	Nilai
1	<i>Score</i> (S)	30
2	<i>Damage taken</i> (D)	45
3	<i>Play time</i> (T)	36.761
4	<i>Lives left</i> (L)	3

Setelah dilakukan normalisasi untuk menyetarakan skala data, rekapitulasi parameter capaian ternormalisasi ditunjukkan pada Tabel 4.21.

Tabel 4. 21 Normalisasi Pemain Percobaan Ketiga

No	Parameter	Nilai Normalisasi
1	<i>Score</i> (S)	0.300
2	<i>Damage taken</i> (D)	0.550
3	<i>Play time</i> (T)	0.877
4	<i>Lives left</i> (L)	1.000

Nilai normalisasi performa di atas diolah untuk menghasilkan Faktor Capaian Pemain (Fcp). Nilai Fcp beserta *Best Fitness* populasi kunang-kunang disajikan pada Tabel 4.22.

Tabel 4. 22 Data Fcp dan *Best Fitness* Percobaan Ketiga

No	Parameter	Nilai
1	Faktor Capaian Pemain	0.630
2	<i>Best Fitness</i>	0.619

Dengan nilai Fcp yang tinggi (0,630), algoritma *Firefly* menjalankan iterasi pencarian untuk menyusun konfigurasi bobot kriteria yang sesuai dengan profil performa pemain. Hasil pembobotan disajikan pada Tabel 4.23.

Tabel 4. 23 Bobot Kriteria Hasil Optimasi *Firefly* Percobaan Ketiga

No	Kriteria	Bobot
1	C ₁	0.566
2	C ₂	0.376
3	C ₃	0.012
4	C ₄	0.012
5	C ₅	0.014
6	C ₆	0.020

Tabel 4.23 menunjukkan pergeseran bobot yang signifikan ke arah parameter kelincahan. Algoritma *Firefly* mengalokasikan bobot terbesar pada kriteria C₁ (Berat Sepatu) sebesar 0,566, diikuti oleh C₂ (*Cushioning Level*) sebesar 0,376. Pola ini cukup masuk akal jika dikaitkan dengan profil pemain, karena kontrol pergerakan dan kemampuan menghindar pemain kategori master sudah relatif optimal, sehingga kebutuhan terhadap fitur proteksi maupun penstabil tambahan menjadi lebih rendah. Oleh karena itu, sistem mengalihkan fokus pembobotan ke arah akselerasi kecepatan dengan memprioritaskan instrumen alas kaki yang ringan, guna mendukung efisiensi durasi eksplorasi.

Vektor bobot yang berorientasi pada kecepatan tersebut digunakan dalam perkalian matriks untuk mengevaluasi sepuluh alternatif sepatu. Matriks keputusan ternormalisasi disajikan kembali pada Tabel 4.24.

Tabel 4. 24 Normalisasi Matriks Keputusan Percobaan Ketiga

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.310	0.355	0.308	0.298	0.306	0.291
2	A ₂	0.233	0.284	0.308	0.373	0.306	0.364
3	A ₃	0.388	0.355	0.308	0.298	0.306	0.291
4	A ₄	0.233	0.284	0.308	0.298	0.382	0.364
5	A ₅	0.233	0.355	0.308	0.298	0.382	0.291
6	A ₆	0.388	0.213	0.308	0.224	0.229	0.218
7	A ₇	0.310	0.355	0.308	0.298	0.306	0.291
8	A ₈	0.388	0.355	0.385	0.373	0.306	0.291
9	A ₉	0.310	0.284	0.308	0.298	0.306	0.364
10	A ₁₀	0.310	0.284	0.308	0.373	0.306	0.364

Hasil perkalian matriks tersebut menghasilkan Matriks Ternormalisasi

Terbobot Percobaan Ketiga yang dipaparkan pada Tabel 4.25.

Tabel 4. 25 Matriks Ternormalisasi Terbobot Percobaan Ketiga

No	Alternatif	C1	C2	C3	C4	C5	C6
1	A ₁	0.176	0.134	0.004	0.004	0.004	0.006
2	A ₂	0.132	0.107	0.004	0.004	0.004	0.007
3	A ₃	0.220	0.134	0.004	0.004	0.004	0.006
4	A ₄	0.132	0.107	0.004	0.004	0.005	0.007
5	A ₅	0.132	0.134	0.004	0.004	0.005	0.006
6	A ₆	0.220	0.080	0.004	0.003	0.003	0.004
7	A ₇	0.176	0.134	0.004	0.004	0.004	0.006
8	A ₈	0.220	0.134	0.005	0.004	0.004	0.006
9	A ₉	0.176	0.107	0.004	0.004	0.004	0.007
10	A ₁₀	0.176	0.107	0.004	0.004	0.004	0.007

Berdasarkan koordinat ekstrem kriteria benefit dan cost pada matriks terbobot, ditetapkan batas Solusi Ideal Positif (A^+) dan Solusi Ideal Negatif (A^-) yang disajikan pada Tabel 4.26.

