

**PENERAPAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW) DALAM  
SIMULASI OPERASIONAL SUPERMARKET**

**SKRIPSI**

Oleh :  
**M.TAZKIA IKHSANUL MA'ÁRIF**  
**NIM. 210605110028**



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

**PENERAPAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)  
DALAM SIMULASI OPERASIONAL SUPERMARKET**

**SKRIPSI**

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

Oleh :  
**M.TAZKIA IKHSANUL MA'ÁRIF**  
**NIM. 210605110028**

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

**HALAMAN PERSETUJUAN**

**PENERAPAN METODE SIMPLE ADDITIVE WEIGHTING (SAW)  
DALAM SIMULASI OPERASIONAL SUPERMARKET**

**SKRIPSI**

Oleh :

**M.TAZKIA IKHSANUL MA'ARIF**

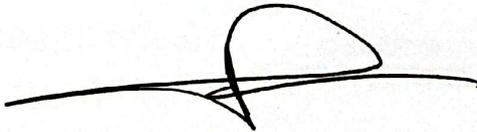
**NIM. 210605110028**

Telah Diperiksa dan Disetujui untuk Diuji:

Tanggal: 19 Juni 2025

Pembimbing I,

Pembimbing II,



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



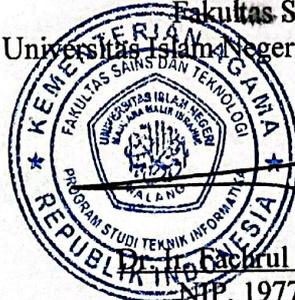
Roro Inda Melani, M.T., M.Sc  
NIP. 19780925 200501 2 008

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



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

**HALAMAN PENGESAHAN**

**PENERAPAN METODE SIMPLE ADDITIVE WEIGHTING (SAW)  
DALAM SIMLUASI OPERASIONAL SUPERMARKET**

**SKRIPSI**

Oleh :  
**M.TAZKIA IKHSANUL MA'ARIF**  
**NIM. 210605110028**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 11 Juni 2025

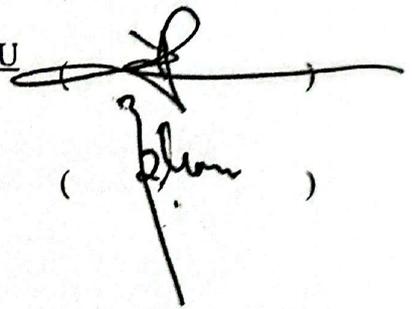
**Susunan Dewan Penguji**

Ketua Penguji : Supriyono, M.Kom  
NIP. 19841010 201903 1 012

Anggota Penguji I : Ashri Shabrina Afrah, M.T  
NIP. 19900430 202012 2 003

Anggota Penguji II : Dr. Ir. Fachrul Kurniawan, M.MT., IPU  
NIP. 19771020 200912 1 001

Anggota Penguji III : Roro Inda Melani, M.T., M.Sc  
NIP. 19780925 200501 2 008



Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang



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

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : M.Tazkia Ikhsanul Ma'arif  
NIM : 210605110028  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Penerapan Metode Simple Additive Weighting (SAW) dalam Simulasi Operasional Supermarket

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, 01 Juni 2025  
Yang membuat pernyataan,



M.Tazkia Ikhsanul Ma'arif  
NIM.210605110028

## **MOTTO**

*“Setiap perubahan besar dimulai dari Langkah kecil yang konsisten” ...*

## **HALAMAN PERSEMBAHAN**

Dengan penuh rasa syukur kepada Allah SWT atas segala karunia, rahmat, dan kasih sayang-Nya yang selalu menyertai setiap langkah penulis hingga skripsi ini dapat diselesaikan. Karya ini penulis dedikasikan kepada:

1. Ibu tercinta, yang senantiasa menjadi sumber cinta yang tiada batas, memberikan dukungan, doa, dan semangat yang tak pernah putus, serta menjadi cahaya penerang dalam setiap langkah penulis.
2. Ayah tercinta, yang selalu mengajarkan arti keteguhan, kerja keras, serta memberikan kasih sayang yang tulus tanpa syarat, dan menjadi teladan yang senantiasa menginspirasi penulis.
3. Adik perempuan tercinta, yang selalu memberikan inspirasi dengan kebijaksanaan dan perhatian, serta selalu menyemangati dengan cinta dan kebaikan yang tiada henti.
4. Diri sendiri, terima kasih telah berjuang tanpa mengenal lelah, tetap teguh menghadapi segala tantangan, dan menyelesaikan setiap langkah ini dengan penuh kesabaran. Semoga karya ini menjadi awal dari perjalanan yang lebih berarti.

## KATA PENGANTAR

Segala puji syukur senantiasa penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode *Simple Additive Weighting* (SAW) dalam simulasi operasional supermarket” ini dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, suri teladan umat manusia, beserta keluarga, sahabat, dan pengikutnya hingga akhir zaman. Ucapan terima kasih sebanyak-banyaknya saya ucapkan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku Rektor UIN Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT., IPU., selaku Kepala Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
4. Dr. Ir. Fachrul Kurniawan, M.MT., IPU., selaku dosen pembimbing pertama, yang telah memberikan bimbingan, saran, dan masukan yang konstruktif dalam penyusunan skripsi ini.
5. Roro Inda Melani, M.T., M.Sc., selaku dosen pembimbing kedua, atas kesabaran dan arahan yang sangat membantu dalam menyempurnakan karya ini.

6. Supriyono, M. Kom., selaku Ketua Dewan Penguji, atas koreksi dan saran yang membangun selama ujian skripsi berlangsung.
7. Ashri Shabrina Afrah, M.T., selaku penguji kedua, atas evaluasi dan masukan yang telah memperbaiki kualitas skripsi ini.
8. Kedua orang tua tercinta, Ibu dan Ayah, serta adik perempuan saya, yang telah menjadi sumber inspirasi, kekuatan, serta doa yang tidak pernah diputus. Terima kasih atas cinta dan pengorbanan yang telah menjadi dasar dari setiap langkah perjalanan penulis.
9. Segenap dosen di Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang yang telah Ikhlas untuk mendidik dan menyalurkan ilmu pengetahuan selama ini.
10. Rekan-rekan dan sahabat Jurusan Teknik Informatika 2021 dan seperjuangan bimbingan skripsi yang memberikan semangat dan dukungan untuk menyelesaikan skripsi.
11. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah banyak membantu dalam menyelesaikan skripsi ini.

Segala upaya telah saya lakukan untuk menyusun laporan ini, akan tetapi tidak menutup kemungkinan apabila dalam laporan ini masih terdapat kekurangan dan kesalahan. Maka sebab itu, saya harapkan terdapat kritik dan saran yang dapat dijadikan masukan dalam menyempurnakan laporan penelitian skripsi saya ini.

Malang, 28 Februari 2025

Penulis

## DAFTAR ISI

<b>HALAMAN PENGAJUAN .....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR .....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR .....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>ABSTRAK .....</b>	<b>xvi</b>
<b>ABSTRACT .....</b>	<b>xvii</b>
<b>مستخلص البحث .....</b>	<b>xviii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	6
<b>BAB II STUDI PUSTAKA .....</b>	<b>7</b>
2.1 Penelitian Terdahulu .....	7
2.2 Manajemen Barang .....	13
2.3 Game Simulasi .....	15
2.4 <i>Simple Additive Weighting (SAW)</i> .....	16
2.4.1 Identifikasi Kriteria dan Alternatif .....	17
2.4.2 Alur Matrix Keputusan .....	18
2.5 Unity .....	20
2.6 <i>Finite State Machine (FSM)</i> .....	22
<b>BAB III DESAIN DAN IMPLEMENTASI .....</b>	<b>25</b>
3.1 Desain Alur Penelitian .....	25
3.2 Skenario Simulasi .....	27
3.2.1 FSM Simulasi .....	27
3.2.2 Perancangan Simulasi .....	40
3.3 Perancangan Metode <i>Simple Additive Weighting</i> .....	42
3.3.1 Pengumpulan Data Alternatif dan Kriteria .....	43
3.3.2 Perhitungan Normalisasi Data .....	46
3.3.3 Perhitungan Skor Alternatif .....	48
3.3.4 Prediksi Rekomendasi menggunakan SAW .....	51
3.4 Rencana Pengujian .....	52
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>55</b>
4.1 Implementasi Simulasi .....	55
4.2 Implementasi Metode SAW .....	80
4.3 Hasil dan Analisis Implementasi SAW .....	86

4.4 Pengujian Usability .....	103
4.5 Integrasi Sains dan Islam.....	109
4.5.1 Muamalah Mu'Allah.....	109
4.5.2 Muamalah Mu'annas dan Mu'alam .....	111
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>113</b>
5.1 Kesimpulan .....	113
5.2 Saran .....	114
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 3.1 Alur Penelitian.....	25
Gambar 3.2 FSM Skenario Simulasi .....	28
Gambar 3.3 FSM Perhitungan SAW .....	30
Gambar 3.4 FSM NPC Customer .....	32
Gambar 3.5 FSM Interact Computer .....	34
Gambar 3.6 FSM Interact Book .....	35
Gambar 3.7 FSM Interact Shelf.....	37
Gambar 3.8 FSM Interact Sign.....	38
Gambar 3.9 FSM Interact Cashier .....	39
Gambar 3.10 Ilustrasi Simulasi .....	41
Gambar 3.11 Skenario perhitungan SAW .....	42
Gambar 4.1 Tampilan Menu .....	56
Gambar 4.2 Tampilan Tutorial .....	57
Gambar 4.3 Tampilan Area Gudang.....	58
Gambar 4.4 Tampilan Interaksi Buku.....	59
Gambar 4.5 Tampilan Data Barang di gudang.....	60
Gambar 4.6 Tampilan Input quantity barang untuk ditransfer .....	61
Gambar 4.7 Tampilan Box yang dipilih.....	62
Gambar 4.8 Tampilan restock barang di display .....	63
Gambar 4.9 Tampilan Input harga jual barang.....	64
Gambar 4.10 Tampilan barang yang sudah direstock.....	65
Gambar 4.11 Tampilan OpenStore .....	66
Gambar 4.12 Tampilan pelanggan membeli barang .....	67
Gambar 4.13 Tampilan antrian customer.....	68
Gambar 4.14 Tampilan pelanggan checkout.....	69
Gambar 4.15 Tampilan Button EndDay .....	70
Gambar 4.16 Tampilan Card List .....	71
Gambar 4.17 Tampilan Result SAW .....	72
Gambar 4.18 Tampilan Notifikasi Rekomendasi .....	73
Gambar 4.19 Tampilan kandidat rekomendasi.....	74
Gambar 4.20 Tampilan Box gudang.....	75
Gambar 4.21 Tampilan interact computer .....	76
Gambar 4.22 Tampilan menu komputer .....	77
Gambar 4.23 Tampilan Shop komputer.....	78
Gambar 4.24 Tampilan Box Spawn.....	79
Gambar 4.25 Tampilan barang yang terjual percobaan pertama.....	87
Gambar 4.26 Tampilan rincian data yang diperoleh percobaan pertama .....	88
Gambar 4.27 Tampilan notifikasi kandidat percobaan pertama.....	91
Gambar 4.28 Tampilan kandidat barang di display percobaan pertama.....	92
Gambar 4.29 Tampilan barang yang terjual percobaan kedua .....	93

Gambar 4.30 Tampilan rincian data yang diperoleh percobaan kedua.....	94
Gambar 4.31 Tampilan notifikasi kandidat percobaan kedua .....	96
Gambar 4.32 Tampilan kandidat barang di display percobaan kedua .....	97
Gambar 4.33 Tampilan barang yang terjual percobaan ketiga.....	98
Gambar 4.34 Tampilan rincian data yang diperoleh percobaan ketiga .....	100
Gambar 4.35 Tampilan notifikasi kandidat percobaan ketiga.....	102
Gambar 4.36 Tampilan kandidat barang didisplay percobaan ketiga.....	103

## DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian ini .....	11
Tabel 3.1 Tabel Kriteria .....	44
Tabel 3.2 Data Bobot .....	44
Tabel 3.3 Data Barang .....	45
Tabel 3.4 Data Normalisasi .....	47
Tabel 3.5 Rating Alternatif.....	50
Tabel 3.6 Daftar Pertanyaan Kuesioner SUS .....	52
Tabel 4.1 Pseudocode Matriks Data Barang .....	81
Tabel 4.2 Pseudocode Normalisasi Data Barang.....	82
Tabel 4.3 Pseudocode Skor Alternatif .....	83
Tabel 4.4 Pseudocode Descending Alternatif .....	84
Tabel 4.5 Data Barang percobaan kedua.....	87
Tabel 4.6 Rating Alternatif percobaan pertama .....	89
Tabel 4.7 Data Barang percobaan kedua.....	93
Tabel 4.8 Rating alternatif percobaan kedua.....	95
Tabel 4.9 Data Barang pada percobaan ketiga .....	99
Tabel 4.10 Rating alternatif percobaan ketiga.....	101
Tabel 4.11 Hasil Kuesioner SUS .....	104
Tabel 4.12 Skor total SUS .....	106
Tabel 4.13 Hasil skor SUS .....	107
Tabel 4.14 Pedoman SUS .....	108

## ABSTRAK

Ma'arif, M. Tazkia Ikhsanul. 2025. **Penerapan Metode *Simple Additive Weighting* (SAW) dalam Simulasi Operasional Supermarket**. Skripsi, Program Studi Teknik Informatika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Fachrul Kurniawan, M.MT, IPU (II) Roro Inda Melani, M.T, M.Sc.

**Kata Kunci:** Simulasi Supermarket, *Simple Additive Weighting* (SAW), Prioritas Restok Barang, Game Edukasi, Manajemen Stok, *System Usability Scale* (SUS).

Proses pengambilan keputusan dalam manajemen stok menjadi aspek penting dalam operasional sebuah supermarket. Permasalahan yang sering terjadi adalah ketidaktepatan dalam menentukan prioritas penyimpanan dan pengadaan barang, akibat keputusan yang masih bersifat manual atau tanpa perhitungan berbasis data yang jelas. Hal ini berpotensi menimbulkan ketidakseimbangan stok, seperti kekurangan barang yang diminati atau penumpukan barang yang kurang laku. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode *Simple Additive Weighting* (SAW) dalam simulasi operasional supermarket guna membantu menentukan prioritas restok barang secara objektif. Metode SAW dipilih karena mampu mengolah beberapa kriteria penting seperti harga, jumlah stok, masa kadaluarsa, dan data permintaan untuk menghasilkan skor prioritas setiap barang. Nilai tersebut ditampilkan dalam sistem ranking agar pengambilan keputusan restok menjadi lebih efektif dan terukur. Hasil pengujian kelayakan menggunakan *System Usability Scale* (SUS) memperoleh skor rata-rata 80,28 yang termasuk kategori baik. Temuan ini menunjukkan bahwa sistem yang dikembangkan mudah digunakan, dapat menyajikan informasi prioritas restok dengan akurat, dan berpotensi menjadi media edukatif bagi pemain dalam memahami konsep pengambilan keputusan berbasis data di lingkungan supermarket.

## ABSTRACT

Ma'arif, M.Tazkia Ikhsanul. 2025. **The Implementation of the Simple Additive Weighting (SAW) Method in Supermarket Operational Simulation.** Thesis, Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Supervisors: (I) Dr. Fachrul Kurniawan, M.MT, IPU (II) Roro Inda Melani, M.T, M.Sc.

**Keywords:** Supermarket Simulation, Simple Additive Weighting (SAW), Goods Restocking Prioritization, Educational Game, Stock Management, System Usability Scale.

The decision-making process in stock management is an important aspect in the operation of a supermarket. The problem that often occurs is the inaccuracy in determining the priority of storage and procurement of goods, due to decisions that are still manual or without clear data-based calculations. This has the potential to cause stock imbalances, such as a shortage of goods in demand or an accumulation of goods that are not selling well. To overcome these problems, this research applies the Simple Additive Weighting (SAW) method in simulating supermarket operations to help determine the priority of stocking goods objectively. The SAW method was chosen because it is able to process several important criteria such as price, stock quantity, expiration period, and demand data to produce a priority score for each item. The score is displayed in a ranking system to make restock decision making more effective and measurable. The results of feasibility testing using the System Usability Scale (SUS) obtained an average score of 80.28 which is in the good category. This finding shows that the developed system is easy to use, can present restock priority information accurately, and has the potential to become an educational medium for players in understanding the concept of data-based decision making in a supermarket environment. The decision-making process in stock management is an important aspect in the operation of a supermarket. Problems that often occur are.

## مستخلص البحث

لونساحا ايكزات. م، فرامع. ٢٠٢٥. ليغشت تاكاحم في (SAW) طيسبلا في فاضلا حيجرتلا تقيرط قبيطت تجماج، ايجولونكتلاو مولعا تيلك، تيتامولعما تسدنه تسارد جمانرب، تحورطا. يربكلا رجاتملا في ريتسجام، ن او اينروكل ورخف د) ١ ( : فرشملا. جنلاما تيموكلحا تيملاسلا ميهارب ا كلام اتلاوم ورور) ٢ ( ايجولونكتلاو مولعا في ريتسجام، تيتامولعما تسدنه في ريتسجام، تيتامولعما تسدنه ايجولونكتلاو مولعا في ريتسجام، في نلايم ادنا.

الكلمات المفتاحية: رارقلا ناختا، علسلا عارش، (SAW) طيسبلا في فاضلا حيجرتلا، تكرام ربولسا تاكاحم تاكاحملا تيعل.

في تلا بعناشلا لكاشملا نمو بتكرامز بوسلا ليغشت في امهم اعزج نورخملا قراد في رارقلا ناختا تيلمع دعت تمياف ريغ و ا تيديقت لظت في تلا تاز ارقلا بتسبب، معناضبلا ريفوتو نيزخت ببولوا ديخت في قديلا مدعي هر هظن و ا ببولطما معناضبلا صقت لثم، نورخملا نراوت مدعي لا كذا في دوي ن ان كميو، محضاو تينايت تاسد في اء تيجارلا ريغ معناضبلا نم ضافلا. معجلا تقيرط تقيرط قبيطت تخبلا اده ما، لكاشملا هذه تجماعلو معناضبلا ريفوت دواعي ببولوا ديخت في دواعسمل بتكرامز بوسلا ليغشت تاكاحم في (SAW) طيسبلا في حيجرتلا تيمكو، ريسلا لثم، امهم ريباعم دوع تجماعم في اء اهترنق SAW تقيرط زابتخا متو بي عوضوم لكشيد في قديلا هذه هظنو، معاضب ل كل ببولوا تجرد و ا تميقة اعطلا، باطلا تانايو، تيجالاصلا دمو، نورخملا متي ذلا ماظنلا تيجالاصد مبيقت تانند تر هظاو. اسابقو تيلاعف رنكا لكشيد رارقلا ناختا في دواعسمل بتيرتلا ماظن وهو، ٢٨، ٨٠، في ل لصو طاقت دغم في اء ل صدقنا (SUS) ماظنلا مادختسا تيلباق سايقم ل لاخ نم مربوطت ضرعي اء رفاق، لامعتسلا ل هسد مربوطت متي ذلا ماظنلا ن ا تجميئنا هذه ن يئو. "ديجلا" تيفلا ن مضع قيد مناقل رارقلا ناختا ميهافم ميفل ن بيدعلا تيميلعد تليسو نو كي ن ان كميو، تقيد ريفوتلا دواعي ببولوا تامولعم بتكرامز بوسلا تيب في تانايلا في اء

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pertumbuhan ekonomi Indonesia yang pesat telah membawa perubahan besar pada sektor ritel. *Urbanisasi*, peningkatan pendapatan individu, dan daya beli masyarakat yang terus meningkat mendorong perkembangan industri ini. Selain itu, kemajuan teknologi, khususnya dalam bidang *e-commerce* dan layanan digital, telah menyederhanakan proses bagi pelanggan untuk mengakses berbagai produk dan layanan. Menurut (Fuaddah, 2023), pola konsumsi masyarakat saat ini dipengaruhi oleh perubahan gaya hidup yang mengutamakan kenyamanan dan efisiensi. Hal ini memunculkan berbagai model bisnis baru, seperti toko kelontong, supermarket, hingga pusat perbelanjaan modern.

Perkembangan ritel modern, khususnya supermarket, telah mengubah perilaku berbelanja masyarakat. Sebelumnya, pelanggan harus mengunjungi beberapa toko untuk memenuhi kebutuhan sehari-hari. Namun, kehadiran supermarket telah membuat proses berbelanja menjadi lebih efisien dan nyaman. Supermarket menjual berbagai produk dalam satu tempat, sehingga konsumen dapat menemukan segala kebutuhannya tanpa harus berpindah dari satu lokasi ke lokasi lain. Selain itu, perkembangan bisnis ritel berbasis daring (*online*) semakin meningkatkan efisiensi dan kenyamanan berbelanja bagi konsumen (Pramudya, 2024).

Supermarket menawarkan kemudahan yang sesuai dengan gaya hidup modern yang terus berkembang, seperti pembayaran cepat dan berbagai opsi pembayaran digital. Teknologi seperti aplikasi *checkout* dan pembayaran otomatis telah membuat pengalaman berbelanja menjadi lebih cepat dan mudah. Semua faktor ini telah meningkatkan popularitas supermarket, yang tidak hanya memenuhi kebutuhan dasar, tetapi juga memberikan pengalaman berbelanja yang lebih nyaman (Rizkiyah et al., 2021).

Menurut (Sasanuma et al., 2022), meskipun terdapat kemajuan teknologi, tantangan dalam manajemen supermarket tetap ada, terutama dalam hal manajemen inventaris. Manajemen yang kompeten sangat penting untuk operasional supermarket guna memastikan produk tersedia dalam jumlah yang memadai dengan harga yang bersaing. Selain itu, manajemen yang baik memberikan kontribusi signifikan terhadap kepuasan pelanggan dan mengurangi potensi kerugian finansial.

Persediaan didefinisikan sebagai semua barang yang dijual atau tersedia untuk digunakan dalam jangka waktu tertentu, dan merupakan salah satu komponen penting dalam suatu organisasi ritel. Kurangnya pengelolaan persediaan yang tepat dapat melemahkan kemampuan supermarket dalam memenuhi harapan konsumen serta menurunkan tingkat kepercayaan dan loyalitas pelanggan. Oleh karena itu, pengelolaan inventaris yang efektif diperlukan untuk menjamin ketersediaan produk yang dibutuhkan, mengurangi pemborosan, dan mempertahankan persepsi positif di kalangan konsumen (Sridhar et al., 2021).

Manajemen persediaan yang efisien memiliki keterkaitan erat dengan prinsip-prinsip dasar dalam ajaran Islam, khususnya dalam hal efektivitas

operasional, pemanfaatan sumber daya secara optimal, dan penghindaran praktik pemborosan (israf) (M. Syarif Adi Pramana & Muhammad Saiful Khair, 2024). Konsep israf, yang berarti berlebihan atau melampaui batas, dilarang dalam Islam sebagaimana disebutkan dalam Surah Al-A'raf ayat 31 :

يُحِبُّ لَا يَبْتِغِي أَدَمَ خُدُوزَا زَيْنَتِكُمْ عِنْدَ كُلِّ مَسْجِدٍ وَكُلُوا وَاشْرَبُوا وَلَا تُسْرِفُوا إِنَّهُ  
الْمُسْرِفِينَ □ ٣١

*“Wahai anak cucu adam, pakailah pakaianmu yang indah pada setiap (memasuki) masjid dan makan serta minumlah, tetapi janganlah berlebihan. Sesungguhnya dia tidak menyukai orang-orang yang berlebihan.” (Q.S.Al-A'raf: 31).*

Dalam Tafsir Ibnu Katsir (Darus Salam, Jilid 3, Hal. 406), dijelaskan bahwa larangan berlebihan (israf) dalam ayat ini mencakup segala bentuk pemborosan dalam makan, minum, dan penggunaan harta. Ibnu Katsir menekankan bahwa pemborosan merupakan tindakan yang dapat membawa dampak buruk bagi individu dan masyarakat, seperti munculnya sifat tamak dan ketidaksyukuran terhadap nikmat Allah. Oleh karena itu, Islam mengajarkan keseimbangan dalam menikmati rezeki yang diberikan oleh Allah (Ibnu Katsir, 2020).

Salah satu pendekatan yang dapat membantu pengelolaan persediaan adalah metode *Simple Additive Weighting* (SAW). Metode ini memberikan kerangka sistematis untuk pengambilan keputusan berdasarkan berbagai kriteria, seperti biaya produksi, permintaan konsumen, dan pengendalian persediaan. Dengan memanfaatkan pendekatan tersebut, dapat dihasilkan simulasi berbasis game yang dirancang untuk mereplikasi operasional supermarket (Suryana & Purnama, 2022).

Simulasi manajemen persediaan telah menjadi alat efektif untuk memberikan pengalaman praktis dalam pengelolaan inventaris. Melalui simulasi,

peserta dapat memahami berbagai aspek operasional supermarket, seperti pengisian ulang produk, pengelolaan persediaan, dan pengendalian biaya, tanpa perlu terjun langsung ke lapangan. Selain itu, simulasi memungkinkan pembelajaran yang mendekati situasi nyata, membantu melatih keterampilan pengambilan keputusan, meminimalkan kesalahan dalam perencanaan, serta meningkatkan efisiensi operasional secara keseluruhan. Meskipun simulasi memiliki keterbatasan dalam merepresentasikan faktor eksternal yang kompleks, seperti dinamika pasar dan perilaku konsumen, pendekatan ini tetap relevan sebagai alat pelatihan dan pengembangan strategi manajemen persediaan (Agustina, 2023).

Pendekatan simulasi dalam pengelolaan inventaris di supermarket memiliki manfaat yang signifikan. Simulasi memungkinkan pengujian berbagai strategi operasional tanpa risiko nyata, membantu pengembang dalam mengeksplorasi metode pengelolaan sumber daya yang lebih efisien, serta memungkinkan pelaku industri untuk merancang sistem manajemen yang lebih responsif terhadap kebutuhan pasar. Dalam (Saputra et al., 2020), sistem simulasi yang dirancang menggunakan teknologi RFID pada sistem *Point of Sale* (POS) menunjukkan peningkatan efisiensi dalam transaksi dan pengelolaan inventaris, yang pada akhirnya mendukung optimalisasi layanan pelanggan dan operasional supermarket secara keseluruhan (Suryana & Purnama, 2022).

Dalam penelitian ini masalah dalam pengelolaan persediaan sering muncul ketika tidak adanya praktik langsung yang memungkinkan individu memahami dinamika pengelolaan inventaris secara menyeluruh. Minimnya pengalaman ini dapat menyebabkan kesalahan dalam perencanaan, pemborosan sumber daya, serta

ketidakefisienan operasional. Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan solusi berupa simulasi berbasis *game* yang dirancang untuk mereplikasi operasional supermarket. Simulasi ini memberikan pengalaman praktis yang mendekati situasi nyata dan membantu meningkatkan keterampilan pengambilan keputusan dalam mengelola inventaris secara efisien dan terstruktur.

## **1.2 Rumusan Masalah**

Adapun pernyataan masalah dalam penelitian ini adalah bagaimana penerapan *Simple Addive Weighting* (SAW) dalam manajemen stok pada simulasi operasional supermarket untuk menentukan prioritas penyimpanan dan pengadaan barang.

## **1.3 Batasan Masalah**

Berdasarkan latar belakang yang telah dijabarkan sebelumnya. Adapun Batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Penelitian ini menggunakan kriteria dasar seperti harga, permintaan pelanggan, stok, dan kadaluarsa dalam penerapannya, dengan tambahan aspek pengelolaan barang di supermarket yang mencakup gudang dan rak *display*.
2. Data yang digunakan dalam simulasi merupakan data berdasarkan interaksi pemain dalam simulasi, dimana data akan disederhanakan otomatis melalui sistem simulasi, menyesuaikan dengan permintaan pelanggan dalam permainan.

#### **1.4 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah menerapkan metode *Simple Additive Weighting* (SAW) dalam manajemen stok pada simulasi operasional supermarket untuk menentukan prioritas penyimpanan dan pengadaan barang secara optimal.

#### **1.5 Manfaat Penelitian**

Harapan dari penelitian yang dilakukan yakni dapat memberikan manfaat dan maslahat di kemudian hari, antara lain sebagai berikut: Berdasarkan latar belakang yang telah dijabarkan sebelumnya. Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Simulasi ini dapat menjadi alat edukasi yang membantu memahami proses pengambilan keputusan berbasis data dalam manajemen inventori.
2. Pemain bisa memahami prinsip SAW melalui pengalaman langsung dalam menentukan keputusan inventory yang optimal.
3. Penelitian ini memberikan contoh penerapan metode SAW dalam bentuk interaktif melalui simulasi berbasis game.

## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Dalam Penelitian yang berjudul “Aplikasi Game Simulasi Operasional Supermarket dengan *CreateJS* di PT. Harja Gunatama Lestari (Pratama & Suhendi, 2022) membahas pengembangan aplikasi permainan simulasi untuk melatih karyawan dalam menghadapi tantangan operasional di supermarket. Dengan pertumbuhan pesat bisnis ritel di Indonesia, supermarket perlu memberikan pelayanan yang cepat dan efisien untuk mengurangi keluhan pelanggan terkait waktu tunggu. Penelitian ini menggunakan metode *R&D* dengan pendekatan *Four-D (4D)*, yang mencakup tahap *Define* untuk menetapkan kebutuhan simulasi, *Design* untuk merancang permainan, *Develop* untuk menghasilkan simulasi menggunakan *CreateJS*, dan *Disseminate* untuk menyebarluaskan aplikasi ke karyawan. Hasil penelitian menunjukkan bahwa aplikasi simulasi yang dikembangkan dapat mengurangi kesalahan dalam pengelolaan barang dan meningkatkan pelayanan kasir, di mana karyawan yang dilatih dengan aplikasi ini menunjukkan peningkatan dalam menangani transaksi dan mengelola persediaan barang.

Dalam Penelitian yang berjudul "Simulasi Antrian dalam Optimalisasi Layanan di Supermarket Rita Pasaraya" (Prakoso et al., 2023) membahas penerapan teori antrian untuk meningkatkan efisiensi layanan di Supermarket Rita Pasaraya, yang merupakan salah satu supermarket terbesar di Purwokerto. Penelitian ini

berfokus pada analisis sistem antrian yang terjadi di kasir, di mana pengunjung sering mengalami waktu tunggu yang lama, terutama pada jam-jam sibuk. Dengan menggunakan model antrian *multi-channel single phase*, penelitian ini menganalisis pola kedatangan dan pelayanan pelanggan untuk menemukan sistem antrian yang ideal. Hasil observasi menunjukkan bahwa tingkat kedatangan pelanggan tertinggi terjadi pada pukul 15.30 - 16.00 WIB, dengan rata-rata kedatangan dan pelayanan sebanyak 9 orang. Penelitian ini merekomendasikan pengurangan jumlah kasir dari empat menjadi tiga untuk meningkatkan efektivitas pelayanan dan mengurangi idle time.

Dalam penelitian berjudul "*Simulation of Inventory Management Systems in Retail Stores: A Case Study*" (Sridhar et al., 2021) membahas pentingnya manajemen persediaan dalam sektor ritel, terutama dalam menghadapi ketidakpastian permintaan. Penelitian ini bertujuan untuk mengembangkan model simulasi yang dapat membantu dalam pengelolaan persediaan di toko ritel dengan menggunakan perangkat lunak simulasi Arena. Dalam studi kasus ini, data penjualan dan pembelian selama tiga tahun dikumpulkan untuk menganalisis pola permintaan dan karakteristik persediaan. Model simulasi yang dikembangkan memungkinkan peneliti untuk menguji berbagai kebijakan manajemen persediaan dan mengevaluasi dampaknya terhadap tingkat persediaan, biaya, dan kehilangan penjualan. Hasil penelitian menunjukkan bahwa penerapan sistem manajemen persediaan yang dioptimalkan dapat mengurangi tingkat persediaan hingga 40% dan kehilangan penjualan hingga 87%. Selain itu, sistem yang diusulkan juga dioptimalkan menggunakan modul *OptQuest* dalam perangkat lunak Arena, yang

lebih lanjut mengurangi tingkat persediaan hingga 73% dibandingkan dengan sistem tradisional.

Dalam Penelitian "*A Simulation Based Inventory Optimization Heuristic for a Retail Supply Chain*" (Mahmoudi et al., 2023) membahas pengembangan dan penerapan model simulasi untuk mengoptimalkan manajemen inventaris dalam rantai pasokan ritel. Penelitian ini berfokus pada seorang pemasok ritel yang menyediakan perlengkapan pelindung dengan permintaan yang tinggi dan variabilitas yang signifikan, serta waktu pengiriman yang tidak dapat diprediksi. Tujuan utama dari penelitian ini adalah untuk mengevaluasi kinerja sistem saat ini, menentukan apakah target perputaran inventaris yang ditetapkan oleh pemasok dapat dicapai, dan mengoptimalkan model untuk mendapatkan kuantitas pemesanan yang optimal guna meminimalkan total biaya. Metodologi yang digunakan melibatkan pembangunan model simulasi dalam perangkat lunak *ARENA*, yang mencakup analisis data historis untuk memperkirakan permintaan dan menerapkan model *Newsvendor* untuk menangani variabilitas permintaan. Penelitian ini juga mempertimbangkan biaya penyimpanan, biaya kekurangan, dan biaya pemesanan dalam analisisnya. Hasil simulasi menunjukkan bahwa model yang dioptimalkan dapat memberikan kuantitas pemesanan yang lebih efisien dan mengurangi total biaya dibandingkan dengan sistem yang ada.

Dalam penelitian yang berjudul "*Analisis Sistem Pendukung Keputusan Menentukan Produk Terlaris dengan Metode Simple Additive Weighting*" (Aprilia & Wahidin, 2022) membahas penerapan metode *SAW* dalam sistem pendukung keputusan untuk menentukan produk terlaris di suatu perusahaan. Penelitian ini

bertujuan untuk membantu manajemen dalam mengambil keputusan yang lebih baik terkait pemilihan produk yang akan dipasarkan, dengan mempertimbangkan berbagai kriteria yang relevan. Metodologi yang digunakan meliputi identifikasi kriteria yang mempengaruhi penjualan produk, seperti harga, kualitas, popularitas, dan margin keuntungan, dengan data yang dikumpulkan dari berbagai sumber, termasuk data penjualan historis dan survei pelanggan. Metode SAW diterapkan untuk memberikan bobot pada setiap kriteria dan menghitung nilai akhir untuk setiap produk berdasarkan kriteria yang telah ditentukan. Hasil analisis menunjukkan bahwa metode SAW efektif dalam mengidentifikasi produk terlaris, dengan memberikan peringkat yang jelas berdasarkan kriteria yang relevan.

Dalam penelitian berjudul "*Decision Support System for Election Suppliers of Goods Using the Simple Additive Weighting (SAW) Method*" (Khairani et al., 2024), dibahas penerapan metode SAW dalam sistem pendukung keputusan untuk menentukan supplier barang pada sebuah toko ritel bernama *Diva Store*. Penelitian ini bertujuan untuk membantu manajemen toko dalam memilih supplier yang optimal berdasarkan sejumlah kriteria penting, di antaranya adalah harga pembelian, kualitas barang, jadwal pengiriman, masa berlaku barang, dan layanan garansi. Metode yang digunakan meliputi pengumpulan data primer melalui wawancara dengan manajer toko dan data sekunder dari dokumen terkait. Metode SAW diterapkan dengan memberikan bobot pada masing-masing kriteria, lalu melakukan perhitungan nilai akhir setiap alternatif supplier berdasarkan bobot tersebut. Hasil penelitian menunjukkan bahwa metode SAW efektif dalam memberikan rekomendasi supplier terbaik dengan tingkat akurasi sistem mencapai

73,77% berdasarkan pengujian menggunakan metode *confusion matrix*, sehingga mampu mempermudah dan mempercepat proses pengambilan keputusan yang sebelumnya dilakukan secara manual dan subjektif.

Dalam penelitian yang berjudul “*Leveraging the Decision Support System and Simple Additive Weighting Method for Optimal Retail Location Identification*” (Harahap et al., 2022) dibahas penerapan metode SAW dalam sistem pendukung keputusan untuk menentukan lokasi ritel baru yang paling optimal. Penelitian ini bertujuan membantu manajemen dalam memilih lokasi toko berdasarkan kriteria kepadatan penduduk, kedekatan dengan pesaing, rata-rata pendapatan, dan biaya sewa. Metodologi yang digunakan meliputi penentuan bobot tiap kriteria (0,4 untuk kepadatan penduduk; 0,3 kedekatan pesaing; 0,2 rata-rata pendapatan; 0,1 biaya sewa), pengumpulan data kuantitatif untuk setiap alternatif lokasi, serta perhitungan skor tertimbang untuk masing-masing alternatif dengan SAW. Hasil analisis menunjukkan bahwa Lokasi A memperoleh skor tertinggi (14.200), diikuti Lokasi B (13.800) dan Lokasi C (13.500), sehingga Lokasi A direkomendasikan sebagai lokasi ritel baru karena memberikan nilai manfaat tertinggi berdasarkan kriteria yang telah ditetapkan.

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian ini

No	Peneliti	Judul Penelitian	Metode	Hasil
1	Angga Pratama, Suhendi, Mukminin	Aplikasi Game Simulasi Operasional Supermarket dengan <i>CreateJS</i> di PT. Harja Gunatama Lestari	<i>R&amp;D (Four-D: Define, Design, Develop, Disseminate)</i>	Simulasi berbasis <i>CreateJS</i> meningkatkan efisiensi operasional, mengurangi kesalahan pengelolaan barang, dan mempercepat layanan kasir.
2	Indro Prakoso, Amanda Sofiana,	Simulasi Antrian dalam Optimalisasi Layanan di Supermarket Rita Pasaraya	<i>Multi-channel Single Phase Queue</i>	Rekomendasi mengurangi kasir dari empat menjadi tiga untuk meningkatkan efektivitas layanan.

No	Peneliti	Judul Penelitian	Metode	Hasil
	Sarah Nurmalawati			
3	Indro Prakoso, Amanda Sofiana, Sarah Nurmalawati	<i>Simulation of Inventory Management Systems in Retail Stores: A Case Study</i>	Simulasi Software Arena	Pengurangan tingkat persediaan hingga 73% dan kehilangan penjualan hingga 87% dengan modul <i>OptQuest</i> .
4	Vedurmu di Shruti Chandrik a	<i>A Simulation Based Inventory Optimization Heuristic for a Retail Supply Chain</i>	Simulasi Software Arena	Optimasi kuantitas pemesanan yang lebih efisien, mengurangi total biaya, dan meningkatkan perputaran inventaris.
5	Siagian, Kifti, Hutahaean	Analisis Sistem Pendukung Keputusan Menentukan Produk Terlaris dengan Metode <i>Simple Additive Weighting</i>	<i>Simple Additive Weighting</i>	Metode SAW efektif dalam memberikan peringkat produk berdasarkan kriteria yang relevan.
6	Gusti Aprilandri, Rumini, Dina Maulina	<i>Decision Support System for Election Suppliers of Goods Using the Simple Additive Weighting (SAW) Method</i>	<i>Simple Additive Weighting</i>	Hasil pengujian menunjukkan akurasi sistem 73,77% dengan persentase error 26,23%, serta perhitungan manual sama persis dengan perhitungan sistem.
7	M. Ade Kurnia Harahap, Hardisal, Ahyuna, Robbi	<i>Leveraging the Decision Support System and Simple Additive Weighting Method for Optimal Retail Location Identification</i>	<i>Simple Additive Weighting</i>	Analisis merekomendasikan lokasi toko sebagai tempat terbaik untuk membuka cabang baru berdasarkan pendekatan data-driven dan objektif.
8	M.Tazkia Ikhsanul Ma'arif	Penerapan Metode <i>Simple Additive Weighting</i> dalam Simulasi Operasional Supermarket	<i>Simple Additive Weighting</i>	-

## 2.2 Manajemen Barang

Menurut (Selina et al., 2022), komponen utama dalam sistem inventaris meliputi pencatatan barang masuk dan keluar, sistem pemantauan stok, serta pelaporan data persediaan secara akurat dan real-time. Penggunaan teknologi informasi dalam sistem inventaris memungkinkan pencatatan barang yang lebih cepat dan akurat dibandingkan metode konvensional, sehingga mengurangi kesalahan dalam pencatatan stok dan mempercepat proses pengambilan Keputusan. Pertama, stock manajemen adalah proses yang melibatkan pemantauan dan pengaturan jumlah barang yang tersedia. Hal ini mencakup penentuan tingkat minimum dan maksimum untuk setiap produk agar tidak terjadi kelebihan atau kekurangan stok.

Kedua, *booking* merupakan prosedur untuk menentukan kapan dan berapa banyak barang yang harus dipesan, yang sering kali didasarkan pada analisis data penjualan historis untuk memprediksi permintaan di masa depan. Proses ini penting untuk menjaga ketersediaan produk dan menghindari kehilangan penjualan akibat kehabisan stok (Jha et al., 2025).

Selanjutnya, *reception and storage* barang adalah langkah penting dalam sistem *inventory*, di mana barang yang dipesan diterima dan disimpan di lokasi yang sesuai dalam supermarket. Proses ini harus dilakukan dengan hati-hati untuk memastikan bahwa barang tidak rusak dan dapat diakses dengan mudah saat dibutuhkan. *inventory control* adalah komponen lain yang esensial, yang mencakup metode untuk mengawasi pergerakan barang, termasuk penghitungan fisik persediaan dan penggunaan teknologi modern seperti barcode atau sistem *RFID*

untuk melacak produk secara *real-time*. Pengendalian yang baik membantu dalam mengidentifikasi barang yang bergerak cepat, lambat, atau mendekati kadaluarsa, sehingga manajemen dapat mengambil tindakan yang tepat. Akhirnya, *reporting and analysis* adalah bagian dari sistem inventory yang menyediakan informasi penting tentang status persediaan. Laporan ini mencakup data tentang tingkat persediaan, kecepatan pergerakan barang, dan tren penjualan, yang semuanya membantu manajemen dalam membuat keputusan yang lebih baik (Alberto, 2024).

Sistem pengendalian persediaan barang laris memberikan sejumlah manfaat penting yang berkontribusi pada keberhasilan operasional sebuah mini market. Pertama, efisiensi operasional meningkat secara signifikan karena sistem ini mengurangi waktu dan tenaga yang diperlukan dalam pengelolaan stok, memungkinkan staf untuk fokus pada tugas-tugas lain yang lebih strategis. Selain itu, sistem ini juga berperan dalam pengurangan biaya, dengan meminimalkan biaya penyimpanan dan mengurangi kerugian akibat barang yang kedaluwarsa atau tidak terjual, sehingga meningkatkan profitabilitas. Pengendalian persediaan yang baik memastikan ketersediaan barang yang memadai, sehingga pelanggan tidak mengalami kekecewaan akibat kehabisan stok. Dengan demikian, penerapan sistem pengendalian persediaan yang efektif tidak hanya meningkatkan efisiensi dan mengurangi biaya, tetapi juga berkontribusi pada peningkatan layanan pelanggan, menciptakan pengalaman belanja yang lebih memuaskan (M. I. H. Saputra & Nugraha, 2020).

### 2.3 Game Simulasi

Game simulasi telah menjadi alat yang efektif dalam pendidikan, memungkinkan siswa untuk belajar melalui pengalaman interaktif. Penelitian terbaru menunjukkan bahwa game edukasi, dapat meningkatkan keterampilan kognitif dan motivasi belajar siswa. Misalnya, game simulasi yang dirancang untuk mengasah keterampilan menulis dan perhitungan dasar menunjukkan hasil yang positif dalam meningkatkan pemahaman materi. Selain itu, pengembangan game berbasis pendidikan, seperti yang diusulkan dalam penelitian tentang game *Defense Tsunami*, menunjukkan potensi untuk memperluas cakupan pembelajaran dengan mengintegrasikan elemen mitigasi bencana ke dalam gameplay, sehingga siswa tidak hanya belajar konten akademis tetapi juga keterampilan praktis yang relevan dengan situasi dunia nyata (Dwiawinta & Rukmi, 2022)

Berdasarkan (Normah et al., 2022), game simulasi dapat dikategorikan ke dalam beberapa jenis yang menawarkan berbagai pengalaman interaktif bagi pemain. Salah satu kategori yang banyak digunakan adalah serious games, yang dirancang untuk tujuan edukatif atau pelatihan, seperti game simulasi perakitan komputer yang membantu pemain memahami komponen dan cara merakit komputer secara virtual.

Selain itu, terdapat management simulation, yang memungkinkan pemain untuk mengelola sumber daya dan membuat keputusan strategis, seperti dalam game simulasi bisnis. Life simulation memberikan pengalaman dalam mengelola kehidupan sosial, sedangkan technical simulation digunakan untuk memberikan pengalaman dalam bidang teknis, seperti merakit atau memperbaiki perangkat

elektronik. Training simulation juga digunakan dalam berbagai industri untuk meningkatkan keterampilan teknis melalui simulasi interaktif (Ionescu-Feleagă et al., 2025).

#### **2.4 *Simple Additive Weighting (SAW)***

Metode *SAW* adalah salah satu teknik yang banyak digunakan dalam pengambilan keputusan multikriteria (*Multi-Criteria Decision Making - MCDM*) karena kesederhanaannya dan kemampuannya memberikan hasil yang objektif. Metode ini bekerja dengan memberikan bobot pada setiap kriteria yang dipertimbangkan dalam proses pengambilan keputusan, kemudian menghitung nilai total tertimbang untuk setiap alternatif yang tersedia (Wijana et al., 2024).

*SAW* sangat efektif dalam situasi yang melibatkan banyak kriteria, di mana keputusan harus dibuat dengan mempertimbangkan alternatif yang memiliki karakteristik berbeda. Metode ini mempermudah proses dengan mengolah data yang relevan sehingga pengambil keputusan dapat menentukan pilihan secara objektif dan terstruktur. Dalam praktiknya, metode ini melibatkan beberapa langkah penting, termasuk identifikasi kriteria yang akan digunakan, penentuan bobot untuk masing-masing kriteria, serta normalisasi data untuk memastikan bahwa semua nilai dapat dibandingkan secara langsung. Normalisasi ini sangat penting karena data yang diambil dari berbagai sumber atau kriteria mungkin memiliki skala yang berbeda, sehingga tanpa normalisasi, perbandingan antar alternatif akan menjadi tidak akurat (Taherdoost, 2023).

Metode *SAW* juga dikenal dengan istilah penjumlahan berbobot, yang menekankan pentingnya memberikan bobot yang sesuai untuk setiap kriteria

berdasarkan tingkat kepentingannya. Bobot ini biasanya ditentukan oleh pengambil keputusan, yang mempertimbangkan preferensi dan prioritas yang ada dalam pengambilan keputusan yang sedang dilakukan. Setelah bobot ditetapkan, proses penilaian dilanjutkan dengan menghitung nilai preferensi untuk setiap alternatif. Nilai preferensi ini diperoleh dengan menjumlahkan hasil kali antara nilai normalisasi dari setiap alternatif dan bobot dari kriteria yang bersangkutan. Dengan cara ini, *SAW* memungkinkan pengambil keputusan untuk secara objektif menilai dan membandingkan alternatif berdasarkan kriteria yang telah ditentukan, sehingga menghasilkan alternatif terbaik yang dapat dipilih (Vafaei et al., 2021). Berikut penjelasan tentang cara kerja *SAW* :

#### **2.4.1 Identifikasi Kriteria dan Alternatif**

Alternatif adalah pilihan atau opsi yang tersedia untuk dipertimbangkan dalam proses pengambilan keputusan. Alternatif merupakan berbagai solusi atau produk yang akan dievaluasi berdasarkan kriteria yang telah ditentukan. Sedangkan kriteria adalah parameter atau faktor yang digunakan untuk menilai dan membandingkan alternatif. Kriteria ini mencerminkan aspek-aspek penting yang harus diperhatikan dalam pengambilan keputusan. Contohnya, dalam memilih mobil, Mobil A, B, dan C adalah alternatif, sedangkan harga, konsumsi bahan bakar, kenyamanan, keamanan, dan performa adalah kriteria (M. I. H. Saputra & Nugraha, 2020).

Untuk menentukan alternatif dan kriteria dalam pengambilan keputusan menggunakan metode *SAW*, langkah pertama adalah mengidentifikasi tujuan dari keputusan yang akan diambil. Selanjutnya, lakukan riset pasar untuk

mengumpulkan informasi mengenai berbagai opsi yang tersedia. Ini dapat dilakukan melalui observasi. Setelah mengidentifikasi beberapa pilihan, buatlah daftar pilihan yang relevan berdasarkan hasil riset. Setiap alternatif kemudian dievaluasi berdasarkan kriteria yang telah ditetapkan sebelumnya, dengan mengumpulkan data tentang biaya, efektivitas, dan dampak lingkungan dari masing-masing alternatif. Kategori alternatif juga dapat dilakukan untuk memudahkan analisis, misalnya dengan mengelompokkan alternatif berdasarkan jenis metode pemeliharaan (Riki Sulistio et al., 2024).

Setelah alternatif ditentukan, langkah berikutnya adalah mengidentifikasi kriteria yang akan digunakan untuk mengevaluasi setiap alternatif. Libatkan pihak-pihak yang berkepentingan dalam proses ini untuk mendapatkan perspektif yang lebih luas tentang apa yang dianggap penting. Lakukan sesi brainstorming untuk mengumpulkan semua kriteria yang mungkin, seperti harga, konsumsi bahan bakar, kenyamanan, keamanan, dan performa. Setelah itu, evaluasi setiap kriteria untuk memastikan bahwa kriteria tersebut relevan, dapat diukur, dan tidak saling tumpang tindih. Kategorisasikan kriteria ke dalam kelompok yang relevan dan prioritaskan dengan memberikan bobot berdasarkan tingkat kepentingannya (Wijanarko et al., 2024).

#### **2.4.2 Alur Matrix Keputusan**

Matriks keputusan adalah representasi numerik yang digunakan untuk membandingkan berbagai alternatif berdasarkan kriteria yang telah ditentukan. Matriks keputusan memungkinkan pengambil keputusan untuk melihat

perbandingan secara langsung dan memberikan bobot pada setiap kriteria dan alternatif (Khoiroh et al., 2023).

Matriks keputusan  $\chi$  berisi nilai – nilai dari setiap alternatif  $\chi_c$  untuk masing – masing kriteria  $\kappa$  Jika terdapat  $\chi_c$  alternatif dan  $j$  kriteria, maka matriks keputusan ini dapat dituliskan sebagai berikut :

$$\chi = \begin{bmatrix} \chi_{11} & \chi_{12} & \cdots & \chi_{1\kappa} \\ \chi_{21} & \chi_{22} & \cdots & \chi_{2\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{C\kappa} & \chi_{C\kappa} & \cdots & \chi_{C\kappa} \end{bmatrix} \quad (2.1)$$

Disini C adalah jumlah nilai dari alternatif  $\chi$  sedangkan  $\kappa$  adalah jumlah nilai pada kriteria. Misalnya,  $C_{12}$  berarti nilai alternatif pertama pada kriteria kedua (Terttiaavini, 2024).

Setelah matriks perbandingan diisi, untuk membuat nilai dari berbagai kriteria berada pada skala yang sama, matriks keputusan ternormalisasi. Proses normalisasi berbeda berdasarkan jenis kriteria

A. Kriteria Manfaat (Benefit) :

$$R_{C\kappa} = \frac{\chi_{C\kappa}}{\max(\chi_{C\kappa})} \quad (2.2)$$

$\chi_{C\kappa}$  adalah nilai normalisasi dari alternatif ke -C pada kriteria ke - $\kappa$ . Nilai maximum  $\max \chi_{C\kappa}$  dari semua alternatif pada kriteria ke - $\kappa$ .

B. Kriteria Biaya (Cost) :

$$R_{C\kappa} = \frac{\min(\chi_{C\kappa})}{\chi_{C\kappa}} \quad (2.3)$$

$R_{C\kappa}$  adalah nilai normalisasi dari alternatif ke -C pada kriteria ke  $-\kappa$ . Nilai minimum  $\min \chi_{C\kappa}$  dari semua alternatif pada kriteria tersebut dibagi dengan nilai  $\chi_{C\kappa}$  dari alternatif ke -C. Hasil dari normalisasi ini adalah matriks normalisasi  $R_{C\kappa}$  dimana semua nilai berada diantara 0 dan 1 (Nuna et al., 2024).

Setelah normalisasi, bobot untuk setiap kriteria  $W_\kappa$  diterapkan pada nilai normalisasi untuk menghitung skor akhir dari setiap alternatif. Bobot  $W_\kappa$  mencerminkan tingkat kepentingan kriteria tersebut dan biasanya dijumlahkan hingga total 1. Nilai akhir  $v_C$  dari setiap alternatif dihitung sebagai berikut :

$$v_C = \sum_{\kappa=1}^{\kappa} W_\kappa \cdot R_{C\kappa} \quad (2.4)$$

$v_C$  adalah nilai akhir atau skor total untuk alternatif ke -C. Bobot  $W_\kappa$  dari setiap kriteria ke  $-\kappa$  dikalikan dengan nilai normalisasi  $R_{C\kappa}$  kemudian semua hasil perkalian ini dijumlahkan untuk mendapatkan  $v_C$ . Setelah mendapat nilai  $v_C$  untuk setiap alternatif, bandingkan nilai akhir atau  $v_C$  dari semua alternatif. Alternatif dengan nilai  $v_C$  tertinggi dianggap sebagai pilihan terbaik (Safitri et al., 2021).

## 2.5 Unity

Unity adalah sebuah game engine yang dikembangkan oleh Unity Technologies, yang dirilis pertama kali pada bulan Juni 2005. Dirancang sebagai platform pengembangan game lintas platform yang memungkinkan pengembang untuk membuat berbagai jenis permainan, termasuk permainan dua dimensi (2D), tiga dimensi (3D), *augmented reality* (AR), dan *virtual reality* (VR). Salah satu keunggulan utama adalah kemampuannya untuk mendukung lebih dari 20 platform,

termasuk PC, konsol, dan perangkat mobile, sehingga memudahkan pengembang untuk mendistribusikan game mereka ke berbagai perangkat (Hussain et al., 2020).

*Unity Technologies* meluncurkan pada bulan Juni 2005, pada konferensi *Apple Worldwide Developers Conference* sebagai game engine eksklusif untuk *Mac OS X*. Sejak saat itu, telah mengalami banyak perkembangan, termasuk peluncuran versi-versi baru yang memperkenalkan fitur-fitur canggih. Pada tahun 2018, diperluas untuk mendukung lebih dari 20 platform, menjadikannya salah satu engine game paling fleksibel di pasar. Perkembangan ini mencerminkan komitmen Unity untuk beradaptasi dengan kebutuhan industri yang terus berubah (Neuman-Donihu et al., 2023).

Unity memiliki arsitektur yang terdiri dari beberapa komponen utama, termasuk Editor, yang merupakan antarmuka pengguna grafis untuk pengembangan game. Di dalam Unity, *GameObjects* adalah entitas dasar yang dapat memiliki berbagai komponen yang memberikan fungsionalitas, seperti fisika, rendering, dan interaksi. Komponen ini memungkinkan pengembang untuk membangun dunia game yang kompleks dengan cara yang modular dan terstruktur (Ullmann et al., 2025).

Unity mendukung berbagai platform, termasuk standalone (.exe), web, *Android*, *iOS*, *Xbox*, *PlayStation*, serta pengembangan game berbasis *Augmented Reality (AR)* dan *Virtual Reality (VR)*. Selain itu, dapat digunakan secara online maupun offline. Beberapa fitur utama meliputi pengembangan game 2D dan 3D, dukungan untuk game *FPS* dan online, kompatibilitas dengan berbagai bahasa pemrograman seperti *C#*, *JavaScript*, dan *Boo*, serta integrasi dengan berbagai

format file seperti *3ds*, *obj*, dan *fbx*. Unity juga memiliki *Asset Store* yang menyediakan berbagai sumber daya untuk membantu pengembang dalam proses pembuatan game (Saefudin & Sudjiran, 2023).

Unity telah digunakan dalam berbagai proyek nyata di industri game dan di luar itu. Misalnya, banyak game populer seperti "*Hollow Knight*" dan "*Inside*" dibangun menggunakan Unity. Selain itu, Unity juga digunakan dalam industri film untuk menciptakan visualisasi dan simulasi yang interaktif, serta dalam arsitektur untuk presentasi desain yang lebih menarik. Unity memiliki komunitas pengembang yang besar dan aktif, dengan banyak forum, grup diskusi, dan sumber daya online yang tersedia untuk membantu pengembang (Hussain et al., 2020).

Unity *Asset Store* juga menyediakan berbagai aset dan alat yang dapat digunakan untuk mempercepat proses pengembangan. Dukungan ini sangat penting, terutama bagi pengembang pemula yang mungkin memerlukan bimbingan dalam mempelajari platform. Melihat tren masa depan, Unity terus beradaptasi dengan perkembangan teknologi terbaru, termasuk integrasi dengan *AI* dan peningkatan kemampuan untuk pengembangan *AR/VR*. Dengan semakin populernya teknologi ini, *Unity* berpotensi untuk tetap menjadi salah satu pilihan utama bagi pengembang game dan aplikasi interaktif di masa depan (Ahamed et al., 2020).

## **2.6 *Finite State Machine (FSM)***

*FSM* adalah model matematis yang digunakan untuk menggambarkan perilaku sistem yang dapat berada dalam sejumlah keadaan terbatas. Dalam *FSM*, sistem dapat berada dalam satu keadaan pada satu waktu, dan transisi antara

keadaan-keadaan tersebut terjadi berdasarkan input atau kondisi tertentu. *FSM* terdiri dari keadaan (*states*), transisi (*transitions*), dan input (*inputs*) yang memicu transisi antara keadaan (Verma et al., 2023). Terdapat dua jenis *FSM*, yaitu :

- a. *Deterministic Finite State Machines (DFSM)*, di mana untuk setiap keadaan dan input tertentu hanya ada satu transisi yang mungkin,
- b. *Nondeterministic Finite State Machines (NFSM)*, di mana untuk keadaan dan input tertentu mungkin ada beberapa transisi yang mungkin.

