

**SISTEM REKOMENDASI *COLLABORATIVE FILTERING* MULTI  
CRITERIA UNTUK REKOMENDASI SKENARIO *TOUR*  
PADA GAME 3D PENGENALAN KAMPUS**

**SKRIPSI**

Oleh:  
**RASHAD FATHIN KURNIAWAN**  
NIM. 19650086



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

**SISTEM REKOMENDASI *COLLABORATIVE FILTERING* MULTI  
CRITERIA UNTUK REKOMENDASI SKENARIO TOUR  
PADA GAME 3D PENGENALAN KAMPUS**

**SKRIPSI**

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

Oleh:  
**RASHAD FATHIN KURNIAWAN**  
NIM. 19650086

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

**HALAMAN PERSETUJUAN**

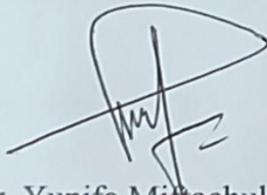
**SISTEM REKOMENDASI *COLLABORATIVE FILTERING* MULTI  
CRITERIA UNTUK REKOMENDASI SKENARIO *TOUR*  
PADA GAME 3D PENGENALAN KAMPUS**

**SKRIPSI**

**Oleh:**  
**RASHAD FATHIN KURNIAWAN**  
**NIM. 19650086**

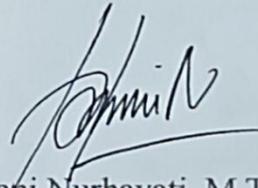
Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 22 Maret 2024

Pembimbing I,



Dr. Yunifa Miftachul Arif, M. T  
NIP. 19830616 201101 1 004

Pembimbing II,



Hani Nurhayati, M. T  
NIP. 19780625 200801 2 006

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



Rashad Fathin Kurniawan, M.MT, IPM  
NIP. 19771020 200912 1 001

**HALAMAN PENGESAHAN**

**SISTEM REKOMENDASI *COLLABORATIVE FILTERING* MULTI  
CRITERIA UNTUK REKOMENDASI SKENARIO *TOUR*  
PADA GAME 3D PENGENALAN KAMPUS**

**SKRIPSI**

**Oleh:**  
**RASHAD FATHIN KURNIAWAN**  
**NIM. 19650086**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 22 Maret 2024

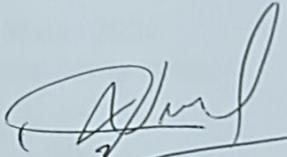
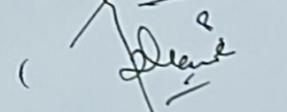
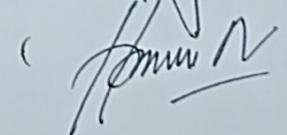
**Susunan Dewan Penguji**

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

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

Anggota Penguji II : Dr. Yunifa Miftachul Arif, M. T  
NIP. 19830616 201101 1 004

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

(  )  
(  )  
(  )  
(  )

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



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

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Rashad Fathin Kurniawan  
NIM : 19650086  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Sistem Rekomendasi *Collaborative Filtering Multi Criteria* Untuk Rekomendasi Skenario *Tour* Pada Game 3D Pengenalan Kampus

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, 22 Maret 2024  
Yang membuat pernyataan,



Rashad Fathin Kurniawan  
NIM. 19650086

MOTTO

～ 能ある鷹は爪を隠す ～

“A wise man keeps some of his talents in reserve”

## HALAMAN PERSEMBAHAN

*Alhamdulillah rabbil 'alamiin,*

Puji syukur kehadiran Allah SWT, Shalawat serta salam bagi Rasul-Nya.

Skripsi ini saya persembahkan kepada keluarga yang sangat dicintai, Bapak Ahadi Kurniawan, DAPE, MScPH yang berjasa menghidupkan, mendidik dan selalu memberi semangat kepada penulis untuk segera menyelesaikan tanggung jawabnya, Ibu Dessy Ferina, SE yang selalu mendoakan sekaligus mengingatkan penulis untuk beribadah dan berbuat kebaikan, Abang Mohammad Firenza Kurniawan yang selalu membantu penulis ketika sedang berada dalam kesulitan dan Adik Keiko Faiha Zhafira yang selalu menjadi menjadi teman dekat yang sama-sama merantau jauh dari rumah.