Tabel 4. 26 Solusi Ideal Positif dan Negatif Percobaan Ketiga

No	Kriteria	A ⁺	A ⁻
1	C ₁	0.132	0.220
2	C ₂	0.134	0.080
3	C ₃	0.005	0.004
4	C ₄	0.004	0.003
5	C ₅	0.005	0.003
6	C ₆	0.007	0.004

Kalkulasi jarak *Euclidean* dari sepuluh alternatif produk terhadap kedua titik solusi ideal dipaparkan pada Tabel 4.27.

Tabel 4. 27 Jarak terhadap Solusi Ideal Percobaan Ketiga

No	Alternatif	D+	D-
1	A ₁	0.044	0.069
2	A ₂	0.027	0.092
3	A ₃	0.088	0.054
4	A ₄	0.027	0.092
5	A ₅	0.002	0.103
6	A ₆	0.103	0.000
7	A ₇	0.044	0.069
8	A ₈	0.088	0.054
9	A ₉	0.051	0.052
10	A ₁₀	0.051	0.052

Melalui pembagian nilai kedekatan relatif tersebut, diperoleh *Koefisien Closeness Preferensi* (V_i) beserta urutan peringkat yang tertera pada Tabel 4.28.

Tabel 4. 28 Nilai Preferensi dan Ranking Percobaan Ketiga

No	Alternatif	V _i	Rank
1	A ₁	0.612	4
2	A ₂	0.774	3
3	A ₃	0.379	9
4	A ₄	0.774	2
5	A ₅	0.982	1
6	A ₆	0.000	10
7	A ₇	0.612	5
8	A ₈	0.379	8
9	A ₉	0.500	7
10	A ₁₀	0.501	6

Berdasarkan Tabel 4.28, alternatif A₅ (*Nike Structure 25*) kembali menempati peringkat pertama dengan nilai preferensi sebesar 0,982, diikuti oleh A₄ (*ASICS Gel-Nimbus 25*) dan A₂ (*Adidas Ultraboost 22*) yang masing-masing memperoleh nilai 0,774. Temuan ini cukup menarik untuk didiskusikan: meskipun bobot kriteria C₁ (Berat Sepatu) mendominasi secara signifikan, hasil akhir TOPSIS tetap mempertimbangkan kombinasi seluruh kriteria secara proporsional. Terpilihnya A₅ menunjukkan bahwa alternatif tersebut memiliki keseimbangan

terbaik antara bobot ringan dan dukungan stabilitas, yang relevan untuk menunjang kelincihan pergerakan pemain tingkat lanjut pada stage berikutnya.

4.3 Hasil Pengujian *Usability*

Pada pengujian ini, data dikumpulkan dari 15 pemain menggunakan kuesioner *System Usability Scale* (SUS). Pengujian ini berfokus pada aspek usability atau kemudahan penggunaan dari antarmuka pengguna (*user interface*) pada game *The Night Dreamers*. Melalui metode SUS, diharapkan dapat diperoleh gambaran yang lebih komprehensif mengenai tingkat kenyamanan, kemudahan navigasi, serta kualitas interaksi pemain dengan elemen-elemen visual dan fungsional yang ada dalam game. Hasil pengujian ini akan menjadi dasar dalam mengevaluasi kualitas antarmuka dan pengalaman pengguna secara keseluruhan.

Tabel 4. 29 Data Kuisisioner System Usability Scale

Responden	Skor Jawaban									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
R1	4	2	5	1	5	2	4	1	5	2
R2	4	2	4	2	4	2	4	2	4	2
R3	4	2	4	2	4	3	5	1	3	1
R4	4	2	5	1	5	1	4	2	5	1
R5	5	2	5	2	5	2	5	5	5	2
R6	3	3	3	2	2	3	5	3	4	3
R7	5	3	4	4	4	3	4	2	4	2
R8	4	2	5	1	4	2	5	2	4	2
R9	5	2	4	4	4	3	4	2	4	3
R10	4	2	4	2	4	2	4	2	5	2
R11	4	3	5	1	3	4	4	2	4	1
R12	4	3	4	2	4	3	4	2	4	2
R13	4	3	4	1	3	3	5	3	4	3
R14	4	1	5	1	4	2	5	2	5	1
R15	5	2	5	1	4	3	5	2	5	3

Data dari hasil kuesioner yang sudah dikumpulkan selanjutnya akan dihitung sesuai dengan aturan perhitungan skor SUS. Dalam perhitungan skor kuesioner ada beberapa aturan sebagai berikut.