*FSM* merupakan metode yang banyak digunakan dalam pengembangan game untuk mengelola perilaku karakter dan objek dalam permainan. Dalam game, memungkinkan pendefinisian berbagai keadaan yang dapat diambil oleh karakter non-pemain (*NPC*), seperti "berjalan", "berlari", "menyerang", dan "bersembunyi", dengan transisi antar keadaan ditentukan oleh kondisi permainan, seperti input dari pemain atau perubahan lingkungan. Keunggulan penggunaan *FSM* dalam game meliputi kemudahan dalam mengelola logika kompleks dengan memecahnya menjadi keadaan-keadaan sederhana yang saling terhubung, peningkatan keterbacaan kode yang mempermudah pengembang dalam memahami dan memodifikasi perilaku karakter, serta kemudahan dalam pengujian dan pemeliharaan karena setiap keadaan dan transisinya dapat diuji secara terpisah. Dengan menggunakan *FSM*, pengembang dapat dengan jelas memisahkan logika untuk setiap keadaan, sehingga mempermudah dalam pengembangan dan debugging, serta meningkatkan efisiensi dalam merancang perilaku *NPC* secara lebih terstruktur dan responsif terhadap perubahan dalam permainan (Asrianda & Zulfadli, 2022).

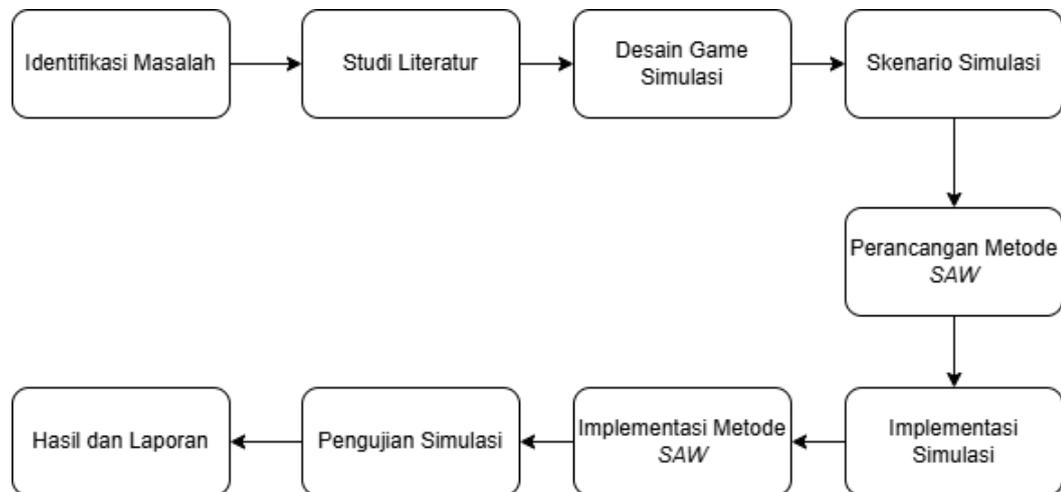
*FSM* adalah model matematika yang digunakan untuk merepresentasikan dan mengendalikan aliran eksekusi berdasarkan serangkaian keadaan dan transisi yang ditentukan. Dalam pengembangan game menggunakan Unity, *FSM* sering diimplementasikan untuk mengatur perilaku karakter atau objek lainnya. Pengembang dapat menggunakan skrip *C#* untuk mendefinisikan berbagai keadaan dan logika transisi antar keadaan tersebut. Unity juga menyediakan alat seperti *Animator Controller* yang memungkinkan pengaturan transisi antar animasi berdasarkan keadaan tertentu. Dengan menerapkannya, karakter dalam game dapat beradaptasi dengan cepat terhadap perubahan situasi, menciptakan pengalaman bermain yang lebih dinamis dan responsif. Secara keseluruhan, penerapannya dalam pengembangan game di Unity memberikan struktur yang jelas dan fleksibilitas yang diperlukan untuk menciptakan gameplay yang menarik dan interaktif (Wahyudi, Saputra, Nugroho, 2022).

## BAB III

### DESAIN DAN IMPLEMENTASI

#### 3.1 Desain Alur Penelitian

Penelitian ini menggunakan desain penelitian kuantitatif dengan pendekatan simulasi untuk mengimplementasikan metode SAW dalam operasional supermarket. Desain penelitian difokuskan pada pengelolaan barang dengan rekomendasi utama, yaitu barang yang perlu direstock. Berdasarkan hasil perhitungan harian pada simulasi, Pendekatan ini bertujuan untuk mengaplikasikan rekomendasi kepada simulasi yang dibangun. Diagram alur penelitian ditunjukkan pada gambar untuk memberikan gambaran visual mengenai tahapan penelitian :



Gambar 3.1 Alur Penelitian

##### a. Identifikasi Masalah

Langkah awal penelitian dimulai dengan perencanaan dan identifikasi masalah yang akan diselesaikan, yaitu pengelolaan stok di supermarket menggunakan simulasi dalam game.

b. Studi Literatur

Pada tahap ini dilakukan studi literatur terkait topik-topik yang berhubungan dengan penelitian untuk memahami pembahasan tersebut lebih dalam. Topik-topik yang dimaksud yaitu : perancangan simulasi berbasis game, SAW, dan juga penelitian terkait.

c. Desain Game Simulasi

Pada tahap ini, merancang lingkungan game 3D berbasis mobile untuk simulasi supermarket. Pada tahap ini, berbagai elemen simulasi didesain, yaitu Environment, sistem operasional.

d. Perancangan Metode SAW

Pada tahap ini, berfokus pada penerapan metode SAW dalam simulasi, termasuk : menentukan kriteria, bobot, normalisasi, dan melakukan perhitungan akhir

e. Implementasi Simulasi

Pada tahap ini, konsep desain game yang telah direncanakan diwujudkan ke dalam bentuk game. Proses ini akan dirancang menggunakan game Unity engine 3D.

f. Implementasi Metode SAW

Selanjutnya, rancangan simulasi ditambahkan sistem perhitungan metode SAW dengan pengambilan data dimana data ini diperoleh secara dinamis melalui interaksi pemain di dalam simulasi. Hasil nya yaitu berupa kandidat rekomendasi barang untuk direstock.

g. Pengujian Simulasi

Pada tahap ini, sistem akan diuji untuk memastikan akurasi, efektivitas, dan keberlanjutan sistem dalam jangka panjang. Pengujian dilakukan dengan melibatkan pemain yang berperan sebagai manajer supermarket. Evaluasi dilakukan menggunakan metode *System Usability Scale (SUS)*.

h. Hasil dan Laporan

Setelah semuanya telah selesai, hasil akan dibuat berdasarkan penelitian, yang nantinya akan dilaporkan dalam bentuk laporan. Kesimpulan dan uji coba simulasi akan dimasukkan disini.

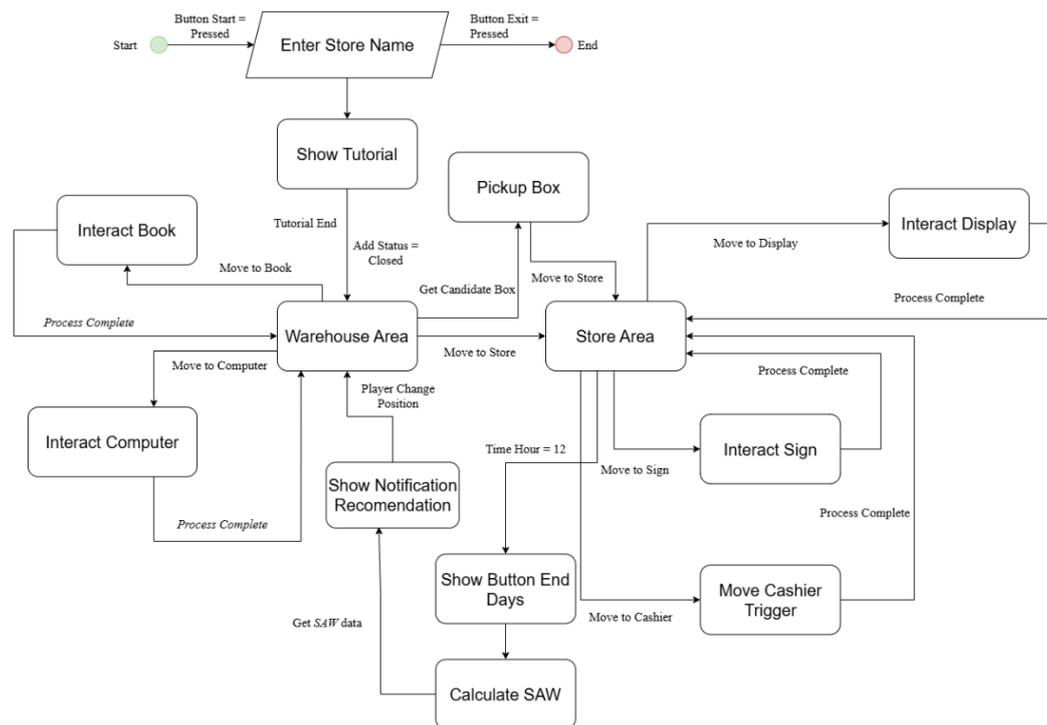
### 3.2 Skenario Simulasi

Selain itu, perancangan mencakup sistem interaksi lingkungan, di mana pemain dapat berinteraksi dengan elemen-elemen seperti komputer toko untuk melakukan pengaturan, rak untuk menata barang, dan gudang untuk mengambil stok. Terdapat juga perancangan alur waktu simulasi, yang terdiri dari siklus harian: toko dibuka untuk melayani pelanggan, diikuti dengan penutupan toko, analisis data, dan pemberian rekomendasi sebelum memulai hari berikutnya.

#### 3.2.1 FSM Simulasi

Pada Simulasi ini, *Finite State Machine* dirancang untuk merepresentasikan alur skenario bermain dan integrasi perhitungan *SAW* di setiap siklus hari. *FSM* diawali pada *state idle*. Ketika simulasi baru dimulai, kemudian berpindah ke *CustomerSpawn* untuk memunculkan *NPC* pembeli. Setiap *NPC* akan melakukan *patrol* serta mengecek apakah toko *status* nya buka atau tidak. Jika benar maka mereka masuk toko dan mereka meninjau daftar item dan menilai kriteria untuk

memilih alternatif barang. Setelah alternatif barang diperoleh lalu mereka berpindah ke *Cashier* untuk menyelesaikan pembayaran. Disini lah alur ambil data alternatif untuk menghitung *SAW* pada akhir hari. Ketika alternatif barang di scan data kriteria diambil dan disimpan berulang terus sampai *EndDays*. Data yang diambil akan dihitung berdasarkan metode *SAW* dan akan ditampilkan pada laporan harian. Hasil dari metode tersebut akan menghasilkan rekomendasi barang mana yang perlu direstock pada hari selanjutnya serta barang mana yang menjadi kandidat untuk diperhatikan stok nya. Untuk lebih jelasnya bisa dilihat pada gambar *FSM* berikut ini :



Gambar 3.2 *FSM* Skenario Simulasi

Berdasarkan Gambar 3.2, alur simulasi operasional supermarket dimulai ketika pemain menekan tombol Start dan diarahkan ke proses *Enter Store Name*

untuk menentukan nama toko yang akan dikelola. Setelah nama toko dimasukkan, permainan dilanjutkan ke tahap *Show Tutorial* yang memberikan petunjuk dasar mengenai mekanisme permainan. Apabila tutorial selesai, status permainan diperbarui menjadi *Closed*, dan pemain berpindah ke area gudang (*Warehouse Area*) sebagai area utama awal permainan.

Di dalam *Warehouse Area*, pemain memiliki beberapa opsi interaksi, yaitu *Interact Book* dan *Interact Computer*, di mana masing-masing interaksi tersebut memiliki proses *FSM* tersendiri yang akan dijelaskan pada bagian berikutnya. Selain itu, pemain juga dapat berpindah ke *Store Area* untuk melakukan proses *Pickup Box* yang akan mendapatkan data calon box, lalu diarahkan ke *Interact Display* untuk menata barang pada display toko. Setiap proses interaksi yang terjadi akan kembali ke *Store Area* setelah proses dinyatakan complete.

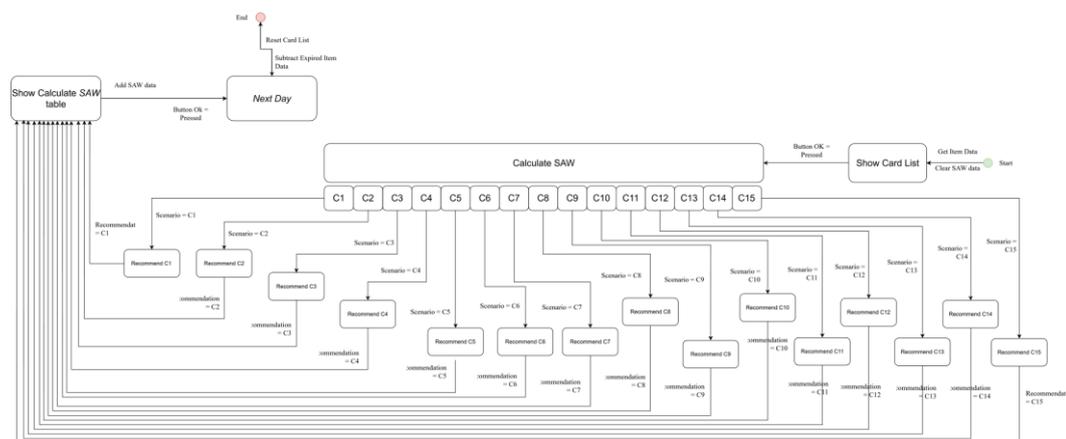
Pada saat waktu permainan mencapai jam 12 (*Time Hour = 12*), sistem akan menampilkan notifikasi rekomendasi yang berisi hasil analisis metode *Simple Additive Weighting (SAW)*. Rekomendasi ini ditampilkan melalui proses *Show Notification Recommendation* yang mengambil data *SAW*, lalu pemain dapat mengakhiri hari melalui *Show Button End Days*. Setelah tombol akhir hari muncul, sistem akan menghitung skor secara otomatis menggunakan *Calculate SAW*.

Selain itu, di dalam *Store Area*, pemain juga dapat berinteraksi dengan *Interact Sign* untuk melihat tanda-tanda penting dalam operasional toko, serta interaksi dengan *Move Cashier Trigger* sebelum proses penutupan hari dilakukan. Setiap interaksi dalam *FSM* ini saling terhubung dan kembali ke *Store Area* setelah

proses selesai. Adapun proses *Move to Sign*, *Move to Display*, dan *Move to Cashier* terjadi sesuai urutan interaksi yang dipilih pemain selama simulasi berlangsung.

Secara keseluruhan, simulasi ini membentuk sebuah siklus manajemen operasional toko harian yang dimulai dari pengelolaan nama toko, penataan stok barang, pengambilan keputusan strategis melalui rekomendasi SAW, hingga pengaturan posisi kasir dan penutupan hari. Setiap proses interaksi seperti *Interact Computer*, *Interact Book*, *Interact Display*, *Interact Sign*, *Interact Cashier* akan dijelaskan lebih lanjut pada *FSM* tambahan yang mendetailkan proses di dalam masing-masing interaksi tersebut.

a. *FSM* Perhitungan SAW



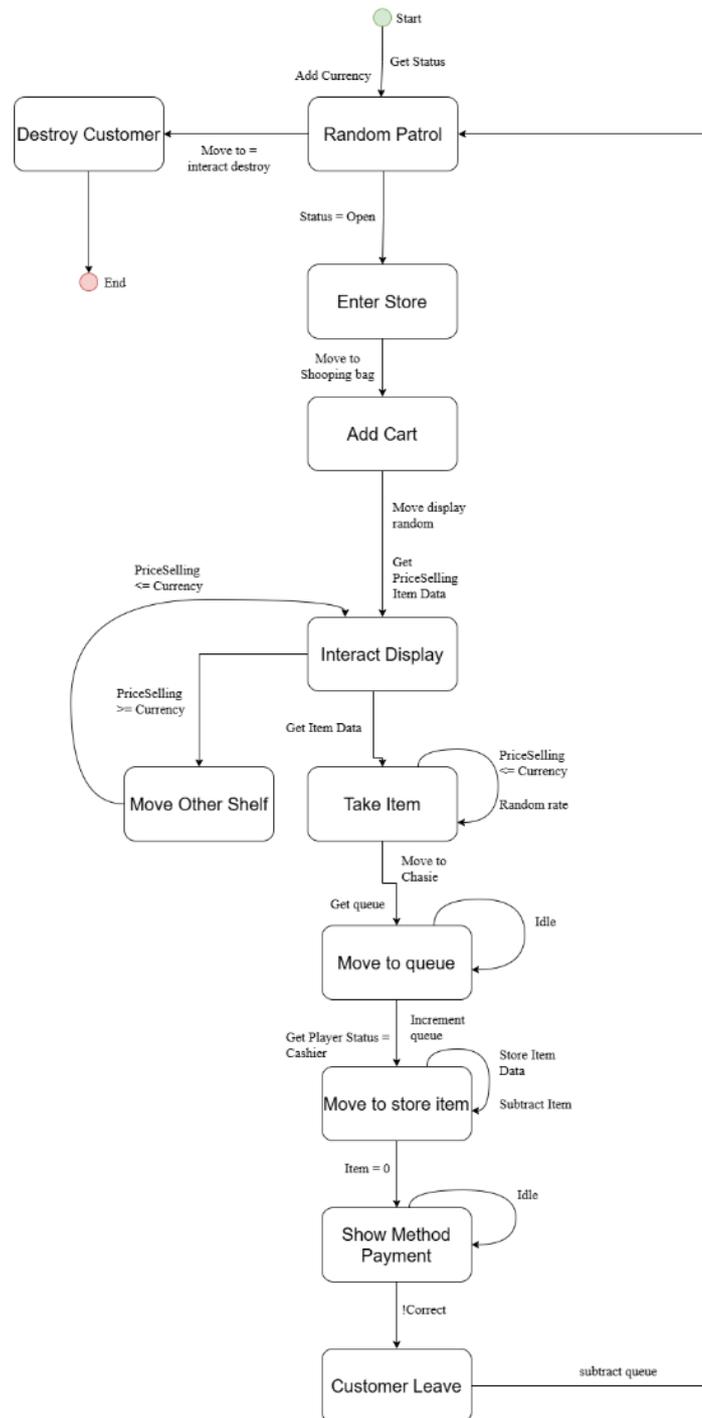
Gambar 3.3 *FSM* Perhitungan SAW

Dari gambar 3.3 perhitungan SAW ini akan terjadi pada tahap *result day* setelah aktivitas operasional supermarket selesai dilakukan pada *FSM* 3.2. Saat pemain memasuki *result day*, sistem akan memulai proses pengambilan data item melalui state *Show Card List*. Pada tahap ini, sistem akan mengambil seluruh data item yang tersedia di dalam game, kemudian membersihkan data hasil perhitungan SAW sebelumnya agar tidak tercampur dengan data baru. Selanjutnya, ketika

pemain menekan tombol *OK*, sistem akan menjalankan proses *Calculate SAW*. Proses ini menggunakan metode *SAW* untuk menghitung nilai preferensi masing-masing item berdasarkan lima belas kriteria (C1 sampai C15) yang telah ditentukan sebelumnya.

Hasil perhitungan dari metode *SAW* kemudian akan ditampilkan dalam *Show Calculate SAW Table* untuk menunjukkan skor masing-masing item. Setiap hasil perhitungan akan menentukan skenario rekomendasi yang sesuai. Misalnya, jika skenario yang dihasilkan adalah C1, maka sistem akan masuk ke node *Recommend C1*. Begitu pula untuk skenario-skenario lainnya, mulai dari C2 hingga C15. Setiap node rekomendasi tersebut akan menyimpan hasil rekomendasi berupa item terbaik berdasarkan skenario perhitungan yang diperoleh. Setelah pemain menekan tombol *OK* di halaman hasil perhitungan, data rekomendasi ini akan disimpan melalui proses *Add SAW Data* ke sistem untuk digunakan pada hari berikutnya.

Proses ini diakhiri dengan state *Next Day*, di mana sistem akan melakukan *Reset Card List* untuk membersihkan rekomendasi sebelumnya, dan *Subtract Expired Item Data* untuk menghapus item-item yang telah melewati masa kadaluarsa. Setelah itu, hari akan berganti, dan sistem siap untuk kembali menjalankan siklus operasional supermarket.

b. *FSM Customer*Gambar 3.4 *FSM NPC Customer*

*FSM* ini menggambarkan alur perilaku customer dalam simulasi, mulai dari awal kedatangan hingga meninggalkan toko setelah melakukan transaksi. Proses dimulai dari *state Start*, di mana sistem melakukan inisialisasi dengan mengambil *status* customer dan menambahkan *currency* sebagai saldo belanja customer. Setelah itu, customer akan bergerak secara acak melalui *state Random Patrol* di area sekitar supermarket. Ketika status toko terbuka (*Status = Open*), customer akan berpindah ke *state Enter Store* untuk mulai berbelanja.

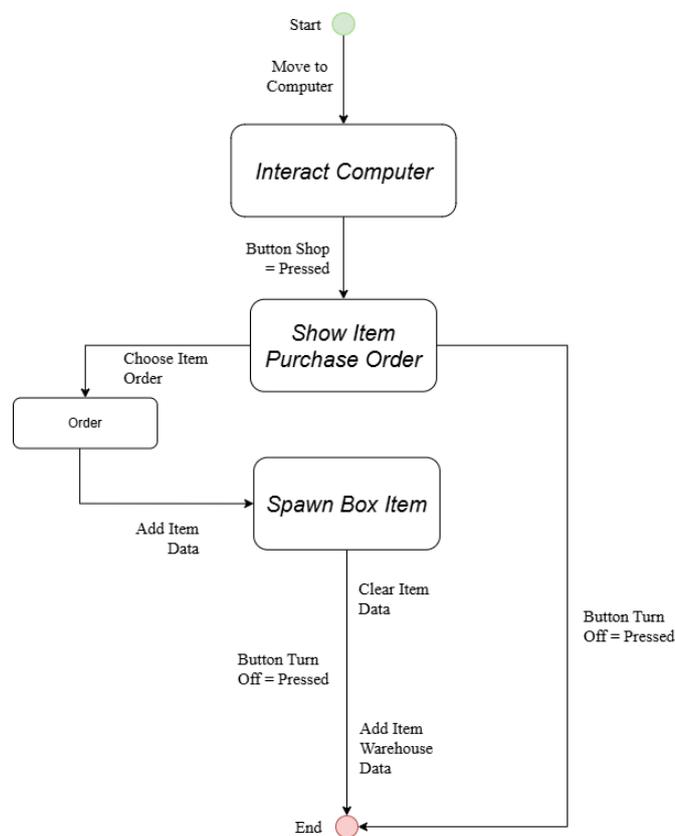
Di dalam toko, *customer* bergerak menuju keranjang belanja melalui *state Add Cart*, kemudian bergerak secara acak menuju rak display item dengan memeriksa harga jual (*PriceSelling*) dan data item yang tersedia di *state Interact Display*. Jika harga item lebih murah atau sama dengan saldo ( $PriceSelling \leq Currency$ ), maka customer dapat mengambil item tersebut (*Take Item*) dan menyimpannya dalam keranjang. Jika harga item melebihi saldo ( $PriceSelling >= Currency$ ), maka customer akan berpindah ke rak lain (*Move Other Shelf*) untuk mencari item lain.

Saat customer berhasil mengambil item, mereka akan menuju ke antrian kasir melalui *state Move to queue*, dengan sistem mencatat jumlah antrian. Jika player memiliki status sebagai kasir (*Get Player Status = Cashier*), maka customer di antrian akan dilayani satu per satu. Proses ini dilakukan dengan memindahkan customer ke *state Move to store item*, di mana sistem akan menyimpan data item yang dibeli dan mengurangi stok item di rak.

Setelah semua item dalam keranjang habis (*Item = 0*), customer akan memasuki *state Show Method Payment* untuk memilih metode pembayaran. Jika

transaksi berhasil, customer akan meninggalkan toko melalui state *Customer Leave* sambil mengurangi jumlah antrian. Jika kondisi tidak sesuai, proses pembayaran akan diulang hingga transaksi berhasil. Setelah meninggalkan toko, customer akan dihapus dari game melalui state *Destroy Customer*, menandai akhir dari siklus perilaku customer tersebut.

c. *FSM Interaksi ke Computer*

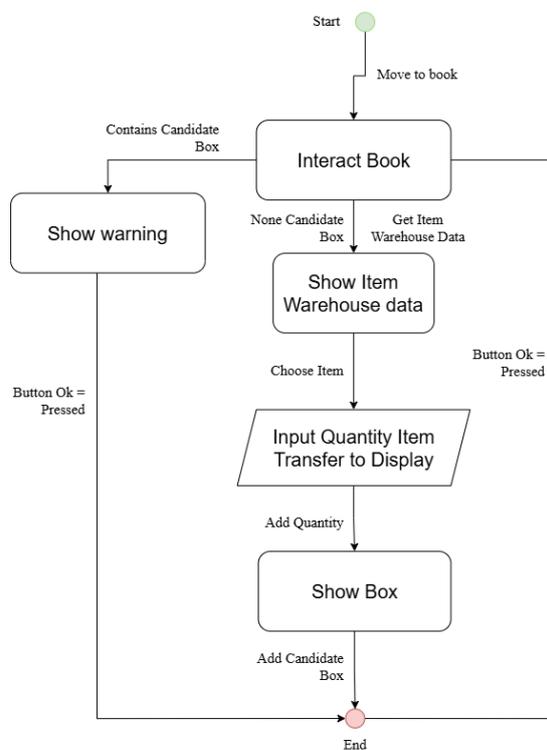


Gambar 3.5 *FSM Interact Computer*

Berdasarkan Gambar 3.5, proses *Interact Computer* bagian dari *FSM* simulasi yang berfungsi untuk melakukan pemesanan stok barang ke dalam gudang. Proses dimulai saat pemain melakukan interaksi dengan komputer. Setelah itu, pemain dapat menekan *Button Shop* untuk menampilkan *tampilan Item Purchase*

*Order*, di mana pemain bisa melihat daftar item yang tersedia untuk dibeli. Pada tahap ini, pemain memiliki dua pilihan, yaitu melakukan pemesanan barang atau menonaktifkan komputer. Jika pemain memilih untuk melakukan *Order* terhadap item tertentu, maka sistem akan memproses pesanan tersebut dan memunculkan *box* yang berisi item hasil pembelian melalui proses *Spawn Box Item*. Sebaliknya, jika pemain tidak melakukan pemesanan atau ingin membatalkan proses, pemain dapat langsung menekan *Button Turn Off* untuk mematikan komputer dan keluar dari menu pembelian. Saat komputer dimatikan, data pesanan yang belum diproses akan dihapus (*Clear Item Data*) dan data item yang sudah dibeli akan ditambahkan ke data gudang (*Add Item Warehouse Data*). Proses ini kemudian berakhir setelah semua tindakan selesai dilakukan.

d. *FSM Interaksi ke Buku*

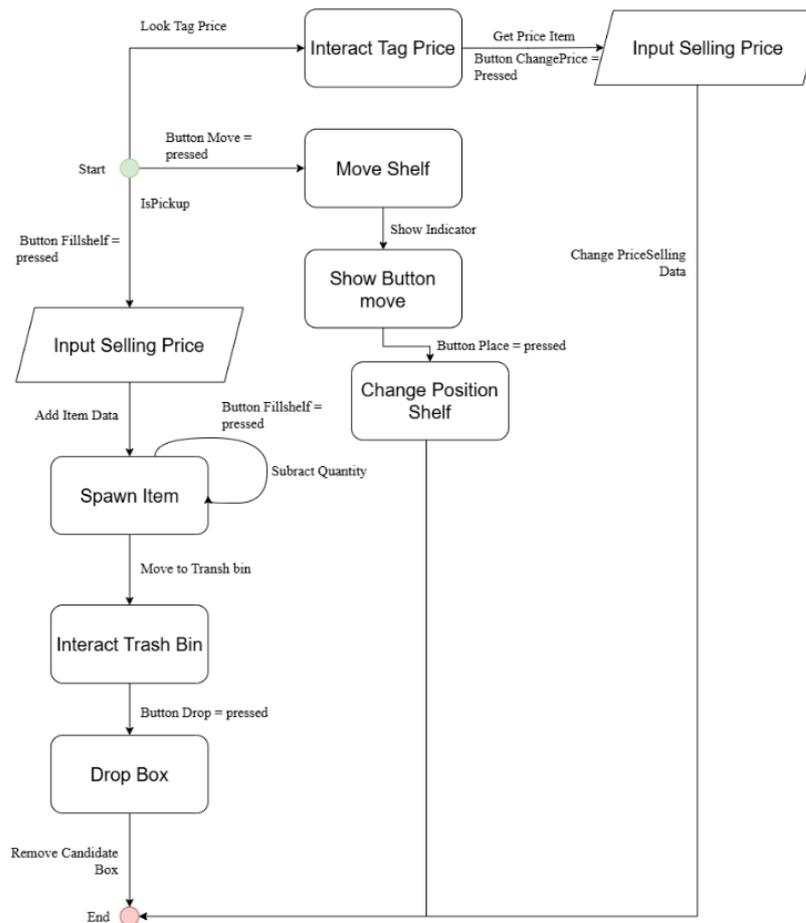


Gambar 3.6 *FSM Interact Book*

Berdasarkan Gambar 3.6, proses *Interact Book* digunakan untuk memindahkan item dari gudang ke rak display. Proses dimulai saat pemain bergerak menuju buku yang berfungsi sebagai media pengelolaan stok. Ketika pemain melakukan interaksi dengan buku, sistem akan melakukan pengecekan apakah terdapat *Candidate Box* (box yang masih tersisa atau belum dikelola). Jika masih ada, maka akan muncul peringatan (*Show Warning*) yang memberitahukan bahwa pemain harus menyelesaikan pengelolaan box sebelumnya terlebih dahulu sebelum memproses item baru.