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor responden akan dikurangi 1 ($X - 1$).
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari responden ($5 - X$).

Pada tahap ini, peraturan perhitungan nomor 1 dan 2 dilakukan. Tabel 4.30 menunjukkan hasil konversi nilai mentah responden sesuai dengan aturan perhitungan SUS, di mana setiap baris merepresentasikan total skor dasar.

Tabel 4. 30 Hasil Konversi Nilai Mentah *System Usability Scale*

Responden	Skor Jawaban										Total
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
R1	3	3	4	4	4	3	3	4	4	3	35
R2	3	3	3	3	3	3	3	3	3	3	30
R3	3	3	3	3	3	2	4	4	2	4	31
R4	3	3	4	4	4	4	3	3	4	4	36
R5	4	3	4	3	4	3	4	0	4	3	32
R6	2	2	2	3	1	2	4	2	3	2	23
R7	4	2	3	1	3	2	3	3	3	3	27
R8	3	3	4	4	3	3	4	3	3	3	33
R9	4	3	3	1	3	2	3	3	3	2	27
R10	3	3	3	3	3	3	3	3	4	3	31
R11	3	2	4	4	2	1	3	3	3	4	29
R12	3	2	3	3	3	2	3	3	3	3	28
R13	3	2	3	4	2	2	4	2	3	2	27
R14	3	4	4	4	3	3	4	3	4	4	36
R15	4	3	4	4	3	2	4	3	4	2	33

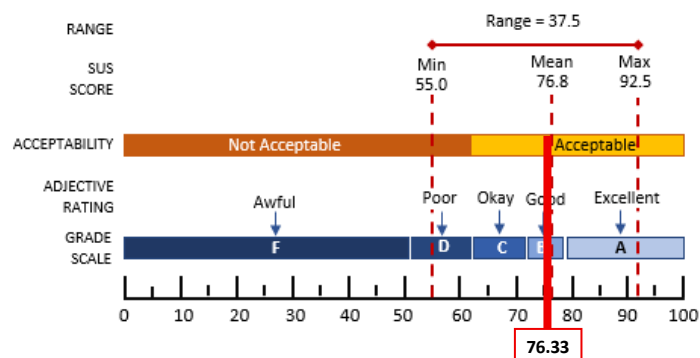
Setelah didapatkan total nilai konversi untuk setiap responden, langkah selanjutnya adalah mengalikannya dengan 2.5 untuk mendapatkan skor akhir SUS.

Tabel 4. 31 Skor Akhir *System Usability Scale*

Responden	Skor SUS (Total x 2.5)
R1	87.5
R2	75.0
R3	77.5
R4	90.0
R5	80.0
R6	57.5
R7	67.5
R8	82.5
R9	67.5

Responden	Skor SUS (Total x 2.5)
R10	77.5
R11	72.5
R12	70.0
R13	67.5
R14	90.0
R15	82.5
Total Skor SUS	1145.0
Skor-Rata-Rata SUS	76.33

Berdasarkan perhitungan dari 15 responden, skor rata-rata SUS yang didapatkan pada pengujian usability game *The Night Dreamers* adalah 76.33. Selanjutnya, skor rata-rata SUS dibandingkan dengan matriks penilaian SUS standar (*SUS Adjective Rating dan Acceptability Range*).



Gambar 4. 7 Penilaian Skor SUS

Merujuk pada skala penilaian SUS, skor 76.33 termasuk ke dalam kategori kelayakan *Acceptable* (Dapat Diterima) dengan predikat *Good* dan *Grade Scale B*.

Klasifikasi tersebut menunjukkan bahwa secara fungsional dan desain antarmuka, responden menilai fitur sistem rekomendasi sepatu adaptif, kemudahan navigasi peta, visualisasi antarmuka, serta kontrol mekanika game berjalan dengan sangat baik, mudah dipelajari (*learnability*), tinggi tingkat ketergunaannya, serta

layak digunakan oleh pemain tanpa menimbulkan kendala kognitif atau kebingungan yang berarti.

4.4 Analisis dan Pembahasan Hasil Penelitian

Sub-bab ini menyajikan analisis serta pembahasan dengan mengaitkan temuan teknis sistem adaptif *Firefly*-TOPSIS (Sub-bab 4.2) dengan hasil evaluasi ketergunaan oleh pengguna melalui metode *System Usability Scale* (Sub-bab 4.3). Pembahasan ini diarahkan untuk menjawab Rumusan Masalah 1.2, yaitu sejauh mana integrasi metode *Firefly*-TOPSIS mampu menunjukkan tingkat adaptabilitas rekomendasi pemilihan sepatu terhadap fluktuasi data capaian pemain secara real-time pada game *The Night Dreamers*.

4.4.1 Analisis Sintesis Temuan Sistem dan Ketergunaan (*Usability*)

Berdasarkan keseluruhan hasil eksperimen, terdapat keterkaitan yang cukup jelas antara perilaku mekanis algoritma di dalam *adaptation engine* dengan kenyamanan yang dirasakan oleh pemain. Keterkaitan ini dapat ditinjau dari dua sudut pandang berikut.

1. Responsivitas vektor bobot terhadap gaya bermain

Hasil pengujian pada tiga skenario (Tabel 4.5, Tabel 4.14, dan Tabel 4.23) menunjukkan bahwa sistem tidak menghasilkan keputusan pembobotan yang tetap pada setiap kondisi. Ketika pemain bermain secara ceroboh dan menerima *damage* tinggi (Percobaan Kedua), algoritma *Firefly* menggeser bobot utama ke arah kriteria proteksi fisik, yaitu *Traction* (0,557) dan *Durability* (0,405). Sebaliknya, ketika pemain menunjukkan penguasaan

mekanika permainan yang baik (Percobaan Ketiga), sistem mengurangi penekanan pada fungsi proteksi dan mengalihkan sebagian besar bobot (0,566) ke kriteria Berat Sepatu (C_1) untuk mendukung efisiensi durasi eksplorasi (*speedrunning*). Pola ini mengindikasikan bahwa pembobotan kriteria pada sistem bersifat kontekstual, bukan tetap di seluruh kondisi permainan.

2. Dampak terhadap persepsi pengguna (*User Experience*)

Fleksibilitas mekanis algoritma ini tampaknya turut berkontribusi pada skor kepuasan pengguna yang diperoleh dalam pengujian *usability*. Hasil evaluasi SUS menunjukkan skor rata-rata 76,33, yang menempatkan game ini pada kategori *Acceptable* dengan predikat *Good* (Skala Klasifikasi B). Skor yang relatif tinggi pada pertanyaan Q1 (daya tarik game) dan Q5 (fungsi fitur berjalan logis) memberikan indikasi bahwa rekomendasi sepatu yang dihasilkan oleh metode TOPSIS dipersepsikan cukup adil, fungsional, dan kontekstual oleh ke-15 partisipan saat mereka menghadapi transisi tantangan antar level.

4.4.2 Justifikasi Keunggulan Integrasi *Firefly*-TOPSIS

Rumusan masalah dalam penelitian ini mempertanyakan bukti adaptabilitas rekomendasi pemilihan sepatu menggunakan integrasi *Firefly*-TOPSIS secara real-time. Berdasarkan pembahasan hasil pada Bab IV, terdapat beberapa argumen yang mendukung keunggulan integrasi kedua metode tersebut.

1. Mengatasi keterbatasan bobot statis pada TOPSIS konvensional

Pada metode TOPSIS standar, nilai bobot kriteria (w_j) umumnya ditentukan di awal oleh pengembang game atau berdasarkan pertimbangan pakar (*expert judgment*), dan bersifat konstan sepanjang permainan. Dengan pendekatan tersebut, urutan rekomendasi sepatu cenderung tetap sama tanpa mempertimbangkan kondisi performa pemain yang berubah-ubah. Dengan mengintegrasikan *Firefly Algorithm*, vektor bobot dihitung ulang secara dinamis pada setiap akhir level berdasarkan fluktuasi metrik *Score*, *Damage Taken*, *Play Time*, dan *Lives Left*. Mekanisme ini memungkinkan *adaptation engine* untuk merespons perubahan performa pengguna, alih-alih bekerja dengan bobot yang ditetapkan secara tetap sejak awal.