Namun jika tidak terdapat *Candidate Box*, sistem akan menampilkan *Item Warehouse Data* yang berisi daftar item yang tersedia di gudang. Pemain kemudian dapat memilih item yang ingin dipindahkan ke rak display. Setelah memilih item, pemain akan diarahkan untuk memasukkan jumlah item yang akan ditransfer melalui proses *Input Quantity Item Transfer to Display*. Jumlah item yang diinput akan ditambahkan sebagai data kuantitas, lalu sistem akan memunculkan tampilan box (*Show Box*) yang berisi item sesuai dengan jumlah yang ditentukan. Box tersebut akan tercatat sebagai *Candidate Box* yang nantinya bisa dikelola lebih lanjut, dan proses pun berakhir.

e. *FSM* Interaksi ke Display



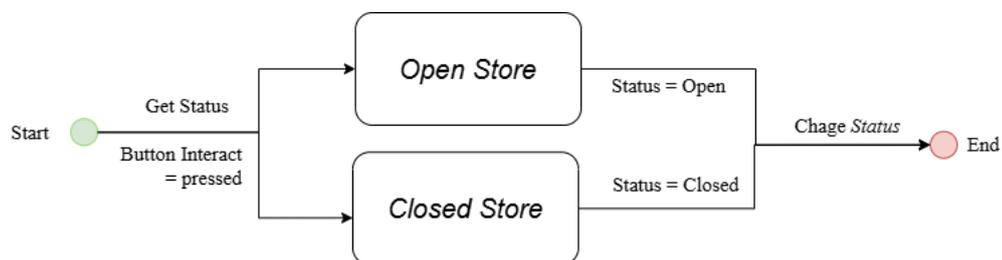
Gambar 3.7 FSM Interact Shelf

Berdasarkan Gambar 3.7, proses ini menggambarkan pengelolaan item di dalam area supermarket, mulai dari menentukan harga jual, memindahkan rak, hingga menempatkan barang ke rak display. Proses dimulai dengan pengecekan kondisi *IsPickup*, yaitu status yang diperoleh dari FSM sebelumnya pada Gambar 3.2, di mana jika pemain sebelumnya sedang memegang box (*candidate box*), maka *IsPickup* akan bernilai true, dan jika tidak, bernilai false. Apabila pemain menekan tombol *Move*, proses akan dilanjutkan ke *Move Shelf* untuk memindahkan rak yang ditandai dengan indikator posisi. Setelah itu, sistem akan menampilkan *Button*

*Move*, dan jika tombol *Place* ditekan, posisi rak akan dipindahkan sesuai keinginan pemain.

Selain memindahkan rak, pemain juga dapat melakukan interaksi dengan *Tag Price* untuk melihat harga barang. Jika tombol *Change Price* ditekan, maka pemain diarahkan ke proses *Input Selling Price* untuk mengubah harga jual item tertentu. Selanjutnya, jika pemain menekan tombol *Fill Shelf*, sistem akan melakukan proses *Spawn Item*, yaitu menambahkan item dari box ke rak display sesuai jumlah yang diinput. Setiap kali tombol *Fill Shelf* ditekan, jumlah kuantitas barang dalam box akan berkurang. Apabila sisa barang dalam box sudah selesai dikelola, pemain dapat membawa box tersebut ke *Trash Bin* untuk dibuang. Interaksi dengan *trash bin* dilakukan melalui proses *Interact Trash Bin*, lalu pemain dapat menekan tombol *Drop* yang akan menjalankan proses *Drop Box* untuk menghapus box dari scene dan menghapus status *Candidate Box* dari sistem. Proses ini memastikan bahwa box yang sudah kosong atau tidak terpakai tidak lagi tercatat dalam data *candidate box* aktif di sistem.

f. *FSM* Interaksi dengan Penanda Supermarket

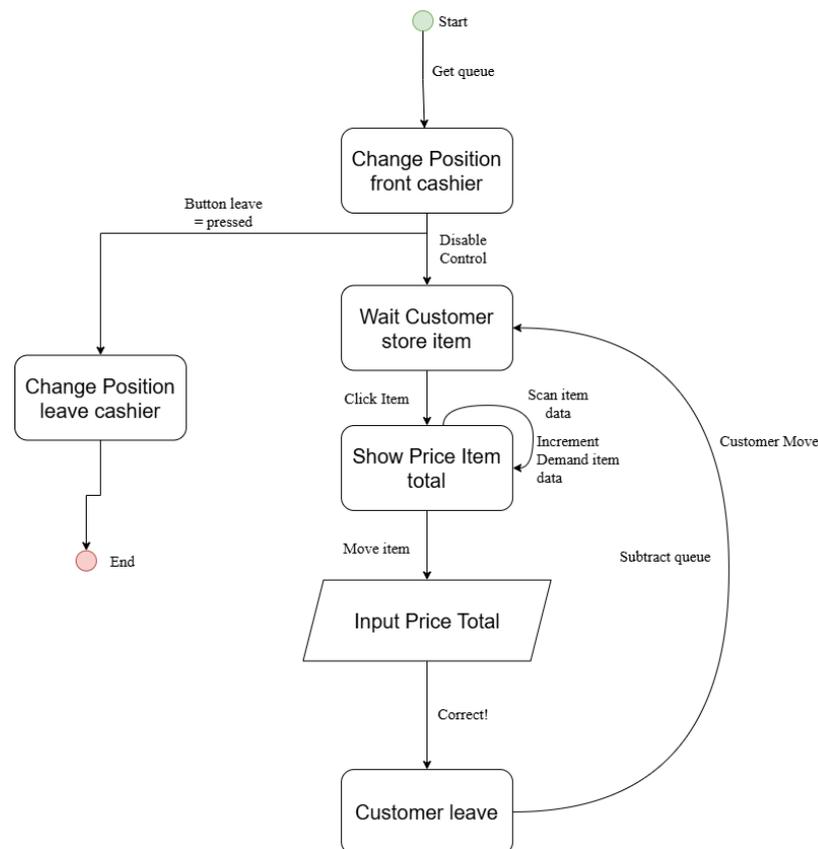


Gambar 3.8 *FSM* *Interact Sign*

Pada gambar 3.8 menggambarkan alur logika untuk mengatur status operasional toko dalam simulasi. Proses dimulai dari titik awal (*Start*), di mana

sistem akan mengecek status toko saat ini melalui aksi "*Get Status*". Berdasarkan hasil pengecekan, sistem akan masuk ke salah satu dari dua keadaan, yaitu "*Open Store*" jika toko sedang buka, atau "*Closed Store*" jika toko sedang tutup. Setelah status saat ini diketahui, sistem kemudian menunggu interaksi pemain berupa penekanan tombol (*Button Interact = pressed*). Ketika tombol ditekan, sistem akan melakukan aksi "*Change Status*", yaitu mengubah status toko dari buka menjadi tutup atau sebaliknya. Proses ini berakhir pada titik "*End*", menandakan bahwa status toko telah berhasil diubah. *FSM* ini digunakan untuk mengontrol secara logis bagaimana pemain membuka dan menutup toko dalam simulasi.

g. *FSM* Interaksi dengan Cashier



Gambar 3.9 *FSM Interact Cashier*

Pada gambar 3.9 menggambarkan proses interaksi antara kasir dan pelanggan serta player di dalam simulasi. Proses dimulai dari titik "Start" dengan sistem mengambil antrian pelanggan yang ada. Selanjutnya, pemain atau karakter kasir akan berpindah ke posisi depan kasir (*Change Position front cashier*), dan kontrol permainan akan dinonaktifkan untuk memfokuskan pada proses transaksi. Setelah itu, sistem masuk ke tahap "*Wait Customer store item*", yaitu menunggu pelanggan meletakkan barang belanjaan mereka di meja kasir. Ketika pemain mengklik barang tersebut, sistem akan menampilkan harga total dari item tersebut melalui tahap "*Show Price Item total*", sambil melakukan pemindaian data item dan meningkatkan data permintaan item (*increment demand item data*). Barang yang dipindahkan kemudian dilanjutkan ke tahap "*Input Price Total*", di mana pemain memasukkan harga total yang ditampilkan. Jika input benar, maka pelanggan akan meninggalkan area kasir (*Customer leave*), dan antrian akan dikurangi. Proses kemudian kembali ke tahap menunggu pelanggan berikutnya. Namun, jika pemain menekan tombol untuk meninggalkan kasir (*Button leave = pressed*), maka sistem akan mengubah posisi karakter ke luar area kasir (*Change Position leave cashier*) dan mengakhiri proses di titik "*End*".

### **3.2.2 Perancangan Simulasi**

Tampilan ini adalah area tempat pemain mengelola toko atau menjalankan simulasi.



Gambar 3.10 Ilustrasi Simulasi

Pada Gambar 3.5 ditampilkan ilustrasi simulasi operasional supermarket yang dirancang untuk merepresentasikan proses pengelolaan supermarket secara interaktif. Simulasi ini memungkinkan pemain berperan sebagai pengelola supermarket yang bertanggung jawab terhadap berbagai aspek operasional, mulai dari pengaturan stok barang, proses restock, pelayanan pelanggan, hingga penghitungan skor performa harian. Dalam perancangannya, setiap barang memiliki data atribut seperti harga, stok, masa kadaluarsa, dan tingkat permintaan. Ketika stok barang mulai menipis, sistem akan secara otomatis melakukan pengecekan ketersediaan dan memunculkan opsi restock. Keputusan *restock* dilakukan berdasarkan perhitungan prioritas menggunakan metode *SAW*, yang dirancang untuk membantu pemain menentukan barang mana yang perlu diprioritaskan berdasarkan kriteria yang telah ditentukan. Selain itu, sistem juga dirancang untuk mensimulasikan aktivitas pelanggan yang berbelanja, memilih barang, dan melakukan pembayaran di kasir. Seluruh alur ini dirancang untuk memberikan gambaran mengenai proses operasional supermarket yang dinamis,

sekaligus menguji efektivitas sistem pengambilan keputusan berbasis SAW dalam situasi simulasi manajemen stok barang.

### 3.3 Perancangan Metode *Simple Additive Weighting*



Gambar 3.11 Skenario perhitungan SAW

Dari Diagram diatas perancangan rekomendasi pada penelitian ini mencakup empat tahap utama, yaitu pengumpulan data, normalisasi data, perhitungan skor, dan prediksi rekomendasi. Tahap pertama yaitu pengumpulan data, data kriteria akan digunakan setelah pemain menambahkan barang dan sistem merekam data operasional supermarket secara dinamis, seperti permintaan barang, harga, stok, dan tanggal kadaluarsa, selama toko beroperasi. Data ini kemudian digunakan sebagai dasar untuk analisis lebih lanjut.

Selanjutnya, data yang telah dikumpulkan akan melalui proses normalisasi menggunakan metode *SAW*. Normalisasi dilakukan untuk menyetarakan skala nilai setiap kriteria sehingga dapat dibandingkan secara objektif. Setelah data dinormalisasi, perhitungan skor dilakukan dengan memberikan bobot pada setiap kriteria sesuai dengan tingkat kepentingannya. Skor ini mencerminkan prioritas dari masing-masing barang berdasarkan parameter yang telah ditentukan.

Kemudian tahap ketiga yaitu perhitungan skor alternatif, di mana data yang telah dinormalisasi dikalikan dengan bobot masing-masing kriteria. Bobot diberikan berdasarkan tingkat kepentingan setiap kriteria, seperti stok yang memiliki bobot lebih tinggi dibanding harga, karena ketersediaan barang lebih penting untuk memenuhi kebutuhan pelanggan. Hasil dari tahap ini berupa skor alternatif untuk setiap barang, yang mencerminkan prioritasnya dalam pengambilan keputusan.

Dan terakhir yaitu prediksi rekomendasi, di mana barang-barang dengan skor tertinggi akan diidentifikasi untuk tindakan tertentu, seperti restock. Hasil rekomendasi ini akan ditampilkan sebelum toko dibuka keesokan harinya, sehingga pemain dapat mengambil langkah strategis berdasarkan analisis data yang telah dilakukan. Pendekatan ini dirancang untuk memastikan bahwa rekomendasi bersifat adaptif terhadap perubahan data harian, memungkinkan pemain untuk mengelola inventori secara optimal dalam skenario simulasi.

### **3.3.1 Pengumpulan Data Alternatif dan Kriteria**

Data barang salah satu elemen inti dalam simulasi Supermarket, yang dirancang untuk memberikan pengalaman manajemen inventori secara realistis.

Tabel data barang mencakup berbagai atribut penting yang menggambarkan setiap produk yang tersedia di supermarket, seperti ID barang, nama barang, harga, stok, masa kadaluarsa, dan permintaan. Setiap atribut ini berperan penting dalam mekanisme pengambilan keputusan dan simulasi operasional. Berikut tabel data kriteria dan alternatif barang yang digunakan dalam simulasi operasional supermarket:

Tabel 3.1 Tabel Kriteria

<b>Kriteria</b>	<b>Keterangan</b>
$K_1$	Harga
$K_2$	Stok
$K_3$	Masa Kadaluarsa
$K_4$	Data Permintaan

Dengan data perhitungan manual bobot masing-masing dari kriteria tabel diatas sebagai berikut:

$$\text{Bobot Total} = \kappa_1 + \kappa_2 + \kappa_3 + \kappa_4 = 1 \quad (2.1)$$

Tabel 3.2 Data Bobot

<b>Harga</b>	<b>Stok</b>	<b>Kadaluarsa</b>	<b>Permintaan</b>
0,1	0,4	0,2	0,3

Dari tabel di atas, kriteria harga diberikan bobot sebesar 0.1, karena aspek biaya tetap menjadi salah satu faktor penting dalam efisiensi operasional supermarket. Produk dengan harga yang lebih rendah dapat membantu mengurangi biaya pengadaan. Stok juga diberikan bobot 0.4, sebagai bobot tertinggi, karena ketersediaan barang sangat berpengaruh terhadap kelancaran operasional dan kepuasan pelanggan.

Masa kadaluarsa memiliki bobot 0.2, karena meskipun penting untuk menjaga kualitas barang, aspek ini masih dapat dikelola melalui rotasi stok dan

jadwal pembelian yang teratur. Terakhir, permintaan diberikan bobot 0.3, karena produk dengan tingkat permintaan tinggi berpotensi meningkatkan penjualan dan profitabilitas. Melalui pembobotan ini, nilai normalisasi pada masing-masing kriteria akan dikalikan dengan bobotnya untuk menghasilkan skor akhir setiap produk.

Tabel 3.3 Data Barang

<b>ID Alternatif</b>	<b>Nama Barang</b>	<b>Harga Awal</b>	<b>Masa Kadaluarsa</b>	<b>Permintaan Awal</b>
$C_1$	Cola	Rp18.000	9 hari	10
$C_2$	Fanta Orange	Rp17.500	6 hari	10
$C_3$	Lay's Potato Chips	Rp15.000	4 hari	10
$C_4$	Nutella Chocolate Spread	Rp65.000	18 hari	10
$C_5$	Candy Pack	Rp12.000	12 hari	10
$C_6$	Chocapic Cereal	Rp38.000	9 hari	10
$C_7$	Whole Wheat Bread	Rp22.000	7 hari	10
$C_8$	Barilla Pasta	Rp28.000	24 hari	10
$C_9$	Fresh Milk	Rp25.000	7 hari	10
$C_{10}$	Cooking Oil	Rp38.000	12 hari	10
$C_{11}$	Sunlight Dish Soap	Rp15.000	24 hari	10
$C_{12}$	Dishwater Detergent	Rp45.000	9 hari	10
$C_{13}$	Pepsi Soda	Rp18.000	6 hari	10
$C_{14}$	Camembert Cheese	Rp75.000	6 hari	10
$C_{15}$	Whipping Cream	Rp40.000	6 hari	10

Dari tabel diatas, id barang digunakan sebagai identitas unik untuk setiap produk, mempermudah pencatatan dan pengelolaan inventori dalam game. Nama barang memberikan deskripsi produk yang dapat dikenali oleh pemain, seperti "Cola" atau " Fresh Milk", data barang akan bersifat dinamis berdasarkan interaksi pemain, semakin banyak barang maka akan semakin banyak juga skor alternatif yang dihasilkan.

Selanjutnya harga barang bersifat dinamis, di mana pemain dapat mengubah harga sesuai strategi penjualan, tetapi terdapat harga default atau rekomendasi yang

disesuaikan dengan perkembangan aktivitas jual beli di supermarket. Pemain perlu mempertimbangkan perubahan harga ini untuk menjaga efisiensi operasional dan daya tarik barang di mata pelanggan.

Stok barang juga bersifat dinamis, bergantung pada jumlah barang yang distock oleh pemain melalui mekanisme dalam game. Pemain harus memastikan stok tetap mencukupi untuk memenuhi permintaan pelanggan sekaligus menghindari kelebihan stok yang dapat mengurangi efisiensi. Tetapi dari data akan diberikan sama rata 12 unit untuk setiap barang untuk menghitung nilai skor alternatif yang didapat. Masa kadaluarsa barang, yang dihitung dalam hari, memberikan informasi waktu sebelum produk tidak layak jual. Pemain diharapkan memprioritaskan pengelolaan barang dengan masa kadaluarsa pendek agar tidak mengalami kerugian akibat produk yang kadaluarsa.

Sistem permintaan barang dirancang untuk bersifat dinamis, dimulai dengan permintaan default yang telah ditetapkan untuk setiap barang. Ketika *NPC customer* membeli suatu barang, permintaan untuk barang tersebut akan meningkat secara otomatis dengan menambahkan 1 ke nilai permintaan yang ada. Sistem ini mencerminkan popularitas barang di kalangan pelanggan dan memberikan informasi yang berguna bagi pemain untuk mengelola inventori secara strategis.

### **3.3.2 Perhitungan Normalisasi Data**

Metode *SAW* diterapkan dalam simulasi game Supermarket Simulator untuk membantu pengambilan keputusan dalam pengelolaan inventori barang. Metode ini menggunakan empat kriteria utama: harga, stok, masa kadaluarsa, dan permintaan. bobot setiap kriteria ditentukan untuk mencerminkan tingkat kepentingannya dalam

pengambilan keputusan. Bobot ini biasanya ditentukan berdasarkan analisis kebutuhan, studi literatur, atau kebijakan bisnis yang relevan. Kriteria yang digunakan adalah harga, stok, masa kadaluarsa, dan permintaan.

Proses penghitungan dimulai dengan normalisasi data. Untuk kriteria benefit (permintaan), nilai normalisasi dihitung dengan rumus:

$$R_{C_k} = \frac{\chi_{C_k}}{\text{max value in column}} \quad (2.2)$$

Sebaliknya, untuk kriteria cost (harga, stok, dan masa kadaluarsa), rumusnya:

$$R_{C_k} = \frac{\text{min value in column}}{\chi_{C_k}} \quad (2.3)$$

Normalisasi data pada bagian stok dilakukan untuk mengubah nilai stok dinamis menjadi skala yang seragam agar dapat digunakan dalam perhitungan. Karena stok barang bersifat dinamis, nilai stok dapat berubah setiap harinya berdasarkan tindakan pemain dalam game, seperti menambah stok melalui restock atau mengurangi stok akibat pembelian oleh *NPC customers*. Maka dari itu data stok akan dijadikan permisalan yang akan digunakan perhitungan. Berikut hasil normalisasi:

Tabel 3.4 Data Normalisasi

<b>ID Alternatif</b>	<b>Normalisasi Harga</b>	<b>Normalisasi Stok Awal</b>	<b>Normalisasi Kadaluarsa</b>	<b>Normalisasi Permintaan</b>
$C_1$	0,666	1	0,37	1
$C_2$	0,685	1	0,25	1
$C_3$	0,800	1	0,16	1
$C_4$	0,184	1	0,75	1
$C_5$	1,000	1	0,50	1
$C_6$	0,315	1	0,37	1
$C_7$	0,545	1	0,29	1
$C_8$	0,428	1	1,00	1
$C_9$	0,480	1	0,29	1
$C_{10}$	0,315	1	0,50	1
$C_{11}$	0,800	1	1,00	1

ID Alternatif	Normalisasi Harga	Normalisasi Stok Awal	Normalisasi Kadaluarsa	Normalisasi Permintaan
$C_{12}$	0,266	1	0,37	1
$C_{13}$	0,666	1	0,25	1
$C_{14}$	0,160	1	0,25	1
$C_{15}$	0,300	1	0,25	1

### 3.3.3 Perhitungan Skor Alternatif

Setelah data dinormalisasi, langkah berikutnya dalam metode SAW adalah menghitung skor akhir untuk setiap barang berdasarkan bobot kriteria yang telah ditentukan. Rumus yang digunakan untuk menghitung skor adalah :

$$v_c = \sum W_k \cdot R_{ck} \quad (2.4)$$

Di mana  $V_c$  adalah skor akhir barang ke- $c$ ,  $W_k$  adalah bobot untuk kriteria ke- $k$ , dan  $R_{ck}$  adalah nilai normalisasi untuk barang ke- $c$  pada kriteria ke- $k$ . Berikut ini adalah perhitungan skor yang didapatkan:

- a. ID Barang  $C_1$  (Cola)

$$V_1 = (0,3 \times 0,666) + (0,2 \times 1) + (0,25 \times 0,375) + (0,25 \times 1) = 0,743$$

- b. ID Barang  $C_2$  (Fanta Orange)

$$V_2 = (0,3 \times 0,685) + (0,2 \times 1) + (0,25 \times 0,250) + (0,25 \times 1) = 0,442$$

- c. ID Barang  $C_3$  (Lay's Potato Chips)

$$V_3 = (0,3 \times 0,800) + (0,2 \times 1) + (0,25 \times 0,166) + (0,25 \times 1) = 0,490$$

- d. ID Barang  $C_4$  (Nutella Chocolate Spread)

$$V_4 = (0,3 \times 0,184) + (0,2 \times 1) + (0,25 \times 0,750) + (0,25 \times 1) = 0,352$$

e. ID Barang  $C_5$  (Candy Pack)

$$V_5 = (0,3 \times 1) + (0,2 \times 1) + (0,25 \times 0,500) + (0,25 \times 1) = 0,558$$

f. ID Barang  $C_6$  (Chocapic Cereal)

$$V_6 = (0,3 \times 0,315) + (0,2 \times 1) + (0,25 \times 0,375) + (0,25 \times 1) = 0,639$$

g. ID Barang  $C_7$  (Whole Wheat Bread)

$$V_7 = (0,3 \times 0,545) + (0,2 \times 1) + (0,25 \times 0,291) + (0,25 \times 1) = 0,474$$

h. ID Barang  $C_8$  (Barilla Pasta)

$$V_8 = (0,3 \times 428) + (0,2 \times 1) + (0,25 \times 1) + (0,25 \times 1) = 0,634$$

i. ID Barang  $C_9$  (Fresh Milk)

$$V_9 = (0,3 \times 0,480) + (0,2 \times 1) + (0,25 \times 0,291) + (0,25 \times 1) = 0,435$$

j. ID Barang  $C_{10}$  (Cooking Oil)

$$V_{10} = (0,3 \times 0,315) + (0,2 \times 1) + (0,25 \times 0,5000) + (0,25 \times 1) = 0,529$$

k. ID Barang  $C_{11}$  (Sunlight Dish Soap)

$$V_{11} = (0,3 \times 0,800) + (0,2 \times 1) + (0,25 \times 1) + (0,25 \times 1) = 0,300$$

l. ID Barang  $C_{12}$  (Dishwater Detergent)

$$V_{12} = (0,3 \times 0,266) + (0,2 \times 1) + (0,25 \times 0,375) + (0,25 \times 1) = 0,249$$

m. ID Barang  $C_{13}$  (Pepsi Soda)

$$V_{13} = (0,3 \times 0,666) + (0,2 \times 1) + (0,25 \times 0,25) + (0,25 \times 1) = 0,655$$

n. ID Barang  $C_{14}$  (Camembert Cheese)

$$V_{14} = (0,3 \times 0,160) + (0,2 \times 1) + (0,25 \times 0,25) + (0,25 \times 1) = 0,504$$

o. ID Barang  $C_{15}$  (Whipping Cream)

$$V_{15} = (0,3 \times 0,300) + (0,2 \times 1) + (0,25 \times 0,25) + (0,25 \times 1) = 0,447$$

Tabel 3.5 Rating Alternatif

ID Alternatif	Nama Barang	Skor Alternatif	Peringkat
$C_1$	Cola	0,7437	4
$C_2$	Fanta Orange	0,7182	6
$C_3$	Lay's Potato Chips	0,7316	5
$C_4$	Nutella Chocolate Spread	0,6928	8
$C_5$	Candy Pack	0,875	2
$C_6$	Chocapic Cereal	0,6384	12
$C_7$	Whole Wheat Bread	0,6865	9
$C_8$	Barilla Pasta	0,8285	3
$C_9$	Fresh Milk	0,6669	11
$C_{10}$	Cooking Oil	0,6697	10
$C_{11}$	Sunlight Dish Soap	0,94	1
$C_{12}$	Dishwater Detergent	0,6237	13
$C_{13}$	Pepsi Soda	0,7125	7
$C_{14}$	Camembert Cheese	0,5605	15
$C_{15}$	Whipping Cream	0,6025	14

Setelah menghitung semua skor, dari data default, skor sementara memberikan gambaran prioritas pengelolaan barang dalam supermarket. Barang dengan skor tertinggi menunjukkan kombinasi optimal dari kriteria seperti harga yang kompetitif, stok yang memadai, masa kadaluarsa yang panjang, dan permintaan yang tinggi. Barang-barang ini sebaiknya dipertahankan dalam jumlah stok yang cukup dan dapat menjadi fokus promosi untuk meningkatkan penjualan.

### 3.3.4 Prediksi Rekomendasi menggunakan SAW

Pada tahap prediksi rekomendasi, sistem menggunakan skor alternatif yang telah dihitung pada Tabel 3.5 untuk menentukan barang-barang prioritas yang memerlukan tindakan, seperti restock, atau pengoptimalan harga. Berdasarkan data saat ini, barang dengan skor tertinggi, seperti Sunlight Dish Soap (0,94), Candy Pack (0,875), dan Barilla Pasta (0,8285), mendapatkan prioritas utama dalam rekomendasi karena memiliki kriteria yang memenuhi kebutuhan pelanggan secara optimal. Maka, barang dengan skor tertinggi akan direkomendasikan untuk di-restock guna memastikan ketersediaannya tetap terjaga.

Di sisi lain, barang dengan skor rendah, seperti Camembert Cheese (0,5605) dan Whipping Cream (0,6025), memiliki prioritas lebih rendah karena kurang diminati atau stoknya masih mencukupi. Oleh karena itu, barang dengan skor paling rendah akan direkomendasikan untuk diberi kandidat yang perlu pengawasan guna meningkatkan penjualannya.

Namun, karena data bersifat dinamis, hasil rekomendasi akan berubah seiring interaksi pemain dengan sistem. Misalnya, stok barang akan diperbarui setiap kali pemain menambah atau mengurangi stok, sementara data permintaan meningkat setiap kali *NPC* membeli barang tertentu. Selain itu, harga barang juga bersifat dinamis, memungkinkan pemain untuk menyesuaikan harga berdasarkan perkembangan penjualan dan rekomendasi sistem. Semua perubahan ini mempengaruhi perhitungan skor alternatif, sehingga rekomendasi yang dihasilkan setiap hari akan selalu relevan dengan kondisi terbaru supermarket.

Perlu diingat bahwa nilai peringkat ini diperoleh berdasarkan perhitungan normalisasi dan nilai bobot yang diterapkan pada setiap kriteria. Nilai normalisasi dihitung dengan mempertimbangkan tingkat permintaan sebagai salah satu faktor penting, di mana tingkat permintaan ini selalu berubah secara dinamis mengikuti pola *NPC customers*. Perubahan ini merefleksikan ketidakpastian kondisi nyata di pasar, seperti fluktuasi minat pelanggan terhadap suatu produk. Oleh karena itu, hasil perhitungan dapat bervariasi setiap kali proses simulasi dilakukan, dan sistem harus tetap fleksibel dalam menyesuaikan strategi pengelolaan barang berdasarkan perubahan tingkat permintaan tersebut.