2. Diferensiasi konteks pada alternatif sepatu

Meskipun sepatu A_2 (*Adidas Ultraboost 22*) memiliki nilai spesifikasi teknis dasar yang relatif dominan di dalam database, di mana cenderung akan menduduki peringkat atas pada kondisi tertentu, hasil pengujian menunjukkan bahwa integrasi *Firefly*-TOPSIS tetap mampu menghasilkan diferensiasi peringkat alternatif lain sesuai konteks. Pada Percobaan Pertama (Eksplorasi Aman), misalnya, peringkat pertama justru ditempati oleh A_5 (*Nike Structure 25*) dengan nilai V_i sebesar 0,905, sejalan dengan keunggulan fitur *stability system*-nya yang selaras dengan bobot kriteria C_5 (0,456) hasil optimasi *Firefly*. Pergeseran nilai preferensi (V_i) antar percobaan ini menjadi indikasi bahwa integrasi *Firefly*-TOPSIS mampu menghasilkan sistem pengambilan keputusan multikriteria yang responsif terhadap kondisi performa pemain yang berbeda-beda.

Melalui hasil analisis ini, implementasi metode integrasi *Firefly-TOPSIS* dinyatakan berhasil menjawab rumusan masalah penelitian, menjaga keseimbangan permainan (*gameplay balance*), serta memberikan kontribusi praktis dalam ranah pengembangan game adaptif berbasis keputusan cerdas (*intelligent decision support systems*).

4.5 Integrasi Penelitian dengan Nilai-Nilai Islam

Kajian multimedia dan sistem adaptif dalam penelitian ini diselaraskan dengan tradisi keilmuan Islam melalui pendekatan *Maqashid Syariah*. Tujuan utama pengembangan game prototipe *The Night Dreamers* diarahkan pada pemeliharaan tiga unsur pokok kehidupan (*al-kulliyat al-khams*).

4.5.1 Pemeliharaan Agama (Hifzh al-Din)

Pemeliharaan aspek agama diwujudkan melalui muatan edukasi spiritual substantif, yakni pengenalan keutamaan *Shalat Tahajud*. Internalisasi nilai ini diintegrasikan secara organik ke dalam narasi game. Saat pemain berhasil mengumpulkan kunci, ayat-ayat Al-Qur'an (seperti QS. Al-Baqarah: 286 dan QS. Al-Mulk: 3 – 4) disajikan melalui *Objek Al-Qur'an Text UI* sebagai refleksi keimanan. Fase akhir permainan (*ending scene*) yang menampilkan karakter utama melaksanakan ibadah dirancang untuk merepresentasikan luaran pembelajaran (*learning outcomes*) dari aspek psikomotorik, sehingga menegaskan posisi teknologi sebagai media dakwah kontemporer yang relevan bagi generasi digital.

4.5.2 Pemeliharaan Jiwa (Hifzh al-Nafs)

Dalam dimensi psikologis, game yang tidak seimbang dapat menimbulkan frustrasi atau kebosanan, yang berdampak negatif pada kondisi mental pemain. Sistem adaptif *Firefly*-TOPSIS berkontribusi langsung dalam menjaga kondisi *flow state*, yaitu keseimbangan antara tingkat tantangan game dan kemampuan aktual pemain. Pendekatan ini merefleksikan prinsip moderasi (*wasatiyyah*) dalam Islam, di mana sistem memperlakukan pemain secara adil dan proporsional demi menjaga kenyamanan psikologis pengguna.

4.5.3 Pemeliharaan Akal (Hifzh al-‘Aql)

Penerapan *Firefly Algorithm* yang mengadopsi perilaku kunang-kunang di alam semesta merupakan bentuk optimalisasi akal dalam meniru keteraturan ciptaan Allah SWT. Dari sisi pengguna, game ini melatih kemampuan kognitif, analisis spasial, dan pengambilan keputusan multikriteria saat menentukan strategi pemilihan sepatu. Proses riset komputasi ini merupakan wujud nyata perintah menuntut ilmu untuk melakukan perubahan ke arah yang lebih baik, sejalan dengan QS. Ar-Ra’d: 11.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan eksperimen yang telah dilakukan pada game *The Night Dreamers*, dapat disimpulkan bahwa metode integrasi *Firefly-TOPSIS* berhasil membuktikan tingkat adaptabilitas rekomendasi pemilihan sepatu secara *real-time* berdasarkan fluktuasi data capaian pemain. Sistem secara dinamis mampu melakukan penyesuaian (*gameplay balancing*) melalui dua tahapan komputasi yang responsif terhadap karakteristik gaya bermain.

Pada tahap pertama, modul *Firefly Algorithm* berhasil mengoptimasi dan menggeser vektor bobot kriteria secara kontekstual mengikuti Faktor Capaian Pemain (*Fcp*). Pada kondisi pemain ceroboh yang menerima kerusakan besar (*Eksplorasi Agresif*), prioritas dialihkan secara otomatis ke kriteria proteksi fisik, yakni *Traction* sebesar 0,557 dan *Durability* sebesar 0,405. Sebaliknya, pada kondisi pemain mahir (*Penguasaan Gameplay*), fokus dialihkan menuju kriteria berat sepatu (C_1 sebesar 0,566) untuk mendukung efisiensi waktu eksplorasi.

Pada tahap kedua, modul *TOPSIS* terbukti sensitif dalam mengevaluasi dan menetapkan peringkat alternatif sepatu berdasarkan kedekatan relatif terhadap solusi ideal dari bobot dinamis yang dihasilkan. Hal ini dibuktikan dengan bergesernya rekomendasi utama secara logis sesuai skenario gaya bermain, di mana *Adidas Ultraboost 22* ($V_i = 0,988$) menjadi rekomendasi teratas untuk gaya bermain agresif demi stabilitas fisik, sementara *Nike Structure 25* ($V_i = 0,982$)

menjadi rekomendasi utama bagi pemain *master* untuk mendukung kelincahan pergerakan. Dengan demikian, integrasi kedua metode ini secara komprehensif terbukti mampu menghasilkan rekomendasi yang adaptif dan kontekstual terhadap karakteristik unik setiap pemain.

5.2 Saran

Berdasarkan hasil evaluasi serta batasan masalah yang ada, beberapa saran yang dapat diajukan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Mengimplementasikan varian *Hybrid Firefly Algorithm* dengan strategi pembelajaran dua tingkat untuk meningkatkan stabilitas eksplorasi solusi dan mengurangi risiko konvergensi prematur pada data performa pemain yang fluktuatif.
2. Memperluas cakupan sistem adaptif agar tidak hanya mengatur pemilihan item karakter, tetapi juga elemen permainan lain seperti tingkat kesulitan AI musuh, frekuensi rintangan, atau tata letak level secara prosedural.
3. Menambahkan kriteria teknis sepatu yang lebih beragam serta memperluas jumlah alternatif produk dalam database untuk mengakomodasi variasi tantangan taktis yang lebih luas.
4. Melakukan pengujian *usability* dengan jumlah responden yang lebih besar dan durasi bermain yang lebih panjang guna mengukur dampak jangka panjang sistem terhadap motivasi, *flow state*, dan *player engagement*.
5. Perlu dilakukan evaluasi dan kalibrasi ulang terhadap nilai baseline pada matriks keputusan awal (*Shoe Database*) atau menerapkan metode

pembobotan hibrida yang menyertakan threshold batas bawah. Hal ini penting untuk mengatasi fenomena *dominance effect* di mana alternatif tertentu (seperti A_6) memperoleh nilai preferensi $V_i = 0,000$ secara konsisten akibat kalah dominan pada seluruh kriteria benefit dibandingkan dengan produk premium lainnya.

6. Mengingat adanya kendala penurunan performa kecepatan bingkai (*frame rate drop*) pada beberapa perangkat responden saat mengeksplorasi environment game, penelitian selanjutnya disarankan untuk menerapkan standarisasi spesifikasi sistem minimum (minimal prosesor Quad-Core 2,5 GHz, RAM 8 GB, dan Dedicated GPU dengan VRAM 2 GB).

DAFTAR PUSTAKA

- Andrade, G., Ramalho, G., Santana, H., & Corruble, V. (2006). Challenge-sensitive action selection: an application to game balancing. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*.
- C.T., & A.K. (2025). Personalized instructional strategy adaptation using TOPSIS in intelligent learning systems. *Computers & Education: Artificial Intelligence*, 8. <https://doi.org/10.1016/j.caeai.2025.100200>
- Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper & Row.
- Drachen, A., Seif El-Nasr, M., & Canossa, A. (2013). *Game Analytics: Maximizing the Value of Player Data*. Springer.
- Hassan, M., Malik, A., & Rahman, N. (2021). Adaptive gameplay systems for dynamic learning environments: A survey and framework proposal. *Entertainment Computing*, 39. <https://doi.org/10.1016/j.entcom.2021.100438>
- Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*.
- Hwang, C. L., & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag.
- Katsir, I. (2004). *Tafsir Ibnu Katsir jilid 1* (M. A. Ghoffar, A. Mu'thi, & A. I. Al-Atsari, Terj.; A. M. Al-Sheikh). Pustaka Imam Asy-Syafi'i.
- Katsir, I. (2004). *Tafsir Ibnu Katsir jilid 4* (M. A. Ghoffar, A. Mu'thi, & A. I. Al-Atsari, Terj.; A. M. Al-Sheikh). Pustaka Imam Asy-Syafi'i.
- Katsir, I. (2004). *Tafsir Ibnu Katsir jilid 8* (M. A. Ghoffar, A. Mu'thi, & A. I. Al-Atsari, Terj.; A. M. Al-Sheikh). Pustaka Imam Asy-Syafi'i.
- Koto, & Dafitri. (2024). Optimalisasi Pemilihan Title untuk Senjata pada Game Point Blank Menggunakan Metode TOPSIS. *JITEKH*, 12(2), 85–93.
- Kumar, R. (2024). A Comprehensive Review of MCDM Methods, Applications, and Emerging Trends. *Decision Making Advances*, 3, 185–199. <https://doi.org/10.31181/dma31202569>
- Liu, Y., Zhang, Z., & Li, J. (2020). Dynamic adaptive Firefly Algorithm with global orientation for complex optimization problems. *Applied Soft Computing*, 94. <https://doi.org/10.1016/j.asoc.2020.106451>
- Mahlmann, T., et al. (2010). Predicting player behavior in Tomb Raider: Underworld. *Proceedings of the 2010 IEEE Symposium on Computational*