### 3.4 Rencana Pengujian

Rencana pengujian dalam penelitian ini akan menggunakan metode *System Usability Scale (SUS)* untuk mengevaluasi efektivitas dan kegunaan sistem rekomendasi yang diterapkan dalam simulasi operasional supermarket. Pengujian ini bertujuan untuk menilai sejauh mana simulasi dapat membantu pemain dalam mengelola barang secara tepat, khususnya dalam penerapan metode *Simple Additive Weighting (SAW)* pada fitur restock. Selain itu, pengujian ini juga bertujuan untuk mengevaluasi sejauh mana simulasi mencerminkan kondisi operasional nyata di sebuah supermarket, di mana pemain berperan sebagai manajer yang mengambil keputusan strategis terkait pengelolaan stok, harga, dan rekomendasi barang. Berikut adalah daftar pertanyaan kuesioner yang akan digunakan untuk pengujian menggunakan *SUS*:

Tabel 3.6 Daftar Pertanyaan Kuesioner SUS

No	Pertanyaan
$Q_1$	Saya merasa sistem ini mudah dipahami saat pertama kali digunakan.

No	Pertanyaan
Q <sub>2</sub>	Saya merasa sistem ini terlalu rumit untuk dikelola.
Q <sub>3</sub>	Saya merasa antarmuka sistem memudahkan saya dalam melakukan restock dan pengelolaan barang.
Q <sub>4</sub>	Saya merasa perlu bantuan teknis untuk bisa menggunakan sistem ini.
Q <sub>5</sub>	Saya merasa fitur rekomendasi yang diberikan oleh sistem membantu operasional supermarket.
Q <sub>6</sub>	Saya merasa sistem memberikan informasi yang membingungkan saat melakukan pengaturan stok dan harga.
Q <sub>7</sub>	Saya merasa percaya diri dalam mengambil keputusan berdasarkan rekomendasi dari sistem.
Q <sub>8</sub>	Saya merasa banyak fitur sistem yang tidak konsisten atau tidak bekerja sebagaimana mestinya.
Q <sub>9</sub>	Saya merasa sistem ini memberikan pengalaman yang memadai sebagai manajer supermarket.
Q <sub>10</sub>	Saya merasa perlu mempelajari banyak hal sebelum bisa menggunakan sistem ini secara efektif.

Dari tabel 3.6, menunjukkan daftar 10 pertanyaan yang dirancang untuk menggali aspek-aspek penting yang memengaruhi pengalaman dan kepuasan pengguna. Setiap pertanyaan diukur menggunakan skala Likert 5 poin, mulai dari 1 (sangat tidak setuju) hingga 5 (sangat setuju). Dalam pengolahan hasil, setiap nilai jawaban akan dikonversi sesuai aturan standar SUS, yaitu: untuk pertanyaan dengan nomor ganjil yang bersifat positif, skor dikurangi 1, sedangkan untuk pertanyaan dengan nomor genap yang bersifat negatif, skor dikurangi dari 5. Nilai hasil konversi kemudian dijumlahkan dan dikalikan 2,5 untuk mendapatkan skor total SUS dengan rentang 0–100.

Setelah seluruh skor dari responden dihitung, rata-rata skor *SUS* akan dianalisis untuk mengevaluasi tingkat kegunaan sistem. Interpretasi skor dilakukan dengan kategori, di bawah 50 menunjukkan tingkat kegunaan rendah, 50–70 menunjukkan tingkat kegunaan rata-rata, dan di atas 70 menunjukkan sistem memiliki tingkat kegunaan yang baik dan mudah digunakan. Data ini akan memberikan gambaran apakah sistem simulasi yang dikembangkan telah

memenuhi ekspektasi pengguna, khususnya terkait kemudahan penggunaan antarmuka, kejelasan informasi, keakuratan rekomendasi, dan kesesuaian simulasi dengan kondisi nyata supermarket (Lewis, 2020).

Selain itu, analisis hasil *SUS* juga akan dikombinasikan dengan umpan balik deskriptif dari responden untuk mengidentifikasi area yang perlu diperbaiki, baik pada aspek antarmuka, fitur rekomendasi, maupun elemen simulasi lainnya. Dengan demikian, hasil pengujian kuantitatif melalui *SUS* dan umpan balik kualitatif dari responden akan menjadi dasar dalam pengembangan lanjutan agar simulasi operasional supermarket yang dibuat lebih intuitif, efektif, dan relevan bagi pemain yang berperan sebagai manajer.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pada bab ini akan disajikan hasil implementasi sistem simulasi operasional supermarket yang telah dikembangkan dengan menerapkan metode *Simple Additive Weighting (SAW)* sebagai dasar pengambilan keputusan dalam pengelolaan stok barang. Proses pengujian dilakukan untuk mengetahui sejauh mana sistem mampu berjalan sesuai dengan kebutuhan yang telah dirancang pada tahap sebelumnya serta untuk melihat efektivitas metode *SAW* dalam membantu menentukan prioritas barang yang perlu diadakan kembali.

Selain itu, pada bab ini juga akan dijelaskan mengenai proses simulasi yang dilakukan, data-data yang dihasilkan dari proses tersebut, serta analisis terhadap hasil simulasi. Hasil yang diperoleh kemudian dievaluasi berdasarkan kriteria-kriteria yang telah ditentukan, seperti harga, jumlah stok, masa kadaluarsa, dan tingkat permintaan barang. Selanjutnya, dilakukan pengujian *usability* terhadap sistem menggunakan metode *System Usability Scale (SUS)* guna memperoleh tanggapan pengguna mengenai kemudahan penggunaan, kelengkapan fitur, serta kepuasan terhadap sistem yang telah dibuat.

#### **4.1 Implementasi Simulasi**

Pada game simulasi operasional supermarket ini, pemain berperan sebagai manajer sebuah supermarket yang bertugas untuk mengatur stok barang, menetapkan harga jual, serta melakukan restock berdasarkan kebutuhan dan permintaan pelanggan.



Gambar 4.1 Tampilan Menu

Pada gambar 4.1 simulasi ini diawali dengan tampilan menu utama yang berisi beberapa pilihan, yaitu tombol *Start* untuk memulai permainan baru dan tombol *Exit* untuk keluar dari simulasi. Pada tahap awal, pemain akan melihat tampilan ini sebagai antarmuka pertama yang tersedia. Apabila pemain menekan tombol *Start*, maka sistem akan langsung menampilkan halaman tutorial yang berisi petunjuk dasar mengenai cara bermain serta mekanisme simulasi yang harus diikuti pemain. Tutorial ini bertujuan untuk memberikan panduan awal sebelum pemain memasuki permainan utama. Sebaliknya, jika pemain memilih tombol *Exit*, maka simulasi akan langsung ditutup dan pemain keluar dari aplikasi.



Gambar 4.2 Tampilan Tutorial

Pada gambar 4.2 adalah tampilan saat tutorial berlangsung, kontrol pemain akan dinonaktifkan sementara agar pemain dapat fokus memahami informasi yang disampaikan dalam tutorial tanpa gangguan. Tutorial ini berisi penjelasan mengenai mekanisme dasar simulasi, seperti cara berinteraksi dengan barang dan elemen permainan lainnya. Setelah pemain menyelesaikan tutorial, sistem akan secara otomatis menampilkan *display* berisi barang-barang yang telah disediakan. Terdapat sebanyak 12 jenis barang dengan stok yang berbeda-beda. Setiap barang yang ditampilkan pada *display* tersebut telah memiliki data *default* yang mencakup berbagai kriteria, seperti harga, stok, dan masa berlaku. Data ini berfungsi sebagai nilai awal yang nantinya dapat diubah oleh pemain melalui berbagai interaksi selama simulasi berlangsung. Salah satu contohnya, pemain dapat mengatur ulang harga barang yang ditampilkan di *display* sesuai kebutuhannya untuk strategi penjualan. Perubahan data ini akan langsung memengaruhi kondisi barang dalam simulasi.



Gambar 4.3 Tampilan Area Gudang

Pada gambar 4.3 setelah pemain menyelesaikan tutorial, sistem akan mengarahkan pemain menuju area gudang. Di area ini, pemain memiliki kebebasan untuk mengelola persediaan barang yang tersedia. Pemain dapat melakukan *transfer* barang dari gudang ke area display dengan cara berinteraksi langsung dengan sebuah buku yang terletak di area tersebut. Melalui interaksi ini, pemain bisa memilih barang-barang yang akan dipindahkan ke display untuk dijual kepada pelanggan. Selain itu, pemain juga dapat berinteraksi dengan komputer yang tersedia di area gudang untuk melakukan *Purchase Order* atau pemesanan barang baru. Fitur *Purchase Order* ini memungkinkan pemain menambah stok barang yang diperlukan, sekaligus mengatur strategi pengadaan barang sesuai kebutuhan dan permintaan pelanggan dalam simulasi.



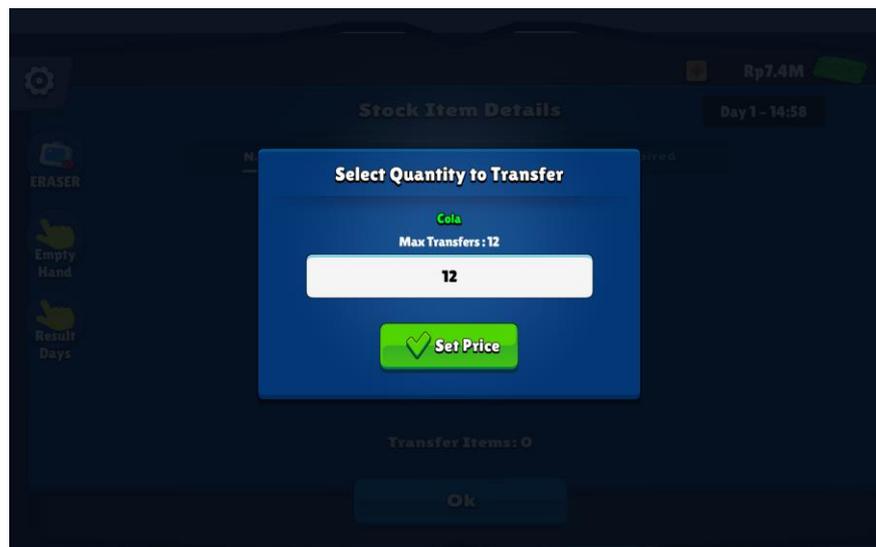
Gambar 4.4 Tampilan Interaksi Buku

Pada gambar 4.4 adalah saat pemain berinteraksi dengan buku yang tersedia di area gudang, sistem akan menampilkan daftar stok barang yang saat itu tersedia di dalam gudang. Pada tampilan ini, pemain dapat melihat informasi lengkap mengenai setiap barang, termasuk jumlah stok yang tersisa dan status masa kadaluarsa dari masing-masing barang. Informasi ini bertujuan membantu pemain dalam mengambil keputusan terkait barang mana yang perlu segera dipindahkan ke display atau disimpan lebih lama di gudang. Melalui interaksi ini, pemain dapat memilih barang tertentu yang ingin ditransfer ke area display dengan cara menandai atau memilih barang tersebut pada daftar, lalu mengonfirmasi proses pemindahan. Barang yang telah dipilih akan otomatis berpindah dari gudang ke display dan siap dijual kepada pelanggan.



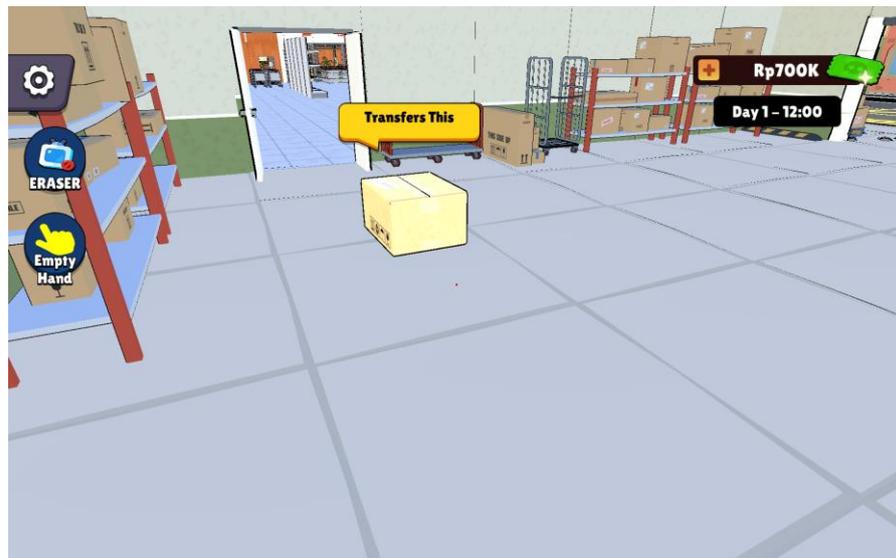
Gambar 4.5 Tampilan Data Barang di gudang

Pada Gambar 4.5 ditunjukkan tampilan ketika pemain melakukan interaksi dengan buku yang tersedia di area gudang. Melalui interaksi ini, sistem akan menampilkan daftar stok barang yang tersimpan di gudang beserta informasi penting seperti jumlah stok yang tersedia dan status masa kadaluarsa masing-masing barang. Selain itu, melalui tampilan ini pemain juga dapat memilih barang-barang tertentu yang ingin dipindahkan ke area display. Pemilihan barang dilakukan dengan memilih nama barang dari daftar stok yang tersedia, sehingga pemain dapat menentukan barang mana saja yang perlu dipindahkan sesuai kebutuhan tanpa harus keluar dari tampilan tersebut.



Gambar 4.6 Tampilan Input quantity barang untuk ditransfer

Pada gambar 4.6 adalah ketika pemain memilih barang untuk ditransfer ke display melalui interaksi dengan buku di area gudang, sistem akan menampilkan jendela konfirmasi berupa tampilan input jumlah barang yang akan ditransfer. Pada tampilan ini, pemain diminta untuk menentukan jumlah atau *quantity* barang yang ingin dipindahkan ke display. Sebagai acuan, nilai *maksimum (quantity max)* yang dapat dimasukkan secara otomatis akan disesuaikan dengan jumlah stok total barang yang tersedia di gudang saat itu. Dengan begitu, pemain tidak dapat memasukkan angka melebihi stok yang ada, sehingga proses transfer tetap sesuai dengan kondisi persediaan yang tercatat. Setelah pemain menentukan jumlah barang yang akan ditransfer dan mengonfirmasi pilihan tersebut, sistem akan langsung memperbarui jumlah stok barang di gudang dan menambahkan barang ke display sesuai jumlah yang ditentukan.



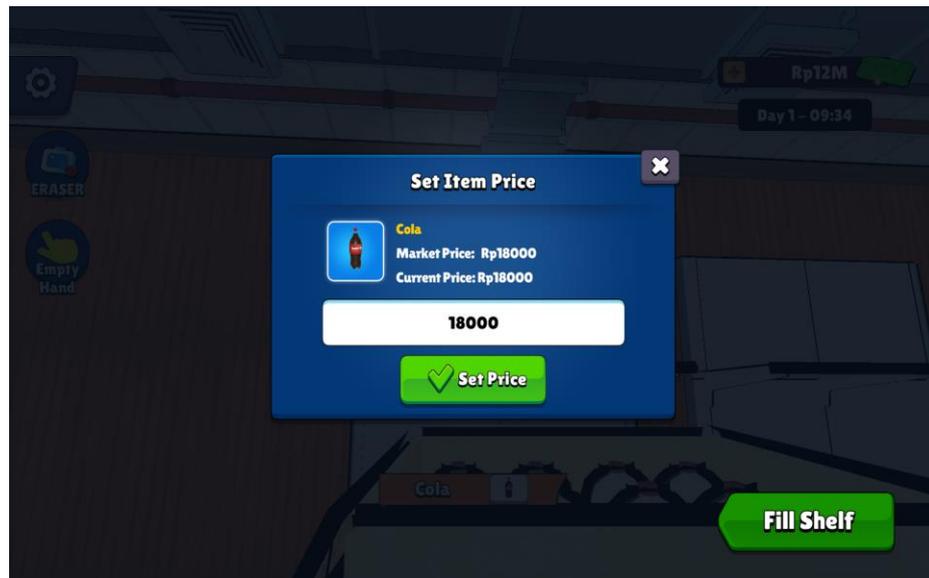
Gambar 4.7 Tampilan Box yang dipilih

Pada gambar 4.7 setelah pemain memasukkan jumlah barang yang akan ditransfer ke display dan mengonfirmasi prosesnya, sistem akan secara otomatis menampilkan sebuah box yang muncul di area gudang. Box ini berisi barang-barang yang telah dipilih oleh pemain sesuai dengan jumlah yang telah ditentukan sebelumnya. Box yang muncul dapat berisi satu jenis barang atau beberapa box jika pemain melakukan beberapa kali transfer. Box tersebut bersifat interaktif, sehingga pemain dapat mengambil atau melakukan pickup terhadap box tersebut. Setelah di-pickup, pemain dapat membawa box tersebut ke area display dan meletakkannya di rak yang tersedia. Proses ini menjadi bagian dari mekanisme pengisian barang ke display, di mana barang yang diambil dari gudang harus secara fisik dipindahkan oleh pemain ke rak display sebelum dapat dijual kepada pelanggan dalam simulasi.



Gambar 4.8 Tampilan restock barang di display

Pada gambar 4.8 ketika pemain mengisi barang ke display melalui proses transfer, data barang yang dipilih akan langsung dicatat di area display. Pada saat yang sama, jumlah stok barang di gudang akan otomatis berkurang sesuai dengan jumlah barang yang ditransfer. Selain jumlah stok, data lain yang terkait dengan barang tersebut, seperti harga, data permintaan, dan masa kadaluarsa, juga akan ikut terbawa dan dicatat di display. Informasi ini penting karena nantinya akan digunakan dalam proses penjualan maupun perhitungan sistem manajemen barang di dalam simulasi. Dengan demikian, setiap barang yang sudah berada di display memiliki data yang terstruktur dan terhubung dengan sistem, sementara stok di gudang terus diperbarui secara dinamis berdasarkan aktivitas transfer yang dilakukan pemain.



Gambar 4.9 Tampilan Input harga jual barang

Pada gambar 4.9 apabila barang yang akan diisi ke display belum memiliki harga jual yang ditetapkan, sistem akan menampilkan jendela input harga terlebih dahulu sebelum proses transfer barang ke display dapat diselesaikan. Pada tampilan ini, pemain diminta untuk memasukkan harga jual barang tersebut sesuai kebutuhannya. Namun, untuk menjaga keseimbangan dalam simulasi, nilai maksimum harga yang dapat dimasukkan akan dibatasi secara otomatis, yaitu tidak boleh melebihi harga pasar (*market price*) ditambah 10%. Dengan adanya batasan ini, pemain tidak dapat menetapkan harga barang yang terlalu tinggi dan tetap berada dalam rentang harga yang wajar sesuai kondisi pasar dalam simulasi. Setelah harga diisi dan dikonfirmasi, proses transfer stok ke display akan dilanjutkan, sekaligus mencatat harga tersebut ke dalam data barang yang ada di display.



Gambar 4.10 Tampilan barang yang sudah direstock

Pada gambar 4.10 Barang-barang yang telah dipajang di rak display akan mulai dibeli oleh customer selama simulasi berjalan. Namun, agar proses transaksi dapat berlangsung, pemain terlebih dahulu harus membuka store atau toko secara manual. Proses membuka store ini dilakukan melalui interaksi tertentu di dalam permainan, biasanya dengan menekan tombol atau memilih opsi *open store*. Setelah store dibuka, customer akan secara otomatis mulai datang dan melakukan pembelian terhadap barang-barang yang tersedia di display sesuai kebutuhan dan preferensi masing-masing customer. Aktivitas ini menjadi bagian utama dari simulasi operasional supermarket, di mana pemain harus memastikan stok barang di display tetap tersedia dan harga yang ditetapkan sesuai kondisi pasar agar penjualan dapat berjalan optimal selama store terbuka.



Gambar 4.11 Tampilan OpenStore

Pada gambar 4.11 ketika pengguna berinteraksi dengan penanda status toko, sistem akan melakukan pengecekan terhadap nilai status yang tersimpan terlebih dahulu. Jika status toko saat ini adalah "*Closed*", maka ketika pengguna mengklik atau berinteraksi dengan penanda tersebut, sistem akan memutar atau mengubah posisi penanda (misalnya dari posisi tertutup ke terbuka) dan secara bersamaan memperbarui nilai status menjadi "*open*". Sebaliknya, jika status toko saat ini adalah "*open*", interaksi yang sama akan memicu penanda untuk berputar kembali (dari terbuka ke tertutup) dan mengubah status menjadi "*closed*". Dengan demikian, mekanisme ini memungkinkan pengguna untuk dengan mudah mengganti status toko hanya dengan satu interaksi, sementara sistem secara otomatis mengelola perubahan nilai dan tampilan penanda status.



Gambar 4.12 Tampilan pelanggan membeli barang

Pada gambar 4.12 ketika status toko bernilai *open*, setiap customer yang melewati depan supermarket akan melakukan pengecekan terhadap status tersebut. Jika status menunjukkan “*open*”, customer dapat masuk ke dalam supermarket untuk melakukan aktivitas belanja (*shooping*). Setelah selesai memilih barang, customer akan menuju ke area kasir dan bergabung dalam antrian untuk melakukan pembayaran. Proses ini memastikan bahwa hanya ketika toko dalam keadaan buka, customer dapat mengakses layanan belanja dan menyelesaikan transaksi di kasir. Dengan demikian, status toko berperan sebagai pengendali akses utama bagi customer sebelum mereka memulai aktivitas belanja di supermarket.



Gambar 4.13 Tampilan antrian *customer*

Pada gambar 4.13 setelah customer bergabung dalam antrian, pemain dapat berinteraksi dengan kasir dengan cara bergerak menuju area putih yang telah ditentukan di sekitar kasir. Area ini berfungsi sebagai zona interaksi tempat pemain dapat mengambil peran sebagai kasir. Ketika pemain melakukan interaksi dengan kasir, antrian customer di depannya akan maju secara otomatis, dan customer yang berada di posisi terdepan akan memulai proses transaksi. Selama proses ini, sistem akan menampilkan daftar barang yang dibeli oleh customer tersebut, memungkinkan pemain (sebagai kasir) untuk melakukan pemindaian (scan) atau pencatatan barang-barang tersebut. Dengan demikian, interaksi ini mensimulasikan proses pembayaran di kasir nyata, di mana pemain bertugas melayani customer dan memproses belanjaan mereka sebelum customer meninggalkan supermarket.



Gambar 4.14 Tampilan pelanggan *checkout*

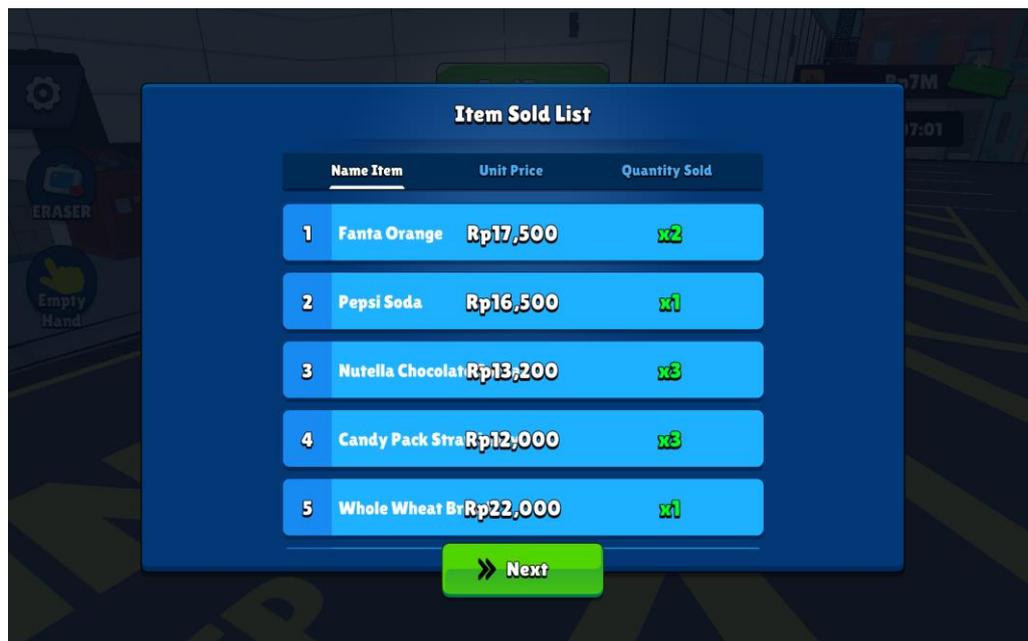
Pada gambar 4.14 ketika customer melakukan *checkout*, pemain dapat menekan atau memilih barang-barang yang ditampilkan di layar. Setiap barang yang ditekan akan bergerak secara otomatis menuju keranjang yang telah disediakan di sisi kiri kasir. Selama proses ini, sistem akan melakukan scan terhadap setiap barang yang dipindahkan, mencatat jenis dan harganya. Setelah semua barang berhasil di-scan, pemain dapat memasukkan total harga dari seluruh barang yang dibeli oleh customer. Selanjutnya, pemain harus memproses pembayaran dengan memasukkan nominal yang sesuai ke dalam *credit card terminal* hingga jumlahnya benar sesuai dengan total harga. Jika pembayaran sudah valid, customer dinyatakan selesai berbelanja (*shopping*) dan akan keluar dari toko. Setelah customer pergi, data penjualan dari transaksi tersebut otomatis tersimpan dalam sistem untuk kemudian ditampilkan di menu *Result Day* di akhir hari. Dengan demikian, seluruh proses dari *checkout*, scan barang, pembayaran, hingga

pencatatan penjualan berjalan secara terstruktur, memungkinkan pemain untuk melacak kinerja toko secara menyeluruh.



Gambar 4.15 Tampilan *Button EndDay*

Ketika waktu operasional toko telah berakhir dan hari dinyatakan selesai, pemain dapat menekan tombol "*End Day*" untuk mengakhiri aktivitas hari tersebut. Setelah tombol ditekan, sistem akan menampilkan sebuah *card list* yang berisi daftar lengkap barang-barang yang terjual sepanjang hari, termasuk detail seperti nama barang, jumlah terjual, dan total pendapatan yang diperoleh. Pemain dapat meninjau informasi ini untuk mengevaluasi penjualan harian. Setelah selesai memeriksa laporan, pemain dapat menekan tombol "*Next*" untuk melanjutkan ke hari berikutnya atau ke menu lainnya, tergantung pada alur permainan. Dengan demikian, fitur "*End Day*" tidak hanya berfungsi sebagai penanda berakhirnya hari, tetapi juga memberikan ringkasan kinerja toko yang berguna untuk perencanaan dan strategi penjualan ke depannya.



	Name Item	Unit Price	Quantity Sold
1	Fanta Orange	Rp17,500	x2
2	Pepsi Soda	Rp16,500	x1
3	Nutella Chocolat	Rp13,200	x3
4	Candy Pack Stra	Rp12,000	x3
5	Whole Wheat Br	Rp22,000	x1

Gambar 4.16 Tampilan *Card List*

Pada gambar 4.16 adalah tampilan menu *card list*, pemain akan melihat tampilan data *komprensif* yang mencakup tiga informasi utama. Pertama, daftar barang yang terjual sepanjang hari, termasuk detail seperti nama barang, jumlah terjual, dan total pendapatan yang dihasilkan. Kedua, informasi mengenai sisa stok barang yang masih tersedia di rak display, membantu pemain memantau persediaan yang hampir habis. Ketiga, hasil perhitungan menggunakan metode *Simple Additive Weighting (SAW)*, yang menganalisis berbagai faktor (seperti tingkat penjualan, stok tersisa, dan profit margin) untuk menentukan prioritas barang yang perlu direstock (ditambah stoknya). Dengan adanya data ini, pemain dapat membuat keputusan yang lebih terarah dalam manajemen inventaris, memastikan barang yang paling dibutuhkan atau menguntungkan tersedia kembali dengan cepat. Menu ini tidak hanya berfungsi sebagai laporan penjualan, tetapi juga sebagai alat strategis untuk mengoptimalkan operasional toko.