Intelligence and Games.

- Meshrif, M., Khan, S., & Kumar, P. (2023). Adaptive game design using machine learning techniques: A comprehensive survey. *Entertainment Computing*, 45. <https://doi.org/10.1016/j.entcom.2023.100532>
- Micale, R., Marannano, G., Giallanza, A., Miglietta, P. P., Agnusdei, G. P., & La Scalia, G. (2019). Sustainable vehicle routing based on Firefly Algorithm and TOPSIS methodology. *Sustainable Futures*, 1, 100001. <https://doi.org/10.1016/j.sftr.2019.100001>
- Power, D. J. (2002). *Decision Support Systems: Concepts and Resources for Managers*. Greenwood Publishing Group.
- Rahman, A., Yuliani, S., & Prasetyo, R. (2023). Optimization of multi-criteria decision systems using hybrid Firefly–TOPSIS approach. *Procedia Computer Science*, 225. <https://doi.org/10.1016/j.procs.2023.09.097>
- Rahman, T., Yusuf, A., & Nugraha, D. (2022). Real-time adaptive mechanisms in game development using player performance analytics. *Journal of Game Design and Development Education*, 12(3). <https://doi.org/10.1016/j.jgdd.2022.102>
- Susila, & Arsa. (2023). Analisis Pengujian Sistem Menggunakan System Usability Scale (SUS).
- Turban, E., Sharda, R., & Delen, D. (2011). *Decision Support and Business Intelligence Systems*. Pearson.
- Wang, J., He, X., & Liu, R. (2023). Hybrid Firefly Algorithm with double-level learning strategy for multi-objective optimization problems. *Expert Systems with Applications*, 227. <https://doi.org/10.1016/j.eswa.2023.120315>
- Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78-84.
- Yang, X.-S., et al. (2022). *Firefly Algorithm and Its Applications*. Frontiers.
- Yannakakis, G. N., & Togelius, J. (2018). *Artificial Intelligence and Games*. Springer.
- Zhou, J., Li, X., & Chen, W. (2021). TOPSIS-based evaluation model integrated with bio-inspired optimization for adaptive control systems. *Knowledge-Based Systems*, 228. <https://doi.org/10.1016/j.knosys.2021.107244>

LAMPIRAN

Lampiran 1. Record Data Responden

No	Stage	Score	Damage Taken	Play Time (s)	Lives Left	Fcp	Rekomendasi Sepatu	Nilai Vi
1	1	30	45	36.761	3	0.630	A5 – Asics Gel-Nimbus 25	0.982
	2	40	65	38.244	3	0.600	A2 – Nike Pegasus 40	0.986
	3	40	120	89.686	2	0.394	A2 – Nike Pegasus 40	0.988
2	1	30	65	108.635	3	0.523	A4 – Adidas Solarboost 5	0.987
	2	40	65	53.589	3	0.589	A2 – Nike Pegasus 40	0.883
	3	60	110	200.424	2	0.380	A5 – Asics Gel-Nimbus 25	0.905
3	1	30	45	40.003	3	0.628	A2 – Nike Pegasus 40	0.916
	2	40	75	37.950	3	0.570	A2 – Nike Pegasus 40	0.953
	3	60	95	86.040	3	0.538	A5 – Asics Gel-Nimbus 25	0.905
4	1	30	40	57.759	3	0.631	A5 – Asics Gel-Nimbus 25	0.928
	2	40	75	44.829	3	0.565	A4 – Adidas Solarboost 5	0.951
	3	60	110	97.677	2	0.448	A4 – Adidas Solarboost 5	0.937
5	1	30	75	75.460	3	0.515	A8 – Saucony Endorphin Speed 3	0.926
	2	20	30	28.649	3	0.651	A4 – Adidas Solarboost 5	0.971
	3	50	85	71.127	3	0.548	A2 – Nike Pegasus 40	0.982
6	1	30	70	47.602	3	0.548	A4 – Adidas Solarboost 5	0.917
	2	40	75	45.933	3	0.564	A5 – Asics Gel-Nimbus 25	0.911
	3	60	150	131.070	2	0.426	A2 – Nike Pegasus 40	0.965
7	1	30	50	193.023	3	0.511	A2 – Nike Pegasus 40	0.985
	2	40	70	88.355	3	0.551	A4 – Adidas Solarboost 5	0.942
	3	60	105	121.049	2	0.433	A2 – Nike Pegasus 40	0.987
8	1	30	60	35.438	3	0.586	A2 – Nike Pegasus 40	0.940
	2	40	75	41.474	3	0.567	A4 – Adidas Solarboost 5	0.858
	3	60	110	102.786	2	0.445	A8 – Saucony Endorphin Speed 3	0.918
9	1	30	40	49.981	3	0.637	A2 – Nike Pegasus 40	0.961

	2	40	60	38.108	3	0.615	A4 – Adidas Solarboost 5	0.985
	3	60	90	94.598	3	0.547	A4 – Adidas Solarboost 5	0.985
10	1	30	50	75.205	3	0.590	A5 – Asics Gel-Nimbus 25	0.987
	2	40	75	89.586	3	0.535	A5 – Asics Gel-Nimbus 25	0.874
	3	60	120	137.690	2	0.422	A5 – Asics Gel-Nimbus 25	0.987
11	1	30	45	52.486	3	0.620	A4 – Adidas Solarboost 5	0.970
	2	40	85	50.347	3	0.531	A5 – Asics Gel-Nimbus 25	0.986
	3	60	95	68.854	3	0.549	A2 – Nike Pegasus 40	0.963
12	1	30	60	40.015	3	0.583	A5 – Asics Gel-Nimbus 25	0.948
	2	40	90	45.892	3	0.519	A5 – Asics Gel-Nimbus 25	0.984
	3	40	65	42.465	3	0.597	A8 – Saucony Endorphin Speed 3	0.979
13	1	30	50	1.036.398	3	0.440	A5 – Asics Gel-Nimbus 25	0.951
	2	40	60	95.093	3	0.577	A4 – Adidas Solarboost 5	0.903
	3	60	100	188.713	2	0.388	A4 – Adidas Solarboost 5	0.985
14	1	30	50	55.034	3	0.603	A5 – Asics Gel-Nimbus 25	0.846
	2	40	70	48.626	3	0.578	A8 – Saucony Endorphin Speed 3	0.849
	3	60	90	50.211	3	0.577	A4 – Adidas Solarboost 5	0.938
15	1	30	75	155.914	3	0.461	A5 – Asics Gel-Nimbus 25	0.919
	2	40	85	59.459	3	0.525	A8 – Saucony Endorphin Speed 3	0.958
	3	60	90	83.590	3	0.554	A4 – Adidas Solarboost 5	0.868

Lampiran 2. Website Spesifikasi Teknis Sepatu

No	Alternatif	Link
1	Nike Air Zoom Pegasus 40	https://www.nike.com/id/t/pegasus-40-road-running-shoes-3JpHzl
2	Adidas Ultraboost 22	https://jdsports.id/product/adidas-ultraboost-22-gz0127?srsId=AfmBOoqMtqJJPcK-YRMM0-LL2qTvrGRB_nxwMVQU4WLOXi1l6lh23Cli
3	New Balance Fresh Foam 1080v13	https://runrepeat.com/new-balance-fresh-foam-x-1080-v-13
4	ASICS Gel-Nimbus 25	https://www.asics.com/gb/en-gb/mk/gel-nimbus-25
5	Nike Structure 25	https://runrepeat.com/nike-structure-25
6	Adidas Adizero SL	https://www.adidas.co.id/id/sepatu-adizero-sl/IG3334.html
7	ASIC Gel-Cumulus 25	https://runrepeat.com/asics-gel-cumulus-25
8	Puma Velocity Nitro 2	https://runrepeat.com/puma-velocity-nitro-2
9	Brooks Ghost 15	https://runrepeat.com/brooks-ghost-15
10	Saucony Ride 16	https://geeksonfeet.com/shoogeeks/saucony-ride16-006009016