Rank	Item Name	Price Unit	Current Stok	Expires in (Days)	Demand Data	Alternative Score
1	Candy Pack Strawberry	19000	9	12	13	0.573
2	Fanta Orange	17500	10	6	12	0.554
3	Camembert Cheese	44000	9	6	13	0.550
4	Dishwater Detergent	9800	10	9	12	0.524

Customers Served :x12  
 0.4      0.2  
 0.3      0.1

Profit: **+453700**

Ok

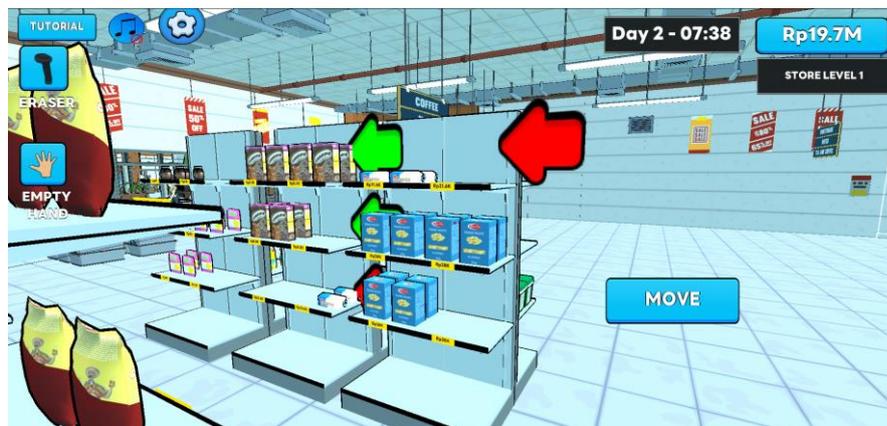
Gambar 4.17 Tampilan Result SAW

Pada gambar 4.17 setelah pemain selesai melihat laporan harian pada menu *card list*, simulasi akan secara otomatis berlanjut ke hari berikutnya. Selama transisi ini, sistem akan memproses pengurangan data barang kadaluarsa, menghapus atau menandai item yang sudah melewati masa berlaku berdasarkan durasi simulasi. Ketika memasuki awal hari kedua, pemain akan menerima notifikasi rekomendasi yang dihasilkan dari perhitungan metode *SAW* pada hari sebelumnya. Notifikasi ini menampilkan informasi penting seperti barang kategori Best Seller, item dengan stok kritis, atau produk prioritas yang perlu segera direstock berdasarkan analisis kinerja penjualan dan kriteria lainnya. Dengan demikian, pemain memiliki panduan strategis untuk mengatur stok dan fokus penjualan di hari baru, meningkatkan efisiensi toko dan memaksimalkan keuntungan. Fitur ini memastikan bahwa keputusan manajemen inventaris berbasis data, sekaligus memperkenalkan dinamika baru setiap harinya dalam simulasi.



Gambar 4.18 Tampilan Notifikasi Rekomendasi

Pada gambar 4.18 notifikasi yang muncul tidak hanya menampilkan barang Best Seller, tetapi juga memberikan daftar kandidat barang di rak yang perlu segera dilakukan restock berdasarkan hasil analisis metode SAW. Barang-barang ini diprioritaskan karena berbagai faktor seperti stok yang menipis, tingkat penjualan tinggi, atau profit margin yang menguntungkan. Setelah menerima notifikasi, pemain dapat bergerak menuju display untuk melihat secara langsung lokasi dan kondisi kandidat barang yang direkomendasikan. Dengan mengecek display, pemain dapat memverifikasi stok yang tersisa dan memutuskan langkah selanjutnya, seperti melakukan pemesanan ulang atau mengatur ulang tata letak barang untuk meningkatkan penjualan. Fitur ini memudahkan pemain dalam mengelola inventaris secara proaktif, memastikan bahwa barang-barang penting selalu tersedia untuk customer, sekaligus menghindari kehilangan peluang penjualan akibat stok kosong.



Gambar 4.19 Tampilan kandidat rekomendasi

Pada gambar 4.19 dalam sistem rekomendasi *restock*, kandidat barang ditandai dengan dua warna berbeda yang memiliki arti spesifik. Barang yang ditandai dengan warna merah merupakan rekomendasi barang yang perlu segera diperhatikan stoknya karena berada dalam kondisi kritis atau hampir habis, namun belum memerlukan *restock* segera. Sementara itu, barang yang ditandai dengan warna hijau menunjukkan kandidat barang yang benar-benar perlu *restock* secepatnya karena stoknya sudah mencapai titik minimum atau termasuk dalam kategori best seller. Untuk melakukan proses *restock*, pemain dapat bergerak menuju area gudang tempat penyimpanan stok cadangan berada. Di gudang ini, pemain bisa mengambil barang-barang yang sesuai dengan rekomendasi warna hijau kemudian memindahkannya ke rak display untuk mengisi stok yang kosong. Sistem warna ini membantu pemain memprioritaskan barang mana yang perlu ditangani terlebih dahulu berdasarkan urgensi kebutuhan *restock*, sehingga manajemen inventaris dapat dilakukan secara lebih efisien dan efektif.



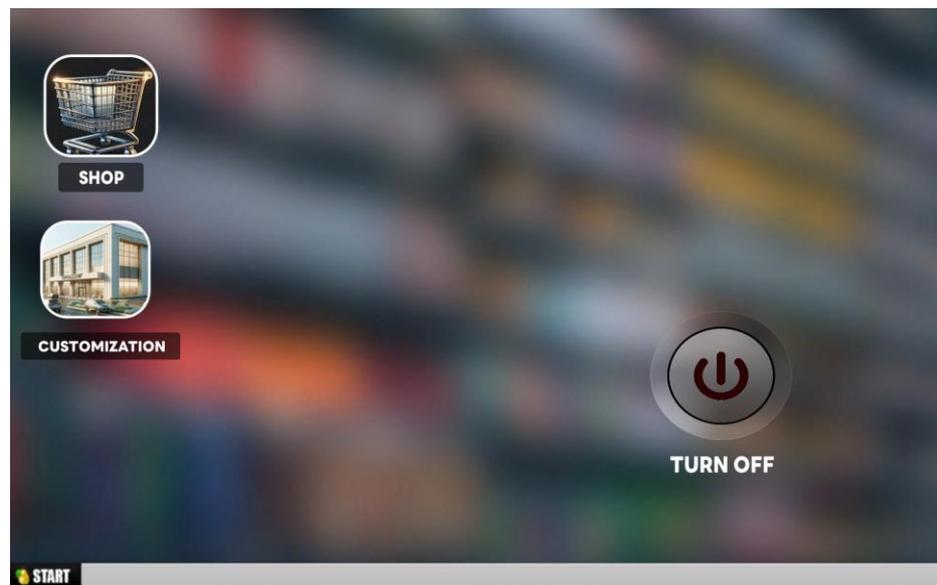
Gambar 4.20 Tampilan Box gudang

Pada gambar 4.20 pada area gudang, pemain memiliki dua opsi untuk melakukan restock barang tergantung pada ketersediaan stok. Jika terdapat box barang yang tersedia di gudang (yang merupakan stok cadangan), pemain dapat melakukan transfer barang ke rak display dengan cara berinteraksi menggunakan buku. Buku ini berfungsi sebagai antarmuka untuk memindahkan barang dari gudang ke area penjualan. Namun, jika tidak ada box barang yang tersedia di gudang (stok kosong), maka pemain perlu melakukan purchase order terlebih dahulu dengan berinteraksi melalui komputer yang tersedia di gudang. Melalui komputer ini, pemain dapat memesan barang baru dari supplier untuk mengisi kembali stok gudang. Sistem ini memungkinkan manajemen inventaris yang fleksibel, di mana pemain harus selalu memantau ketersediaan stok di gudang dan memutuskan apakah cukup melakukan transfer barang atau perlu melakukan pemesanan baru, sehingga menjaga kelancaran pasokan barang di supermarket.



Gambar 4.21 Tampilan interact computer

Pada gambar 4.21 ketika pemain berinteraksi dengan komputer di area gudang, sistem akan menampilkan antarmuka sederhana yang khusus berisi fitur *Shop* (Pembelian). Pada tampilan ini, pemain dapat melihat daftar supplier yang tersedia beserta barang-barang yang mereka sediakan. Setiap item menampilkan informasi penting seperti nama barang, harga per unit, dan minimum order. Pemain dapat memilih supplier dan barang yang ingin dipesan, kemudian menentukan jumlah yang dibutuhkan untuk restock. Sistem tetap memungkinkan pemain untuk melakukan proses *purchase order* secara efektif guna mengisi ulang stok gudang yang kosong. Tampilan yang sederhana ini dirancang untuk memudahkan pemain dalam melakukan pembelian tanpa kompleksitas berlebih, sesuai dengan kebutuhan dasar manajemen inventori dalam simulasi ini.



Gambar 4.22 Tampilan menu komputer

Pada gambar 4.22 pada tampilan komputer, pemain memiliki dua opsi utama yaitu melakukan order barang atau mematikan komputer. Jika pemain memilih untuk melakukan order barang, sistem akan menampilkan inventory barang yang berisi daftar lengkap produk-produk yang dapat dipesan dari supplier. Inventory ini menampilkan informasi penting seperti nama barang, harga per unit, stok yang tersedia, serta minimum order yang harus dipenuhi. Pemain dapat memilih barang yang ingin dipesan sesuai dengan kebutuhan restock, kemudian menentukan jumlah pesanan yang diinginkan. Setelah order barang selesai dilakukan, pemain dapat keluar dari menu inventory dan kembali ke tampilan utama komputer, lalu memilih opsi untuk mematikan komputer jika sudah selesai menggunakan.



Gambar 4.23 Tampilan Shop komputer

Pada sistem ini pembelian di komputer gudang, pemain dapat melakukan proses pembelian barang dengan mekanisme yang sederhana namun fungsional. Pertama pemain memilih barang yang ingin dibeli dari daftar inventory supplier, kemudian menentukan jumlah unit yang akan dipesan sesuai kebutuhan restock. Setelah jumlah ditentukan, barang tersebut akan secara otomatis masuk ke dalam keranjang tersebut, sistem akan langsung memproses pesanan dengan cara *spawn box* fisik secara instan, siap untuk kemudian ditransfer ke rak display ketika dibutuhkan. Mekanisme ini memungkinkan proses restock berjalan efisien tanpa menunggu waktu pengiriman, sehingga pemain dapat langsung mengelola persediaan barang di supermarket. Visual box yang muncul di gudang juga memberikan representasi fisik yang jelas tentang stok yang tersedia, memudahkan pemain dalam mengatur *inventory* mereka.



Gambar 4.24 Tampilan Box Spawn

Setelah box barang muncul (*spawn*) di area penerimaan gudang, pemain dapat melakukan pickup untuk memindahkannya ke lokasi penyimpanan yang lebih permanen di dalam gudang. Proses ini melibatkan interaksi langsung dengan box tersebut, dimana pemain cukup mendekati dan menekan tombol interaksi untuk mengangkat dan memindahkan box ke posisi yang diinginkan. Setelah box-barang tersusun rapi di gudang, pemain kemudian dapat memindahkan barang-barang tersebut ke rak display dengan berinteraksi menggunakan buku seperti yang dijelaskan sebelumnya. Buku ini berfungsi sebagai antarmuka transfer barang yang mencatat dan mengelola perpindahan stok dari gudang ke area penjualan. Siklus simulasi kemudian akan berulang dimulai dari pengisian rak, membuka toko, melayani pelanggan, hingga mengevaluasi penjualan di akhir hari. Pemain akan terus menjalankan siklus ini secara berulang selama simulasi berjalan, dengan tambahan dinamika seperti menerima rekomendasi restock dari sistem SAW, memproses pembelian barang baru ketika stok menipis, dan mengoptimalkan tata

letak barang berdasarkan data penjualan. Pola permainan yang berulang ini dirancang untuk memberikan pengalaman manajemen toko yang komprehensif, sekaligus melatih pemain dalam membuat keputusan bisnis berbasis data secara konsisten.

#### **4.2 Implementasi Metode SAW**

Pengimplementasian sistem dilakukan pada simulasi operasional supermarket dalam game Supermarket Simulator. Sistem ini menggunakan metode *Simple Additive Weighting (SAW)* untuk membantu proses pengambilan keputusan dalam menentukan prioritas item yang akan direstock berdasarkan beberapa kriteria, yaitu harga, stok, masa kadaluarsa, dan permintaan. Metode ini diimplementasikan untuk memberikan sistem perhitungan skor alternatif barang yang akan menjadi dasar pengambilan keputusan dalam fitur *Result Day*. Fitur ini akan merekap performa barang berdasarkan empat kriteria utama, yaitu harga, stok, masa kadaluarsa, dan permintaan pelanggan. Implementasi dilakukan dengan mengikuti langkah-langkah perhitungan yang telah dijelaskan pada bab sebelumnya, kemudian dikembangkan dalam bentuk script menggunakan bahasa pemrograman *C#* di dalam *Unity*. Data yang digunakan bersifat dinamis, dimana stok dan permintaan akan terus berubah sesuai interaksi pemain di dalam game.

Dalam *result day*, langkah yang dilakukan untuk mengimplementasi metode *SAW* dalam pengelolaan barang yaitu sesuai pada bab 3.3, langkah awal diawali dengan menyusun matriks data untuk barang-barang di supermarket. Kemudian dinormalisasi sesuai dengan kriteria. Setelah itu, dihitunglah skor alternatif dari data barang, dan terakhir pengurutan ranking.

a. Penyusunan Matriks Data Barang

Data barang disusun dalam sebuah matriks dengan nilai-nilai berikut: harga, stok, masa kadaluarsa, dan permintaan.

Tabel 4.1 *Pseudocode* Matriks Data Barang

```
function MatriksData(dataBarang)
    matriksData = []
    for i = 0 to length of dataBarang
        nilaiBarang = [dataBarang[i].harga, dataBarang[i].stok,
dataBarang[i].kadaluarsa, dataBarang[i].permintaan]
        matriksData.add(nilaiBarang)
    end for
    return matriksData
end function
```

*Pseudocode* 4.1 merupakan sebuah fungsi yang digunakan untuk membentuk sebuah matriks data dari daftar barang yang telah disediakan dalam game. Fungsi ini bertujuan untuk mengumpulkan seluruh nilai atribut penting dari setiap barang, yaitu harga, stok, masa kadaluarsa, dan permintaan, ke dalam sebuah list dua dimensi (matriks) yang nantinya akan digunakan untuk proses perhitungan metode SAW.

b. Normalisasi Data Barang

Data Normalisasi dihitung untuk setiap kriteria dengan menyesuaikan tipe kriteriannya. Untuk kriteria harga, stok, dan masa kadaluarsa yang termasuk tipe cost, nilai yang lebih kecil akan lebih baik, sehingga dilakukan normalisasi dengan membalik nilai aslinya terhadap nilai maksimum. Sedangkan untuk kriteria permintaan yang merupakan tipe benefit, nilai lebih besar akan lebih baik, sehingga

dinormalisasi dengan membandingkan nilai terhadap nilai maksimum tanpa dibalik.

Tabel 4.2 *Pseudocode* Normalisasi Data Barang

```
function NormalisasiData(matriksData, listKriteria)

    hargaMax = matriksData[0].harga
    stokMax = matriksData[0].stok
    kadaluarsaMax = matriksData[0].kadaluarsa
    permintaanMax = matriksData[0].permintaan

    // Cari nilai Max dari masing-masing kriteria
    for i = 1 to length of matriksData
        if matriksData[i].harga > hargaMax
            hargaMax = matriksData[i].harga
        end if
        if matriksData[i].stok > stokMax
            stokMax = matriksData[i].stok
        end if
        if matriksData[i].kadaluarsa > kadaluarsaMax
            kadaluarsaMax = matriksData[i].kadaluarsa
        end if
        if matriksData[i].permintaan > permintaanMax
            permintaanMax = matriksData[i].permintaan
        end if
    end for

    for i = 0 to length of matriksData
        for j = 0 to length of listKriteria

            if listKriteria[j] = "Harga"
                matriksData[i].normalisasiHarga = 1 -
(matriksData[i].harga / hargaMax)
            else if listKriteria[j] = "Stok"
                matriksData[i].normalisasiStok = 1 -
(matriksData[i].stok / stokMax)
            else if listKriteria[j] = "Masa Kadaluarsa"
```

```

        matriksData[i].normalisasiKadaluarsa = 1 -
(matriksData[i].kadaluarsa / kadaluarsaMax)
        else if listKriteria[j] = "Permintaan"
            matriksData[i].normalisasiPermintaan =
matriksData[i].permintaan / permintaanMax
        end if

    end for
end for

end function

```

Sesuai dengan rumus pada bab sebelumnya, *Pseudocode* 4.1 menampilkan fungsi normalisasi data barang dari hasil matriks sebelumnya. Hasil dari fungsi ini kemudian disimpan ke dalam properti khusus disetiap alternatif.

c. Skor Alternatif

Setelah data dinormalisasi, langkah selanjutnya adalah menghitung skor untuk setiap alternatif. Setiap alternatif akan diberi bobot untuk setiap kriteria sesuai pada tabel 3.3. Skor dihitung dengan menjumlahkan hasil perkalian antara nilai normalisasi dan bobot kriteria untuk setiap alternatif.

Tabel 4.3 *Pseudocode* Skor Alternatif

```

function HitungSkorAlternatif(matriksData, bobot)
    for i = 0 to length of matriksData
        skor = 0
        for j = 0 to length of bobot
            skor = skor + (matriksData[i].normalisasi[j] *
bobot[j])
        end for
        matriksData[i].skor = skor
    end for

end function

```

*Pseudocode* 4.3 menampilkan sebuah fungsi yang digunakan untuk menghitung skor akhir dari masing-masing alternatif setelah dilakukan proses normalisasi dan diketahui nilai bobot dari tiap kriteria. fungsi ini melakukan perulangan sebanyak jumlah alternatif yang tersedia di dalam ‘matriksData’. Di dalam setiap iterasi alternatif, nilai skor awal diset ke 0. Kemudian, dilakukan proses perulangan kedua sebanyak jumlah kriteria. Pada perulangan ini, dilakukan perhitungan nilai normalisasi alternatif terhadap kriteria yang sedang dihitung, lalu dikalikan dengan bobot kriteria tersebut. Nilai hasil perkalian kemudian dijumlahkan ke dalam variabel skor milik alternatif tersebut. Setelah semua kriteria untuk satu alternatif dihitung, nilai skor total tersebut disimpan ke dalam atribut skor milik alternatif yang bersangkutan. Proses ini diulang untuk semua alternatif dalam data.

d. Pengurutan alternatif berdasarkan skor

Setelah semua skor dihitung, alternatif-alternatif diurutkan berdasarkan skor tertinggi hingga terendah. Alternatif dengan skor tertinggi dianggap sebagai pilihan terbaik, sedangkan alternatif dengan skor terendah dianggap sebagai pilihan yang kurang optimal.

Tabel 4.4 *Pseudocode Descending Alternatif*

```
function DescendingAlternatif(matriksData)
  n = length of matriksData
  for i = 0 to n-2
    for j = 0 to n-2
      if matriksData[j].skor < matriksData[j+1].skor then
        // Tukar posisi
        temp = matriksData[j]
        matriksData[j] = matriksData[j+1]
        matriksData[j+1] = temp
```

```
        end if
    end for
end for

return matriksData

end function
```

*Pseudocode* 4.4 dilakukan untuk mengurutkan alternatif berdasarkan nilai skor yang telah diperoleh dari hasil perhitungan sebelumnya. Proses ini bertujuan untuk menentukan urutan prioritas barang yang paling direkomendasikan hingga yang kurang direkomendasikan sesuai dengan hasil skor tertinggi ke terendah. Tahap pertama dimulai dengan menginisialisasi variabel 'n' yang berisi panjang dari data alternatif (jumlah alternatif yang akan diurutkan). Selanjutnya dilakukan perulangan for pertama dari indeks 0 hingga 'n-2' yang berfungsi untuk menentukan banyaknya proses perbandingan antar elemen dalam array. Kemudian, perulangan for kedua dijalankan dari indeks 0 hingga 'n-2', di mana pada setiap iterasi perulangan ini akan dilakukan perbandingan antara skor pada elemen indeks ke-j dengan skor pada elemen indeks ke j+1. Jika nilai skor pada indeks ke-j lebih kecil dibandingkan skor di indeks ke j+1, maka kedua elemen tersebut akan ditukar. Proses penukaran ini dilakukan dengan memanfaatkan variabel sementara temp untuk menyimpan data sementara saat proses pertukaran berlangsung. Tahap terakhir adalah mengembalikan array '*matriksData*' yang sudah terurut secara descending, di mana hasil nilai skor tertinggi berada di urutan paling atas yang ditandai sebagai kandidat rekomendasi restock dan nilai skor terendah di bagian

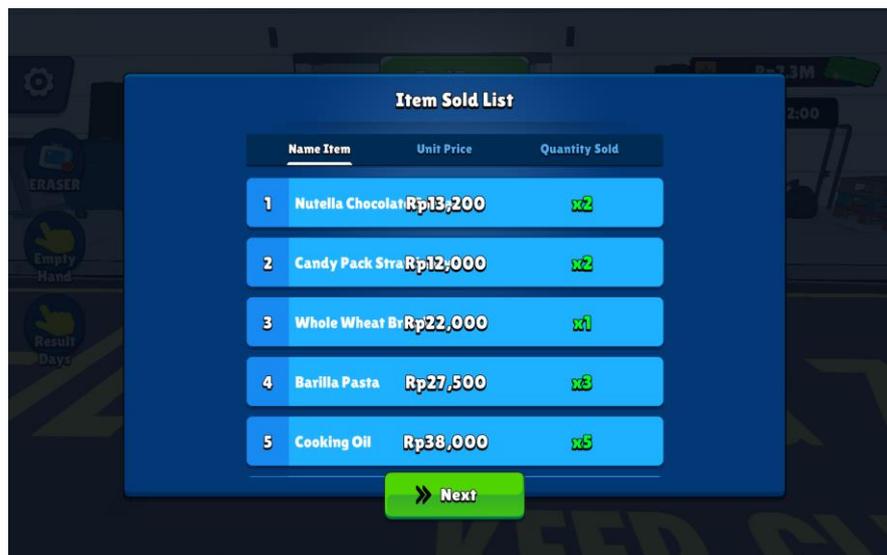
bawah yang ditandai sebagai kandidat rekomendasi barang yang perlu diperhatikan stok nya.

### 4.3 Hasil dan Analisis Implementasi SAW

Pengujian dilakukan setelah pemain menyelesaikan hari pada simulasi penjualan barang. Pada tahap ini, data-data seperti harga barang, jumlah stok, masa kadaluarsa, dan tingkat permintaan sudah tercatat di dalam sistem. Data tersebut kemudian diproses melalui implementasi diatas untuk menghasilkan rekomendasi barang yang paling menguntungkan. Hasil perhitungan akan muncul setelah tombol *endday* ditekan. Ketika tombol ditekan, sistem akan memproses seluruh data barang menggunakan tahapan perhitungan seperti yang dijelaskan pada bab sebelumnya. Setelah itu, Hasil rekomendasi kandidat restock dan kandidat yang perlu diperhatikan stoknya akan ditampilkan dalam bentuk *notification* pada saat gameplay.

#### a. Percobaan Pertama

Pada percobaan pertama, menu *resultday* di hari pertama menampilkan laporan komprehensif mengenai aktivitas penjualan toko. Pemain dapat melihat daftar lengkap barang-barang yang terjual sepanjang hari tersebut, yang disajikan dalam bentuk tabel yang terorganisir. Tampilan ini mencakup nama produk, jumlah unit yang terjual, harga jual. Data ini tidak hanya mencatat transaksi yang berhasil dilakukan. Visualisasi data dibuat jelas dengan tampilan *interface* untuk memudahkan pemahaman terhadap performa penjualan. Informasi ini menjadi dasar penting bagi pemain untuk mengevaluasi strategi penjualan dan menentukan prioritas restock untuk hari-hari berikutnya dalam simulasi.



Gambar 4.25 Tampilan barang yang terjual percobaan pertama

Pada gambar 4.25 pada simulasi percobaan pertama, hasil penjualan pada hari tersebut dilakukan terhadap seluruh inventori barang dengan menggunakan data interaksi pemain pada tahapan simulasi seperti pada gambar 4.16. Data diatas berisi parameter-parameter kunci yang digunakan sebagai kriteria perhitungan, meliputi jumlah pembelian, harga jual di rak, nama barang. Setiap kriteria diberi bobot tertentu sesuai tingkat kepentingannya dalam penentuan prioritas restock. Proses normalisasi data dilakukan terlebih dahulu untuk menyamakan skala pengukuran antar kriteria yang berbeda, kemudian dilanjutkan dengan proses perankingan berdasarkan nilai preferensi tertinggi.

Tabel 4.5 Data Barang percobaan kedua

ID Alternatif	Harga	Stok	Masa Kadaluarsa	Permintaan
$C_1$	Rp18.000	12	9 hari	10
$C_2$	Rp17.500	12	6 hari	10
$C_3$	Rp15.000	12	4 hari	10
$C_4$	Rp13.200	10	18 hari	12
$C_5$	Rp12.000	10	12 hari	12
$C_6$	Rp24.200	12	9 hari	10
$C_7$	Rp22.000	11	7 hari	11

ID Alternatif	Harga	Stok	Masa Kadaluarsa	Permintaan
$C_8$	Rp27.500	9	24 hari	13
$C_9$	Rp25.000	12	7 hari	10
$C_{10}$	Rp38.000	7	12 hari	15
$C_{11}$	Rp15.000	8	24 hari	14
$C_{12}$	Rp19.800	11	9 hari	11
$C_{13}$	Rp16.500	12	6 hari	10
$C_{14}$	Rp44.000	11	6 hari	11
$C_{15}$	Rp16.500	11	6 hari	11

Tabel 4.5 menyajikan data 15 alternatif barang yang didapatkan dari interaksi pemain pada percobaan pertama ( $C_1$  hingga  $C_{15}$ ). Data menunjukkan variasi karakteristik produk, seperti  $C_{10}$  dengan harga tertinggi dan stok terbatas, atau  $C_{11}$  yang memiliki masa kadaluarsa panjang meski harganya ekonomis. Kriteria-kriteria ini akan dinormalisasi dan diberi bobot untuk menghasilkan skor prioritas restock, di mana barang dengan permintaan tinggi. Hasil perhitungan SAW akan mengidentifikasi barang yang perlu segera direstock (misalnya  $C_4$  dengan permintaan 12 unit dan stok 10) atau yang masih aman (seperti  $C_5$  dengan stok 10 unit dan kadaluarsa 12 hari).

Rank	Item Name	Price Unit	Current Stok	Expires In (Days)	Demand Data	Alternative Score
1	Cooking Oil	38000	7	12	15	0.582
2	Candy Pack Strawberry	25000	10	12	12	0.480
3	Sunlight Dish Soap	15000	8	24	14	0.480
4	Whipping Cream	16500	11	6	11	0.467

Customers Served : 10  
 B.Stock = 0.4    B.Expired = 0.2  
 B.Demand = 0.3    B.Price = 0.1  
 Profit: +485200

Gambar 4.26 Tampilan rincian data yang diperoleh percobaan pertama

Pada gambar 4.26 Setelah seluruh data barang terkumpul, sistem melakukan perhitungan rekomendasi restock menggunakan metode *Simple Additive Weighting (SAW)* yang menghasilkan nilai preferensi untuk setiap barang. Proses perhitungan simulasi dilakukan pada tahapan gambar 4.17 diawali dengan normalisasi matriks keputusan untuk menyamakan skala seluruh kriteria, kemudian dilanjutkan dengan mengalikan nilai ternormalisasi dengan bobot kriteria yang telah ditetapkan. Hasil akhir perhitungan menunjukkan nilai preferensi masing-masing barang dalam rentang 0 hingga 1, di mana nilai lebih tinggi mengindikasikan urgensi restock yang lebih besar.

Tabel 4.6 Rating Alternatif percobaan pertama

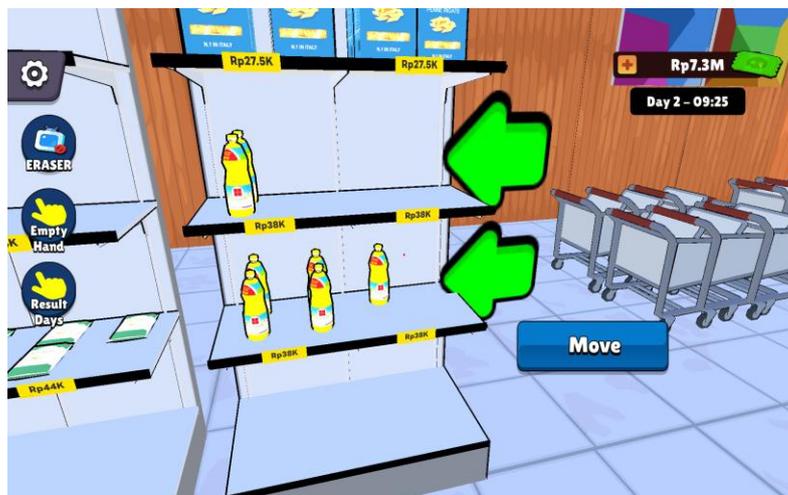
<b>ID Alternatif</b>	<b>Nama Barang</b>	<b>Skor Alternatif</b>	<b>Peringkat</b>
$C_1$	Cola	0,385	14
$C_2$	Fanta Orange	0,411	10
$C_3$	Lay's Potato Chips	0,433	7
$C_4$	Nutella Chocolate Spread	0,427	8
$C_5$	Candy Pack	0,480	2
$C_6$	Chocapic Cereal	0,371	15
$C_7$	Whole Wheat Bread	0,446	5
$C_8$	Barilla Pasta	0,399	12
$C_9$	Fresh Milk	0,386	13
$C_{10}$	Cooking Oil	0,582	1
$C_{11}$	Sunlight Dish Soap	0,480	3
$C_{12}$	Dishwater Detergent	0,434	6
$C_{13}$	Pepsi Soda	0,413	9
$C_{14}$	Camembert Cheese	0,406	11
$C_{15}$	Whipping Cream	0,467	4

Pada tabel 4.6 yang menampilkan hasil peringkat alternatif barang pada percobaan pertama, terlihat bahwa Cooking Oil ( $C_{10}$ ) menempati posisi teratas dengan skor SAW sebesar 0,582, mengungguli 14 alternatif lainnya. Skor tinggi ini merupakan hasil dari kombinasi optimal empat kriteria, yaitu harga sebesar Rp38.000 yang meskipun tergolong premium, namun sebanding dengan tingkat permintaan tertinggi yaitu 15 unit per hari, stok yang berada dalam kondisi kritis sebanyak 7 unit yang hampir tidak mencukupi kebutuhan harian, serta masa kadaluarsa selama 12 hari yang masih dalam kategori aman. Kondisi tersebut menjadikan Cooking Oil ( $C_{10}$ ) sebagai prioritas utama untuk direstock demi mencegah terjadinya kekosongan stok. Sebaliknya, Chocapic Cereal ( $C_6$ ) menempati peringkat terbawah dengan skor sebesar 0,371. Meskipun barang ini memiliki stok sebanyak 12 unit dan masa kadaluarsa selama 9 hari yang masih tergolong normal, harga relatif tinggi sebesar Rp24.200 serta tingkat permintaan yang hanya mencapai 10 unit per hari menjadikannya kurang mendesak untuk segera dilakukan restock. Hasil ini menunjukkan bahwa metode SAW mampu memberikan prioritas restock berdasarkan analisis multidimensi yang mempertimbangkan kombinasi seluruh kriteria secara proporsional, bukan hanya berdasarkan satu faktor tunggal seperti jumlah stok atau permintaan saja.



Gambar 4.27 Tampilan notifikasi kandidat percobaan pertama

Pada gambar 4.27 ditampilkan antarmuka dalam game yang memperlihatkan notifikasi rekomendasi restock setelah percobaan pertama selesai dilakukan seperti pada tahapan gambar 4.18. Notifikasi ini muncul secara otomatis di layar pemain, menampilkan daftar barang yang teridentifikasi sebagai kandidat restock berdasarkan hasil perhitungan metode *SAW*. Visual notifikasi didesain mencolok dengan warna untuk menandakan tingkat urgensi untuk restock segera. Fitur ini tidak hanya memandu pemain dalam pengambilan keputusan manajemen inventori, tetapi juga menambah elemen strategis dalam alur gameplay, di mana pemain harus merespons rekomendasi sistem secara cepat untuk menjaga kelancaran operasional toko.



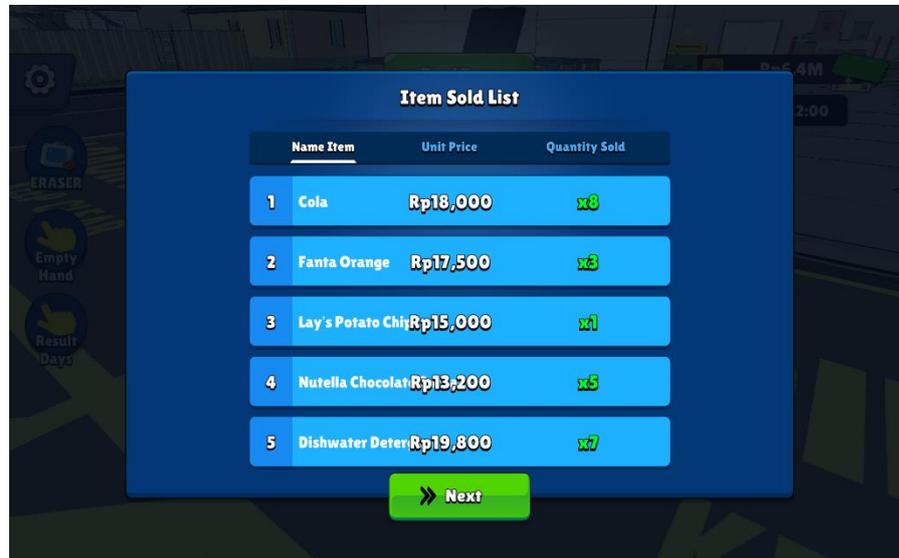
Gambar 4.28 Tampilan kandidat barang di display percobaan pertama

Pada gambar 4.28 ditampilkan bahwa Cooking Oil menjadi barang yang direkomendasikan untuk direstock pada display seperti pada gambar 4.19. Rekomendasi ini sejalan dengan hasil kalkulasi perhitungan menggunakan metode SAW pada tahap sebelumnya, di mana Cooking Oil memperoleh nilai preferensi tertinggi dibandingkan alternatif barang lainnya. Hal ini menunjukkan konsistensi antara proses perhitungan prioritas dan implementasi notifikasi rekomendasi restock dalam antarmuka permainan.

b. Percobaan Kedua

Pada percobaan kedua, menu *resultday* di hari kedua menampilkan laporan komprehensif mengenai aktivitas penjualan toko. Pemain dapat melihat daftar lengkap barang-barang yang terjual sepanjang hari tersebut, yang disajikan dalam bentuk tabel yang terorganisir. Tampilan ini mencakup nama produk, jumlah unit yang terjual, harga jual. Data ini tidak hanya mencatat transaksi yang berhasil dilakukan. Visualisasi data dibuat jelas dengan tampilan interface untuk memudahkan pemahaman terhadap performa penjualan. Informasi ini menjadi

dasar penting bagi pemain untuk mengevaluasi strategi penjualan dan menentukan prioritas restock untuk hari-hari berikutnya dalam simulasi.



Gambar 4.29 Tampilan barang yang terjual percobaan kedua

Pada gambar 4.29 pada simulasi percobaan kedua, hasil penjualan pada hari tersebut dilakukan terhadap seluruh inventori barang dengan menggunakan data interaksi pemain. Data diatas berisi parameter-parameter kunci yang digunakan sebagai kriteria perhitungan, meliputi jumlah pembelian, harga jual di rak, nama barang. Setiap kriteria diberi bobot tertentu sesuai tingkat kepentingannya dalam penentuan prioritas restock. Proses normalisasi data dilakukan terlebih dahulu untuk menyamakan skala pengukuran antar kriteria yang berbeda, kemudian dilanjutkan dengan proses perankingan berdasarkan nilai preferensi tertinggi.

Tabel 4. 7 Data Barang percobaan kedua

ID Alternatif	Harga	Stok	Masa Kadaluarsa	Permintaan
$C_1$	Rp18.000	16	8 hari	18
$C_2$	Rp17.500	21	5 hari	13
$C_3$	Rp15.000	19	17 hari	15
$C_4$	Rp13.200	10	18 hari	12
$C_5$	Rp12.000	12	11 hari	10

ID Alternatif	Harga	Stok	Masa Kadaluarsa	Permintaan
$C_6$	Rp24.200	12	8 hari	10
$C_7$	Rp22.000	12	6 hari	10
$C_8$	Rp27.500	12	23 hari	10
$C_9$	Rp25.000	12	6 hari	10
$C_{10}$	Rp38.000	18	11 hari	10
$C_{11}$	Rp15.000	12	23 hari	10
$C_{12}$	Rp19.800	17	8 hari	17
$C_{13}$	Rp16.500	12	5 hari	10
$C_{14}$	Rp44.000	12	5 hari	10
$C_{15}$	Rp16.500	12	5 hari	10

Tabel 4.7 menyajikan data 15 alternatif barang yang diperoleh dari interaksi pemain pada percobaan kedua ( $C_1$  hingga  $C_{15}$ ). Data ini memperlihatkan variasi karakteristik produk, seperti  $C_{10}$  yang memiliki harga tertinggi sebesar Rp38.000 dengan stok 18 unit dan masa kadaluarsa 11 hari, serta  $C_{11}$  yang menawarkan harga ekonomis sebesar Rp15.000 namun dengan masa kadaluarsa paling panjang, yaitu 23 hari. Setiap barang memiliki tingkat permintaan yang bervariasi, di antaranya  $C_1$  dengan permintaan tertinggi sebanyak 18 unit per hari. Seluruh kriteria ini — harga, stok, masa kadaluarsa, dan permintaan akan dinormalisasi dan diberi bobot sesuai tingkat kepentingannya untuk menghasilkan skor prioritas restock.

Rank	Item Name	Price Unit	Current Stok	Expires in (Days)	Demand Data	Alternative Score
1	Cola	18000	16	8	18	0.611
2	Pepsi Soda	16500	12	5	10	0.577
3	Whipping Cream	16500	12	5	10	0.577
4	Dishwater Deterge	19800	17	8	17	0.573

Customers Served: 12  
 Profit: +416100  
 B.Stock = 0.4    B.Expired = 0.2  
 B.Demand = 0.3    B.Price = 0.1

Gambar 4.30 Tampilan rincian data yang diperoleh percobaan kedua

Pada gambar 4.30 Setelah seluruh data barang terkumpul, sistem melakukan perhitungan rekomendasi restock menggunakan metode *Simple Additive Weighting* (SAW) yang menghasilkan nilai preferensi untuk setiap barang. Proses perhitungan diawali dengan normalisasi matriks keputusan untuk menyamakan skala seluruh kriteria, kemudian dilanjutkan dengan mengalikan nilai ternormalisasi dengan bobot kriteria yang telah ditetapkan. Hasil akhir perhitungan menunjukkan nilai preferensi masing-masing barang dalam rentang 0 hingga 1, di mana nilai lebih tinggi mengindikasikan urgensi restock yang lebih besar.

Tabel 4.8 Rating alternatif percobaan kedua

ID Alternatif	Nama Barang	Skor Alternatif	Peringkat
$C_1$	Cola	0,611	1
$C_2$	Fanta Orange	0,468	10
$C_3$	Lay's Potato Chips	0,423	13
$C_4$	Nutella Chocolate Spread	0,442	11
$C_5$	Candy Pack	0,535	7
$C_6$	Chocapic Cereal	0,533	8
$C_7$	Whole Wheat Bread	0,556	5
$C_8$	Barilla Pasta	0,395	14
$C_9$	Fresh Milk	0,549	6
$C_{10}$	Cooking Oil	0,372	15
$C_{11}$	Sunlight Dish Soap	0,424	12
$C_{12}$	Dishwater Detergent	0,573	4
$C_{13}$	Pepsi Soda	0,577	2
$C_{14}$	Camembert Cheese	0,514	9
$C_{15}$	Whipping Cream	0,577	3

Pada tabel 4.8 yang menampilkan hasil peringkat alternatif pada percobaan kedua, terlihat bahwa Cola  $C_1$  menempati posisi teratas dengan skor SAW sebesar 0,611, mengungguli 14 alternatif lainnya. Skor tinggi ini diperoleh berkat kombinasi nilai optimal dari empat kriteria, yaitu harga Rp18.000 yang masih kompetitif, stok mencukupi sebanyak 16 unit, masa kadaluarsa 8 hari yang masih

aman, serta tingkat permintaan tertinggi sebesar 18 unit per hari. Kondisi ini menjadikan Cola ( $C_1$ ) sebagai prioritas utama untuk direstock guna menjaga ketersediaan barang yang memiliki permintaan sangat tinggi. Sebaliknya, Cooking Oil ( $C_{10}$ ) justru menempati peringkat terbawah dengan skor SAW sebesar 0,372. Meskipun produk ini memiliki harga premium sebesar Rp38.000 dan stok sebanyak 18 unit dengan masa kadaluarsa 11 hari, tingkat permintaan yang hanya 10 unit per hari membuatnya berada dalam kategori kurang mendesak untuk segera dilakukan restock. Hasil ini menunjukkan bahwa metode SAW mampu menyusun prioritas restock secara objektif berdasarkan kombinasi multidimensi dari seluruh kriteria yang tersedia, bukan sekadar mempertimbangkan satu faktor saja seperti harga atau stok, melainkan keseluruhan variabel secara proporsional.



Gambar 4.31 Tampilan notifikasi kandidat percobaan kedua

Pada gambar 4.31 ditampilkan antarmuka dalam game yang memperlihatkan notifikasi rekomendasi restock setelah percobaan kedua selesai dilakukan. Notifikasi ini muncul secara otomatis di layar pemain, menampilkan

daftar barang yang teridentifikasi sebagai kandidat restock berdasarkan hasil perhitungan metode SAW. Visual notifikasi didesain mencolok dengan warna untuk menandakan tingkat urgensi untuk restock segera. Fitur ini tidak hanya memandu pemain dalam pengambilan keputusan manajemen inventori, tetapi juga menambah elemen strategis dalam alur gameplay, di mana pemain harus merespons rekomendasi sistem secara cepat untuk menjaga kelancaran operasional toko.

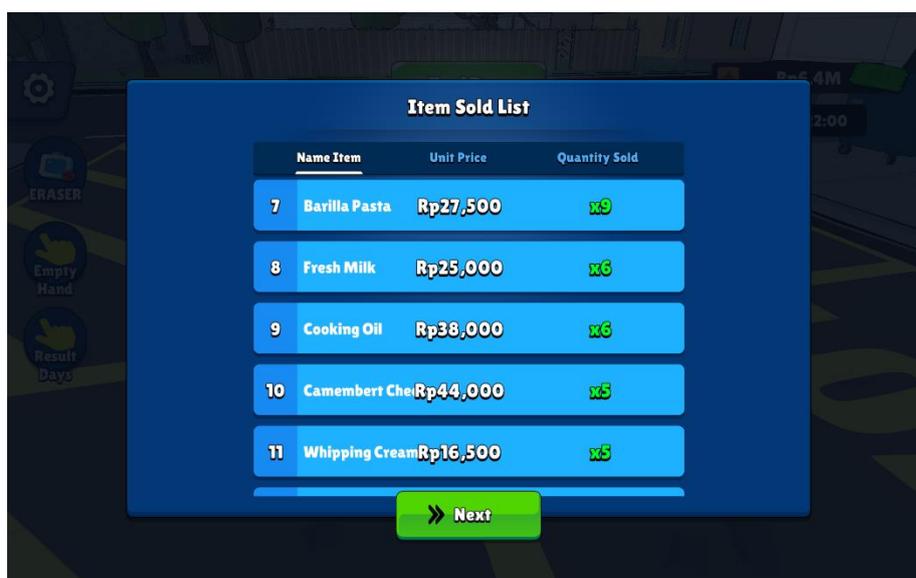


Gambar 4.32 Tampilan kandidat barang di display percobaan kedua

Pada gambar 4.32 ditampilkan bahwa cola menjadi barang yang direkomendasikan untuk direstock. Rekomendasi ini sejalan dengan hasil kalkulasi perhitungan menggunakan metode SAW pada tahap sebelumnya, di mana cola memperoleh nilai preferensi tertinggi dibandingkan alternatif barang lainnya. Hal ini menunjukkan konsistensi antara proses perhitungan prioritas dan implementasi notifikasi rekomendasi restock dalam antarmuka permainan.

c. Percobaan Ketiga

Pada percobaan ketiga, menu *resultday* di hari ketiga menampilkan laporan komprehensif mengenai aktivitas penjualan toko. Pemain dapat melihat daftar lengkap barang-barang yang terjual sepanjang hari tersebut, yang disajikan dalam bentuk tabel yang terorganisir. Tampilan ini mencakup nama produk, jumlah unit yang terjual, harga jual. Data ini tidak hanya mencatat transaksi yang berhasil dilakukan. Visualisasi data dibuat jelas dengan tampilan interface untuk memudahkan pemahaman terhadap performa penjualan. Informasi ini menjadi dasar penting bagi pemain untuk mengevaluasi strategi penjualan dan menentukan prioritas restock untuk hari-hari berikutnya dalam simulasi.



	Name Item	Unit Price	Quantity Sold
7	Barilla Pasta	Rp27,500	x9
8	Fresh Milk	Rp25,000	x6
9	Cooking Oil	Rp38,000	x6
10	Camembert Cheese	Rp44,000	x5
11	Whipping Cream	Rp16,500	x5

Gambar 4.33 Tampilan barang yang terjual percobaan ketiga

Pada gambar 4.33 pada simulasi percobaan ketiga, hasil penjualan pada hari tersebut dilakukan terhadap seluruh inventori barang dengan menggunakan data interaksi pemain. Data diatas berisi parameter-parameter kunci yang digunakan sebagai kriteria perhitungan, meliputi jumlah pembelian, harga jual di rak, nama

barang. Setiap kriteria diberi bobot tertentu sesuai tingkat kepentingannya dalam penentuan prioritas restock. Proses normalisasi data dilakukan terlebih dahulu untuk menyamakan skala pengukuran antar kriteria yang berbeda, kemudian dilanjutkan dengan proses perankingan berdasarkan nilai preferensi tertinggi.

Tabel 4.9 Data Barang pada percobaan ketiga

<b>ID Alternatif</b>	<b>Harga</b>	<b>Stok</b>	<b>Masa Kadaluaarsa</b>	<b>Permintaan</b>
$C_1$	Rp18.000	20	6 hari	14
$C_2$	Rp17.500	24	3 hari	10
$C_3$	Rp15.000	22	1 hari	12
$C_4$	Rp13.200	24	15 hari	10
$C_5$	Rp12.000	6	9 hari	16
$C_6$	Rp24.200	7	6 hari	15
$C_7$	Rp22.000	9	4 hari	13
$C_8$	Rp27.500	11	21 hari	19
$C_9$	Rp25.000	6	4 hari	16
$C_{10}$	Rp38.000	28	9 hari	16
$C_{11}$	Rp15.000	10	21 hari	24
$C_{12}$	Rp19.800	13	6 hari	21
$C_{13}$	Rp16.500	4	3 hari	18
$C_{14}$	Rp44.000	19	3 hari	15
$C_{15}$	Rp16.500	19	3 hari	15

Tabel 4.9 menyajikan data 15 alternatif barang yang diperoleh dari interaksi pemain pada percobaan ketiga ( $C_1$  hingga  $C_{15}$ ). Data ini memperlihatkan variasi karakteristik produk, seperti  $C_{10}$  yang memiliki harga tertinggi sebesar Rp38.000 dengan stok terbanyak sebanyak 28 unit dan masa kadaluarsa selama 9 hari, serta  $C_{11}$  yang menawarkan harga ekonomis sebesar Rp15.000 namun dengan masa kadaluarsa paling panjang, yaitu 21 hari. Setiap barang juga memiliki tingkat permintaan yang beragam, di antaranya  $C_{11}$  dengan permintaan tertinggi sebanyak 24 unit per hari, diikuti oleh  $C_{12}$  sebanyak 21 unit, serta  $C_8$  sebanyak 19 unit per hari. Seluruh kriteria ini harga, stok, masa kadaluarsa, dan permintaa akan

dinormalisasi dan diberi bobot sesuai tingkat kepentingannya untuk menghasilkan skor prioritas restock yang lebih objektif. Proses ini memungkinkan sistem mengidentifikasi barang mana yang perlu segera direstock berdasarkan kondisi multidimensi, bukan hanya dari satu faktor tunggal.



Rank	Item Name	Price Unit	Current Stok	Expires In (Days)	Demand Data	Alternative Score
1	Pepsi Soda	16500	4	3	18	0.803
2	Fresh Milk	25000	6	4	16	0.721
3	Candy Pack Strawberry	18000	6	9	16	0.702
4	Chocapic Cereal	24200	7	6	15	0.677

Customers Served :x8  
 B.Stock = 0.4    B.Expired = 0.2  
 B.Demand = 0.3    B.Price = 0.1

Profit: +569600

Ok

Gambar 4.34 Tampilan rincian data yang diperoleh percobaan ketiga

Pada gambar 4.34 Setelah seluruh data barang terkumpul, sistem melakukan perhitungan rekomendasi restock menggunakan metode *Simple Additive Weighting* (SAW) yang menghasilkan nilai preferensi untuk setiap barang. Proses perhitungan diawali dengan normalisasi matriks keputusan untuk menyamakan skala seluruh kriteria, kemudian dilanjutkan dengan mengalikan nilai ternormalisasi dengan bobot kriteria yang telah ditetapkan. Hasil akhir perhitungan menunjukkan nilai preferensi masing-masing barang dalam rentang 0 hingga 1, di mana nilai lebih tinggi mengindikasikan urgensi restock yang lebih besar.

Tabel 4.10 Rating alternatif percobaan ketiga

ID Alternatif	Nama Barang	Skor Alternatif	Peringkat
$C_1$	Cola	0,492	11
$C_2$	Fanta Orange	0,415	13
$C_3$	Lay's Potato Chips	0493	10
$C_4$	Nutella Chocolate Spread	0,310	15
$C_5$	Candy Pack	0,702	3
$C_6$	Chocapic Cereal	0,677	4
$C_7$	Whole Wheat Bread	0,657	6
$C_8$	Barilla Pasta	0,519	9
$C_9$	Fresh Milk	0,721	2
$C_{10}$	Cooking Oil	0,330	14
$C_{11}$	Sunlight Dish Soap	0,624	7
$C_{12}$	Dishwater Detergent	0,676	5
$C_{13}$	Pepsi Soda	0,803	1
$C_{14}$	Camembert Cheese	0,490	12
$C_{15}$	Whipping Cream	0,551	8

Pada Tabel 4.6 yang menampilkan hasil peringkat alternatif barang pada percobaan kedua, terlihat bahwa Pepsi Soda  $C_{13}$  menempati posisi teratas dengan skor sebesar 0,803, mengungguli 14 alternatif lainnya. Skor tinggi ini diperoleh berkat kombinasi nilai optimal dari empat kriteria yang dimiliki produk tersebut, menjadikannya prioritas utama untuk direstock guna menjaga ketersediaan barang yang memiliki performa terbaik secara keseluruhan. Di posisi kedua, Fresh Milk  $C_9$ , menyusul dengan skor 0,721, diikuti oleh Candy Pack  $C_5$ , dengan skor 0,702. Sebaliknya, Nutella Chocolate Spread  $C_4$ , menempati peringkat terbawah dengan skor SAW sebesar 0,310. Meskipun produk ini memiliki karakteristik tertentu, nilai skor yang rendah menunjukkan bahwa jika dibandingkan dengan alternatif lain, Nutella tidak terlalu mendesak untuk segera dilakukan restock. Sementara itu, Cooking Oil  $C_{10}$  berada di peringkat ke-14 dengan skor 0,330, yang berarti produk ini juga termasuk dalam kategori prioritas rendah. Hasil ini menunjukkan bahwa

metode SAW mampu menyusun prioritas restock secara objektif berdasarkan kombinasi multidimensi dari seluruh kriteria yang tersedia, bukan sekadar mempertimbangkan satu faktor saja seperti harga atau stok, melainkan keseluruhan variabel secara proporsional.



Gambar 4.35 Tampilan notifikasi kandidat percobaan ketiga

Pada gambar 4.35 ditampilkan antarmuka dalam game yang memperlihatkan notifikasi rekomendasi restock setelah percobaan ketiga selesai dilakukan. Notifikasi ini muncul secara otomatis di layar pemain, menampilkan daftar barang yang teridentifikasi sebagai kandidat restock berdasarkan hasil perhitungan metode SAW. Visual notifikasi didesain mencolok dengan warna untuk menandakan tingkat urgensi untuk restock segera. Fitur ini tidak hanya memandu pemain dalam pengambilan keputusan manajemen inventori, tetapi juga menambah elemen strategis dalam alur gameplay, di mana pemain harus merespons rekomendasi sistem secara cepat untuk menjaga kelancaran operasional toko.



Gambar 4.36 Tampilan kandidat barang didisplay percobaan ketiga

Pada gambar 4.36 ditampilkan bahwa pepsi soda menjadi barang yang direkomendasikan untuk direstock. Rekomendasi ini sejalan dengan hasil kalkulasi perhitungan menggunakan metode *SAW* pada tahap sebelumnya, di mana pepsi soda memperoleh nilai preferensi tertinggi dibandingkan alternatif barang lainnya. Hal ini menunjukkan konsistensi antara proses perhitungan prioritas dan implementasi notifikasi rekomendasi restock dalam antarmuka permainan.

#### 4.4 Pengujian Usability

Pengujian dalam penelitian ini bertujuan untuk mengevaluasi tingkat kenyamanan, kemudahan, dan efektivitas sistem simulasi operasional supermarket berbasis metode *Simple Additive Weighting (SAW)* dari perspektif pengguna. Pengujian ini melibatkan 34 orang responden, yang berdasarkan kriteria utama yaitu memiliki pengalaman memainkan game simulasi manajemen sebelumnya. Mayoritas responden berasal dari komunitas *game developer*. Responden dipilih

berdasarkan kriteria utama yaitu memiliki pengalaman memainkan game simulasi manajemen, agar feedback yang diberikan lebih relevan terhadap sistem yang diuji.

Prosedur pengujian dilakukan sesuai dengan rencana yang telah dijelaskan pada Bab 3.4, yaitu dengan meminta responden untuk terlebih dahulu mencoba sistem simulasi yang telah dikembangkan. Setelah proses uji coba, responden diminta mengisi formulir kuesioner *System Usability Scale (SUS)* yang terdiri dari 10 pernyataan, untuk memberikan penilaian mengenai tingkat usability dari sistem tersebut. Selain memberikan penilaian numerik, beberapa responden juga menyampaikan saran dan masukan terkait fitur dan pengembangan sistem simulasi ke depannya.

Setelah responden menyelesaikan sesi bermain simulasi, mereka diminta untuk mengisi kuesioner *System Usability Scale (SUS)* guna mengevaluasi tingkat usability dari sistem yang telah diuji. Kuesioner ini berisi 10 pernyataan yang berkaitan dengan pengalaman penggunaan sistem, meliputi aspek kemudahan penggunaan, kelengkapan fitur, hingga kenyamanan navigasi. Dari hasil pengisian kuesioner tersebut, diperoleh data skor usability dari masing-masing responden yang kemudian diolah sesuai metode perhitungan *SUS*. Adapun hasil rekapitulasi nilai yang diperoleh dapat dilihat pada tabel berikut.

Tabel 4.11 Hasil Kuesioner SUS

Responden	Pertanyaan									
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>	Q <sub>8</sub>	Q <sub>9</sub>	Q <sub>10</sub>
R <sub>1</sub>	3	2	4	4	4	2	4	2	4	4
R <sub>2</sub>	4	2	4	2	5	2	5	1	4	2
R <sub>3</sub>	5	1	5	2	4	1	5	2	5	1
R <sub>4</sub>	5	1	5	2	5	1	5	2	5	2
R <sub>5</sub>	4	1	4	2	4	1	4	1	5	1
R <sub>6</sub>	5	1	4	1	5	1	5	1	5	1
R <sub>7</sub>	4	2	5	2	4	2	4	2	4	2

Responden	Pertanyaan									
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>	Q <sub>8</sub>	Q <sub>9</sub>	Q <sub>10</sub>
R <sub>8</sub>	4	2	4	2	4	2	5	1	5	1
R <sub>9</sub>	5	1	4	1	4	1	5	1	4	1
R <sub>10</sub>	4	1	4	1	4	1	5	1	5	1
R <sub>11</sub>	5	1	5	1	4	1	5	1	5	1
R <sub>12</sub>	4	2	4	1	4	2	5	2	5	2
R <sub>13</sub>	4	2	5	2	5	1	4	2	4	2
R <sub>14</sub>	5	4	5	4	4	2	4	2	4	5
R <sub>15</sub>	5	2	5	4	5	2	4	1	4	3
R <sub>16</sub>	4	2	4	4	5	5	3	4	4	4
R <sub>17</sub>	5	2	4	4	5	5	4	2	4	4
R <sub>18</sub>	4	2	4	4	5	4	5	2	4	4
R <sub>19</sub>	5	2	4	3	5	4	5	2	5	3
R <sub>20</sub>	5	1	5	4	5	5	4	2	2	3
R <sub>21</sub>	4	1	5	3	4	4	5	2	4	4
R <sub>22</sub>	5	1	5	2	5	1	5	1	5	1
R <sub>23</sub>	5	2	4	4	5	3	5	2	5	4
R <sub>24</sub>	5	2	4	3	4	5	4	2	4	4
R <sub>25</sub>	5	1	5	2	5	2	5	1	5	2
R <sub>26</sub>	5	1	4	2	5	1	4	1	5	1
R <sub>27</sub>	5	1	4	1	5	1	4	2	5	1
R <sub>28</sub>	5	2	4	1	5	1	5	1	5	1
R <sub>29</sub>	5	2	5	2	4	1	5	2	5	2
R <sub>30</sub>	5	1	5	1	5	2	5	2	4	1
R <sub>31</sub>	4	2	5	1	5	2	5	1	5	1
R <sub>32</sub>	5	1	5	2	5	2	4	1	5	1
R <sub>33</sub>	5	2	4	2	5	1	5	1	5	1
R <sub>34</sub>	5	1	5	2	5	2	5	2	5	2

Pada tabel 4.11 Data hasil kuesioner yang telah dikumpulkan dari para responden kemudian diolah menggunakan metode perhitungan *System Usability Scale (SUS)*. Proses perhitungan ini mengikuti aturan skoring yang telah dijelaskan secara rinci pada Bab 3.4. Pada tahap awal, setiap jawaban responden pada 10 pernyataan *SUS* akan diberikan skor sesuai ketentuan, yaitu untuk pernyataan bernomor ganjil, nilai skala dikurangi 1, dan untuk pernyataan bernomor genap, nilai 5 dikurangi nilai skala yang dipilih responden. Setelah itu, skor masing-masing butir pernyataan dijumlahkan untuk memperoleh total nilai dari tiap responden.

Hasil total nilai ini kemudian dikalikan dengan faktor 2,5 untuk mendapatkan skor *SUS* akhir per responden dalam rentang 0 hingga 100.

Tabel 4.12 Skor total SUS

Responden	Pertanyaan										Total
	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>	Q <sub>8</sub>	Q <sub>9</sub>	Q <sub>10</sub>	
R <sub>1</sub>	3	2	4	4	4	2	4	2	4	4	25
R <sub>2</sub>	4	2	4	2	5	2	5	1	4	2	33
R <sub>3</sub>	5	1	5	2	4	1	5	2	5	1	37
R <sub>4</sub>	5	1	5	2	5	1	5	2	5	2	37
R <sub>5</sub>	4	1	4	2	4	1	4	1	5	1	35
R <sub>6</sub>	5	1	4	1	5	1	5	1	5	1	39
R <sub>7</sub>	4	2	5	2	4	2	4	2	4	2	28
R <sub>8</sub>	4	2	4	2	4	2	5	1	5	1	33
R <sub>9</sub>	5	1	4	1	4	1	5	1	4	1	37
R <sub>10</sub>	4	1	4	1	4	1	5	1	5	1	36
R <sub>11</sub>	5	1	5	1	4	1	5	1	5	1	39
R <sub>12</sub>	4	2	4	1	4	2	5	2	5	2	32
R <sub>13</sub>	4	2	5	2	5	1	4	2	4	2	32
R <sub>14</sub>	5	4	5	4	4	2	4	2	4	5	28
R <sub>15</sub>	5	2	5	4	5	2	4	1	4	3	30
R <sub>16</sub>	4	2	4	4	5	5	3	4	4	4	21
R <sub>17</sub>	5	2	4	4	5	5	4	2	4	4	24
R <sub>18</sub>	4	2	4	4	5	4	5	2	4	4	24
R <sub>19</sub>	5	2	4	3	5	4	5	2	5	3	29
R <sub>20</sub>	5	1	5	4	5	5	4	2	2	3	26
R <sub>21</sub>	4	1	5	3	4	4	5	2	4	4	28
R <sub>22</sub>	5	1	5	2	5	1	5	1	5	1	40
R <sub>23</sub>	5	2	4	4	5	3	5	2	5	4	30
R <sub>24</sub>	5	2	4	3	4	5	4	2	4	4	25
R <sub>25</sub>	5	1	5	2	5	2	5	1	5	2	37
R <sub>26</sub>	5	1	4	2	5	1	4	1	5	1	37
R <sub>27</sub>	5	1	4	1	5	1	4	2	5	1	37
R <sub>28</sub>	5	2	4	1	5	1	5	1	5	1	37
R <sub>29</sub>	5	2	5	2	4	1	5	2	5	2	35
R <sub>30</sub>	5	1	5	1	5	2	5	2	4	1	37
R <sub>31</sub>	4	2	5	1	5	2	5	1	5	1	37
R <sub>32</sub>	5	1	5	2	5	2	4	1	5	1	36
R <sub>33</sub>	5	2	4	2	5	1	5	1	5	1	37
R <sub>34</sub>	5	1	5	2	5	2	5	2	5	2	37

Pada tabel 4.12 Setelah seluruh nilai total dari masing-masing responden diperoleh, langkah selanjutnya adalah melakukan perhitungan skor *individual* untuk tiap responden. Perhitungan ini dilakukan dengan cara mengalikan total nilai hasil

skoring setiap responden dengan faktor pengali 2,5, sesuai dengan ketentuan perhitungan *SUS* yang telah dijelaskan pada bagian sebelumnya. Hasil dari perhitungan ini merupakan skor *SUS* akhir masing-masing responden dalam rentang nilai 0 hingga 100. Selanjutnya, seluruh skor *SUS* responden tersebut dijumlahkan dan dibagi dengan jumlah responden untuk memperoleh nilai rata-rata skor *SUS*. Nilai rata-rata ini akan digunakan untuk menilai tingkat *usability* sistem secara umum berdasarkan persepsi keseluruhan responden.

Tabel 4.13 Hasil skor *SUS*

<b>Responden</b>	<b>Skor <i>SUS</i></b>
<i>R</i> <sub>1</sub>	72.5
<i>R</i> <sub>2</sub>	62.5
<i>R</i> <sub>3</sub>	57.5
<i>R</i> <sub>4</sub>	82.5
<i>R</i> <sub>5</sub>	92.5
<i>R</i> <sub>6</sub>	92.5
<i>R</i> <sub>7</sub>	87.5
<i>R</i> <sub>8</sub>	97.5
<i>R</i> <sub>9</sub>	70.0
<i>R</i> <sub>10</sub>	82.5
<i>R</i> <sub>11</sub>	92.5
<i>R</i> <sub>12</sub>	90.0
<i>R</i> <sub>13</sub>	97.5
<i>R</i> <sub>14</sub>	80.0
<i>R</i> <sub>15</sub>	80.0
<i>R</i> <sub>16</sub>	70.0
<i>R</i> <sub>17</sub>	75.0
<i>R</i> <sub>18</sub>	52.5
<i>R</i> <sub>19</sub>	60.0
<i>R</i> <sub>20</sub>	60.0
<i>R</i> <sub>21</sub>	72.5
<i>R</i> <sub>22</sub>	65.0
<i>R</i> <sub>23</sub>	70.0
<i>R</i> <sub>24</sub>	100.0
<i>R</i> <sub>25</sub>	75.0
<i>R</i> <sub>26</sub>	62.5
<i>R</i> <sub>27</sub>	92.5
<i>R</i> <sub>28</sub>	92.5
<i>R</i> <sub>29</sub>	92.5
<i>R</i> <sub>30</sub>	92.5
<i>R</i> <sub>31</sub>	87.5
<i>R</i> <sub>32</sub>	92.5
<i>R</i> <sub>33</sub>	92.5
<i>R</i> <sub>34</sub>	90.0

<b>Responden</b>	<b>Skor SUS</b>
<b>Total Skor SUS</b>	2890
<b>Skor Rata-Rata</b>	80.28

Langkah selanjutnya adalah membandingkan nilai rata-rata skor yang telah diperoleh dengan kategori *interpretasi* skor yang berlaku. Perbandingan ini bertujuan untuk menentukan posisi hasil pengujian dalam kategori tingkat usability tertentu sesuai dengan ketentuan standar *SUS*. Adapun ketentuan kategori nilai tersebut dapat dilihat pada tabel interpretasi berikut, di mana setiap rentang nilai memiliki kategori masing-masing, mulai dari kategori "*poor*" hingga "*excellent*". Dengan demikian, skor rata-rata yang telah dihitung akan dipetakan ke dalam kategori tersebut untuk memberikan gambaran mengenai seberapa baik tingkat usability sistem yang telah diuji berdasarkan persepsi pengguna.

Tabel 4.14 Pedoman SUS

<i>SUS Score</i>	<i>Adjective Rating</i>
> 80.3	<i>Excellent</i>
68 – 80.3	<i>Good</i>
68	<i>Okay</i>
51 - 68	<i>Poor</i>
< 51	<i>Awful</i>

Pada gambar 4.14 Berdasarkan hasil perhitungan pada Tabel 4.13, diperoleh nilai rata-rata skor *SUS* sebesar 80,28. Nilai ini menempatkan sistem simulasi operasional supermarket dalam kategori (*good*) hingga "*excellent*" menurut standar interpretasi. Artinya, secara umum sistem ini dinilai layak dan memadai untuk digunakan oleh pengguna. Sebagian besar responden memberikan tanggapan positif terhadap aspek fungsionalitas sistem, kemudahan penggunaan antarmuka, serta kelengkapan fitur yang ditawarkan dalam game.

Namun demikian, hasil evaluasi juga mengungkapkan adanya beberapa catatan perbaikan, khususnya terkait aspek kemudahan navigasi menu serta pengelolaan stok barang. Beberapa responden menyarankan agar sistem navigasi dibuat lebih intuitif dan pengaturan stok barang dipermudah untuk meningkatkan kenyamanan dalam bermain. Oleh karena itu, meskipun sistem sudah memperoleh penilaian usability yang baik, masih terdapat ruang untuk penyempurnaan agar pengalaman pengguna dapat menjadi lebih optimal di versi pengembangan selanjutnya.

#### **4.5 Integrasi Sains dan Islam**

Integrasi antara nilai-nilai islam dan ilmu pengetahuan menjadi aspek penting dalam setiap bentuk pengembangan teknologi, termasuk dalam simulasi operasional supermarket. Penelitian ini tidak hanya menghadirkan sistem pengambilan keputusan berbasis data, tetapi juga mengintegrasikan prinsip-prinsip islam dalam pengelolaan ekonomi dan sosial. Pemain diajak untuk memahami hubungan mereka dengan Allah SWT, sesama manusia dan alam, sesuai prinsip muamalah dalam islam.

##### **4.5.1 Muamalah Mu'Allah**

Muamalah Ma'Allah merujuk pada hubungan vertikal antara manusia dan Allah SWT, yang mencakup seluruh aktivitas manusia yang dilakukan dengan kesadaran akan keberadaan dan perintah-Nya. Setiap tindakan manusia, termasuk dalam aspek sosial dan ekonomi, seharusnya dilandasi oleh niat untuk beribadah dan mencari keridhaan Allah. Hal ini menegaskan bahwa aktivitas muamalah tidak

terlepas dari nilai-nilai spiritual dan etika yang ditetapkan dalam ajaran Islam. Sebagaimana dijelaskan dalam Tafsir Ibnu Katsir, prinsip-prinsip dasar muamalah dalam Islam, seperti keadilan, kejujuran, dan amanah, merupakan manifestasi dari ketaatan kepada Allah dan menjadi landasan dalam menjalankan hubungan antar manusia. Prinsip-prinsip ini tidak hanya berfungsi sebagai pedoman moral, tetapi juga sebagai sarana untuk menjaga keseimbangan dan keharmonisan dalam masyarakat. Al-Qur'an menegaskan pentingnya hubungan ini dalam Q.S Al-Muthaffifin : 1 - 3.

وَيْلٌ لِّلْمُطَفِّفِينَ – الَّذِينَ إِذَا أَكْتَالُوا عَلَى النَّاسِ يَسْتَوْفُونَ – وَإِذَا كَالُواهُمْ أَوْ وَزَنُواهُمْ يُخْسِرُونَ

*“Kecelakaan besarlah bagi orang-orang yang curang (dalam menakar dan menimbang)! (yaitu) orang-orang yang apabila menerima takaran dari orang lain mereka minta dipenuhi, dan apabila mereka menakar atau menimbang untuk orang lain, mereka mengurangi.” (Q.S. Al-Muthaffifin: 1-3).*

Dalam Tafsir Ibnu Katsir dijelaskan bahwa ayat ini menjadi peringatan keras bagi para pedagang dan pelaku usaha agar tidak merugikan orang lain demi keuntungan pribadi. Ayat ini menunjukkan bahwa Islam sangat menekankan pentingnya kejujuran dalam transaksi ekonomi, termasuk dalam hal pengelolaan persediaan barang. Praktik penimbunan barang (*ihtikar*) dengan tujuan menaikkan harga secara tidak wajar, atau curang dalam distribusi barang, dilarang dalam ajaran Islam karena dapat merugikan masyarakat. Oleh karena itu, dalam pengelolaan stok barang, pelaku usaha harus memastikan ketersediaan barang secara wajar, menghindari penimbunan, dan menjaga harga tetap stabil demi kemaslahatan bersama, serta sebagai bentuk ketaatan dan tanggung jawab moral kepada Allah SWT (Idris Siregar, 2024).

#### 4.5.2 Muamalah Mu'annas dan Mu'alam

Dalam Islam, setiap aktivitas sosial dan pengelolaan lingkungan yang dilakukan oleh seorang Muslim harus berlandaskan prinsip keadilan, saling tolong-menolong, serta menghindari perilaku yang merugikan diri sendiri, orang lain, dan alam sekitar. Salah satu prinsip penting yang ditekankan dalam ajaran Islam adalah larangan terhadap pemborosan (*israf*) yang mencakup penggunaan harta secara berlebihan, ketidakhati-hatian dalam pengeluaran, serta pengabaian terhadap kepentingan bersama. Al-Qur'an menegaskan larangan ini dalam Surah Al-A'raf ayat 31 :

يُحِبُّ لَا يُبَيِّئِ أَدَمَ خُدُوا زِينَتَكُمْ عِنْدَ كُلِّ مَسْجِدٍ وَكُلُوا وَاشْرَبُوا وَلَا تُسْرِفُوا إِنَّهُ  
 ۳۱ الْمُسْرِفِينَ □

*“Wahai anak cucu adam, pakailah pakaianmu yang indah pada setiap (memasuki) masjid dan makan serta minumlah, tetapi janganlah berlebihan. Sesungguhnya dia tidak menyukai orang-orang yang berlebihan.” (Q.S.Al-A'raf : 31).*

Dimana memerintahkan manusia untuk memanfaatkan nikmat Allah tanpa berlebih-lebihan, karena Allah tidak menyukai orang-orang yang berbuat *israf*. Larangan ini tidak hanya berlaku dalam aspek konsumsi, tetapi juga dalam penggunaan waktu, tenaga, dan sumber daya alam. Dalam pengelolaan, prinsip ini mengajarkan pelaku usaha untuk senantiasa menjaga keseimbangan antara ketersediaan dan kebutuhan barang, menghindari praktik penimbunan, serta memastikan distribusi yang adil agar tidak menimbulkan ketimpangan sosial. Selain itu, Islam memandang alam sebagai amanah dari Allah SWT yang harus dijaga dan dimanfaatkan secara bijak. Manusia diberi peran sebagai khalifah di

bumi, yang memiliki tanggung jawab moral untuk melestarikan lingkungan dan mencegah kerusakan. Prinsip etika lingkungan dalam Islam menuntut agar setiap aktivitas ekonomi memperhatikan keberlanjutan sumber daya alam, serta memanfaatkan rezeki yang diberikan Allah tanpa berlebih-lebihan atau tabdzir. Sejalan dengan konsep fiqh ekologi berbasis maqashid syariah, pelestarian lingkungan diposisikan sebagai bagian dari upaya menjaga kemaslahatan umum.

Dalam tafsirnya, Syaikh Abdurrahman bin Nashir as-Sa'di menjelaskan bahwa pemborosan merupakan perbuatan yang sangat dibenci oleh Allah dan tergolong perilaku syaitan yang senantiasa mendorong manusia kepada kerusakan (M. Syarif Adi Pramana & Muhammad Saiful Khair, 2024). Hal ini didukung pula oleh penjelasan dalam Tafsir Ibnu Katsir yang menegaskan bahwa manusia diperintahkan untuk memanfaatkan nikmat rezeki secara seimbang tanpa berlebih-lebihan, baik dalam konsumsi, pengelolaan persediaan, maupun dalam memanfaatkan sumber daya (Rozi et al., 2022). Prinsip-prinsip tersebut relevan diterapkan dalam simulasi operasional supermarket sebagai sarana edukasi nilai-nilai Islam, sekaligus mendorong perilaku ekonomi yang adil, efisien, dan ramah lingkungan (Ali Mutakin & Waheeda binti H. Abdul Rahman, 2023).

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Penelitian ini telah berhasil membangun sebuah simulasi operasional supermarket berbasis game mobile yang menerapkan metode *Simple Additive Weighting (SAW)* untuk menentukan prioritas restock barang di supermarket. Sistem ini mampu mengolah berbagai kriteria penting seperti harga barang, jumlah stok, masa kadaluarsa, dan data permintaan untuk menghasilkan ranking prioritas barang yang perlu diadakan ulang.

Hasil implementasi menunjukkan bahwa metode *SAW* dapat diterapkan secara efektif dalam simulasi operasional supermarket, di mana pemain dapat melakukan pengambilan keputusan restock berdasarkan hasil perhitungan sistem. Selain itu, sistem ini juga dilengkapi dengan antarmuka yang interaktif dan sistem hari dalam simulasi untuk memberikan pengalaman simulasi manajemen supermarket yang realistis.

Pengujian sistem menggunakan *System Usability Scale (SUS)* kepada 34 responden yang merupakan pemain game simulasi menunjukkan hasil yang positif. Skor yang diperoleh termasuk dalam kategori baik, yang mengindikasikan bahwa sistem mudah dipahami dan digunakan oleh pemain.

## 5.2 Saran

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, berikut beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut:

- a. Menambahkan lebih banyak kriteria penilaian dalam proses perhitungan SAW, seperti tingkat keuntungan barang atau tren permintaan musiman, untuk memperkaya simulasi pengambilan keputusan.
- b. Mengembangkan fitur analisis laporan penjualan dan stok yang lebih lengkap agar pemain dapat memantau performa supermarket dalam periode tertentu dan mengevaluasi keputusan restock yang telah diambil.
- c. Meningkatkan tampilan antarmuka pengguna (*UI/UX*) agar lebih menarik dan intuitif, serta menyesuaikan dengan kebiasaan pemain game simulasi manajemen modern.
- d. Menambahkan sistem *event* atau kondisi acak (misalnya lonjakan permintaan atau diskon *distributor*) yang dapat mempengaruhi prioritas restock barang, sehingga memberikan tantangan tambahan bagi pemain dalam membuat keputusan.

Melakukan pengujian *usability* dengan jumlah responden yang lebih banyak dan beragam di masa mendatang untuk mendapatkan hasil evaluasi sistem yang lebih representatif.

## DAFTAR PUSTAKA

- Ahamed, S., Das, A., Md Tanjib, S., & Nahar Eity, M. Q. (2020). Study of an Application Development Environment Based on Unity Game Engine. *International Journal of Computer Science and Information Technology*, 12(1), 43–62. <https://doi.org/10.5121/ijcsit.2020.12103>
- Alberto, F. E. (2024). ROI Analysis of RFID-Based Intelligent Inventory Management Systems. *Frontiers in Management Science*, 3(3), 52–59. <https://doi.org/10.56397/fms.2024.06.07>
- Ali Mutakin, & Waheeda binti H. Abdul Rahman. (2023). Fiqh Ekologi; Upaya Merawat Lingkungan Hidup Berbasis Konsep Maqashid Syariah. *Syariah: Journal of Fiqh Studies*, 1(2), 107–126. <https://doi.org/10.61570/syariah.v1i2.31>
- Aprilia, Y. N., & Wahidin, A. J. (2022). Sistem Pendukung Keputusan Menentukan Optik Terbaik Dengan Metode Simple Additive Weighting. *Swabumi*, 10(1), 77–87. <https://doi.org/10.31294/swabumi.v10i1.12226>
- Arasa, R. M., & Achuora, J. O. (2020). Strategic Inventory Management Practices and the Performance of Supermarkets in Nairobi County, Kenya. *European Journal of Business and Management Research*, 5(2), 1–9. <https://doi.org/10.24018/ejbmr.2020.5.2.108>
- Asrianda, A., & Zulfadli, Z. (2022). Konsep Finite State Machine dan implementasinya pada Game. *Sisfo: Jurnal Ilmiah Sistem Informasi*, 6(1), 141. <https://doi.org/10.29103/sisfo.v6i1.8352>
- As-Sa'di, Abdurrahman bin Nashir. 2012. Taisir al-Karim ar-Rahman fi Tafsir Kalam al-Mannan (Tafsir as-Sa'di). Riyadh: Maktabah al-Ubaykan.
- Dwiawinta, W., & Rukmi, A. S. (2022). Pengembangan Game Jelajah Nusantara Untuk Meningkatkan Keterampilan Menulis Kalimat Sederhana Siswa Kelas Ii Sd. 1390–1400
- Fuaddah Syifaul. (2023). Dampak Perubahan Sosial dan Ekonomi Terhadap Pola Konsumsi Masyarakat Urban di Era Digital. *Open Science Framework*. <https://doi.org/10.31219/osf.io/8npaq>
- Harahap, M. A. K., Hardisal, H., Ahyuna, A., & Rahim, R. (2022). Leveraging the Decision Support System and Simple Additive Weighting Method for Optimal Retail Location Identification. *JINAV: Journal of Information and Visualization*, 3(2), 174–180. <https://doi.org/10.35877/454ri.jinav1485>
- Hussain, F., Hussain, A., Shakeel, H., Uddin, N., & Ghouri, T. L. (2020). Unity Game Development Engine: A Technical Survey. *University of Sindh Journal of Information and Communication Technology (USJICT)*, 4(2), 73–81. <http://sujo.usindh.edu.pk/index.php/USJICT/>

- Ionescu-Feleagă, L., Dragomir, V. D., Rîndașu, S. M., Stoica, O. C., Curea, Ștefania C., Bunea, M., & Barna, L. E. L. (2025). Business simulation games from the perspective of accounting and management professors: Implications for sustainability education in universities. *International Journal of Management Education*, 23(2). <https://doi.org/10.1016/j.ijme.2025.101147>
- Ibnu Katsir, Ismail bin Umar. 2000. Tafsir Ibnu Katsir. Jilid 1. Jakarta: Pustaka Imam Asy-Syafi'i
- Jha, H. K., More, H., & Panchal, H. (2025). *DEMAND FORECASTING AND INVENTORY OPTIMIZATION IN SUPPLY CHAIN MANAGEMENT*. 03, 6587–6595.
- Khairani, I., Hutagalung, J. E., & Dewi, M. (2024). Decision Support System for Recruitment of General Election Committee using The Simple Additive Weighting. *Edumatic: Jurnal Pendidikan Informatika*, 8(1), 212–221. <https://doi.org/10.29408/edumatic.v8i1.25690>
- Khoiroh, M., Asmungi, A., Surani, S., & Wirayuda, B. (2023). Pengambilan Keputusan Penentuan Kriteria Prioritas dalam Proses Seleksi Dosen Baru di Universitas XYZ dengan Metode AHP. *Matrik: Jurnal Manajemen Dan Teknik Industri Produksi*, 23(2), 195. <https://doi.org/10.30587/matrik.v23i2.4873>
- Lewis, J. R. (2020). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, 34(7), 577–590. <https://doi.org/10.1080/10447318.2018.1455307>
- M. Syarif Adi Pramana, & Muhammad Saiful Khair. (2024). Israf's Prohibition in QS. al-A'raf: 31 (Abdullah Saeed's Contextual Approach). *Al-Fahmu: Jurnal Ilmu Al-Qur'an Dan Tafsir*, 3(1), 108–124. <https://doi.org/10.58363/alfahmu.v3i1.191>
- Metode, P., Dalam, M. C., Pengelolaan, S., Alat, P., Kantor, T., & Simangunsong, A. (2023). Nomor 2. *Agustus*, 22, 280–289. <https://ojs.trigunadharma.ac.id/index.php/jis/index>
- Neuman-Donihu, E., Jarvis, M., & Zhu, Y. (2023). FastPoints: A State-of-the-Art Point Cloud Renderer for Unity. *IS and T International Symposium on Electronic Imaging Science and Technology*, 35(1). <https://doi.org/10.2352/EI.2023.35.1.VDA-394>
- Normah, Rifai, B., Vambudi, S., & Maulana, R. (2022). Analisa Sentimen Perkembangan Vtuber Dengan Metode Support Vector Machine Berbasis SMOTE. *Jurnal Teknik Komputer AMIK BSI*, 8(2), 174–180. <https://doi.org/10.31294/jtk.v4i2>
- Nuna, A. J., Jangga, A. U., & Setiawi, A. P. (2024). Penerapan Metode SAW dalam Sistem Pendukung Keputusan Seleksi Siswa Baru pada SMKS Bakti Luhur Tambolaka. *Jurnal Ilmu Komputer Dan Bisnis*, 15(1), 169–175.

<https://doi.org/10.47927/jikb.v15i1.728>

- Prakoso, I., Sofiana, A., Nurmalawati, S., Triyanto, R., Rendra, A. R., & Rosyid, A. A. (2023). Simulasi Antrian dalam Optimalisasi Layanan di Supermarket Rita Pasaraya. *Dinamika Rekayasa*, 19(1), 45. <https://doi.org/10.20884/1.dr.2023.19.1.602>
- Pratama, A., & Suhendi, H. (2022). Aplikasi Game Simulasi Operasional Supermarket dengan CreateJS di PT Harja Gunatama Lestari. *EProsiding Teknik Informatika (PROTEKTIF)*, 3(1), 1–10.
- Riki Sulistio, Ahmad Fahreza Nasution, Muhammad Tawaf Akbar, Rima Tamara Aldisa, & Bister Purba. (2024). Implementation of the Simple Additive Weighting (SAW) Method in Selection of Students Recipients of Single Tuition Fee Assistance. *Journal of Computing and Informatics Research*, 3(2), 182–187. <https://doi.org/10.47065/comforch.v3i2.1202>
- Rizkiyah, K., Nurmayanti, L., Macdhy, R. D. N., & Yusuf, A. (2021). PENGARUH DIGITAL PAYMENT TERHADAP PERILAKU KONSUMEN Pengguna Platform Digital Payment OVO. *Jurnal Ilmiah Manajemen*, 16(1), 107–126.
- Rozi, F., Hamidah, T., & Arfan, A. (2022). Konsep Maqasid Syari'ah Perspektif Pemikiran Al-Juwaini dan Al-Ghazali. *Iqtisodina: Jurnal Ekonomi Syariah Dan Hukum Islam*, 5(1), 53–67.
- Saefudin, M., & Sudjiran, S. (2023). Penerapan Perangkat Lunak Unity Dalam Pengembangan Aplikasi Game Dua Dimensi Berbasis Android. *Jurnal Ilmiah SIKOMTEK*, 13(1), 9–16. <https://sikomtek.jakstik.ac.id/index.php/jurnalsikomtek/article/view/28>
- Safitri, M., Sahay, A. S., & Lestari, A. (2021). Implementasi Metode Simple Additive Weighting Dalam Sistem Pendukung Keputusan Penerima Bantuan Program Sembako. *Journal of Information Technology and Computer Science*, 1(1), 87–96. <https://doi.org/10.47111/jointecom.v1i1.2959>
- Saputra, A. R. (2020). Simulasi Sistem Point of Sale menggunakan Radio Frequency Identification pada Pasar Swalayan. *TEMATIKA: Jurnal Penelitian Teknik Informatika dan Sistem Informasi*, 11-17. <https://doi.org/10.56963/tematika.vi.272>
- Saputra, M. I. H., & Nugraha, N. (2020). Sistem Pendukung Keputusan Dengan Metode Analytical Hierarchy Process (Ahp) (Studi Kasus: Penentuan Internet Service Provider Di Lingkungan Jaringan Rumah). *Jurnal Ilmiah Teknologi Dan Rekayasa*, 25(3), 199–212. <https://doi.org/10.35760/tr.2020.v25i3.3422>
- Sasanuma, K., Hibiki, A., & Sexton, T. (2022). An opaque selling scheme to reduce shortage and wastage in perishable inventory systems. *Operations Research Perspectives*, 9, 1–35. <https://doi.org/10.1016/j.orp.2021.100220>
- Selina, Sunoto, A., & Hendrawan. (2022). Perancangan Sistem Informasi Persediaan Barang Pada Minimarket Richard Mandala Jambi. *Jurnal*

- Manajemen Teknologi Dan Sistem Informasi (JMS)*, 2(2), 275–284.  
<https://doi.org/10.33998/jms.2022.2.2.124>
- Sridhar, P., Vishnu, C. R., & Sridharan, R. (2021). Simulation of inventory management systems in retail stores: A case study. *Materials Today: Proceedings*, 47, 5130–5134. <https://doi.org/10.1016/j.matpr.2021.05.314>
- Taherdoost, H. (2023). Analysis of Simple Additive Weighting Method (SAW) as a MultiAttribute Decision-Making Technique: A Step-by-Step Guide. *Journal of Management Science & Engineering Research*, 6(1), 21–24. <https://doi.org/10.30564/jmsr.v6i1.5400>
- Terttiaavini, T. (2024). A Hybrid Approach Using K-Means Clustering and the SAW Method for Evaluating and Determining the Priority of SMEs in Palembang City. *Journal of Intelligent System and Computation*, 6(1), 46–53. <https://doi.org/10.52985/insyst.v6i1.392>
- Ullmann, G. C., Guéhéneuc, Y. G., Petrillo, F., Anquetil, N., & Politowski, C. (2025). SyDRA: An approach to understand game engine architecture. *Entertainment Computing*, 52(2024), 100832. <https://doi.org/10.1016/j.entcom.2024.100832>
- Vafaei, N., Ribeiro, R. A., & Camarinha-Matos, L. M. (2021). Assessing Normalization Techniques for Simple Additive Weighting Method. *Procedia Computer Science*, 199, 1229–1236. <https://doi.org/10.1016/j.procs.2022.01.156>
- Verma, K., Kourav, S., Jangid, M., Sahu, U., & Shivhare, N. (2023). Research on Finite State Machine and Its Real Life Time Applications. *Journal of Information Technology and Sciences*, 9(3), 10–20.
- Wijana, M., Gumelar, R. C., Supriatman, R. D., & Muhyidin, Y. (2024). Aplikasi Sistem Pendukung Keputusan Menentukan Siswa Terbaik Menggunakan Metode SAW. *INTERNAL (Information System Journal)*, 7(1), 18–29. <https://doi.org/10.32627/internal.v7i1.973>
- Wijanarko, H., P, R. G. K., & Puri, E. R. (2024). *Identification of Stakeholder Involvement in Decision-Making Needs ( Case Study: Sanur Port Construction Project )*. 107–115.