Dosen pembimbing penulis, Bapak Dr. Yunifa Miftachul Arif, M. T dan Ibu Hani Nurhayati, M.T, yang telah dengan sabar memberikan bimbingan, memberi masukan dan secara keseluruhan membantu dalam penyelesaian penelitian. Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah membimbing, mengajari, dan memberikan ilmu yang sangat bermanfaat kepada penulis dan berbagai pengalaman lainnya.

Juga teman-teman dari jauh dan dekat yang selalu *available* menjadi penyemangat penulis, juga membantu menyelesaikan dan memberi motivasi mengerjakan skripsi.

## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

*Alhamdulillah rabbil 'alamiin*, puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis diberikan kemudahan dan keberkahan dalam menyelesaikan skripsi ini yang berjudul “**Sistem Rekomendasi Collaborative Filtering Multi Criteria Untuk Rekomendasi Tour Pada Game 3D Pengenalan Kampus**”. Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW., atas syafaat-Nya yang telah menuntun umat manusia menuju jalan yang baik serta risalah islam yang penuh dengan ilmu pengetahuan yang membawa kita dari zaman kebodohan menuju zaman kebenaran yakni Islam.

Penyusunan skripsi ini bertujuan untuk memenuhi salah satu syarat kelulusan sebagai Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Negeri Maulana Malik Ibrahim Malang. Penulis menyadari bahwa penyusunan skripsi ini tidaklah mudah dan keberhasilan penulisan skripsi ini tak luput dari faktor dukungan, bimbingan, motivasi, semangat, serta doa dari banyak pihak. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Zainuddin, M.A., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Maulana Malik Ibrahim Malang.

3. Dr. Fachrul Kurniawan, M.MT, IPM, selaku Ketua Program Studi Universitas Maulana Malik Ibrahim Malang.
4. Dr. Yunifa Miftachul Arif, M. T selaku Dosen Pembimbing I yang dengan sabar membimbing penulis, memberi masukan dan arahan sehingga penulis tidak hanya mampu menyelesaikan pengerjaan skripsi namun juga mengambil banyak hikmah dan pelajaran.
5. Ibu Hani Nurhayati, M.T, selaku Dosen Pembimbing II yang juga bersedia meluangkan waktunya dalam memberikan bimbingan, arahan serta masukan kepada penulis dalam menyelesaikan skripsi ini.
6. Segenap civitas akademika Fakultas Sains dan Teknologi Universitas Islam Maulana Malik Ibrahim Malang terutama seluruh dosen, terima kasih atas bimbingan dan segala ilmu yang telah diberikan.
7. Orang tua dan keluarga penulis yang telah memberikan doa, motivasi dan dukungan sehingga penulis dapat menyelesaikan skripsi.
8. Teman-teman Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang Angkatan 2019 yang telah memberikan dukungan kepada penulis dalam menyelesaikan penelitian skripsi ini.
9. Juga abang Sultan Rambe, adik Keiko Faiha Zhafira dan teman Munirul Huda yang sudah menarik penulis dari tempat gelap dan memberi dorongan terakhir.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan dan jauh dari kata sempurna. Maka dari itu penulis selalu membuka kesempatan seluas-luasnya untuk setiap saran dan kritik yang membangun,

mengembangkan ilmu pengetahuan, serta melaksanakan tugas sebagai hamba Allah yang berkomitmen.

*Wassalamu'alaikum Wr. Wb*

Malang, 22 Maret 2024

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>xiv</b>
<b>ABSTRAK</b> .....	<b>xv</b>
<b>ABSTRACT</b> .....	<b>xvi</b>
<b>مستخلص البحث</b> .....	<b>xvii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	5
1.5 Batasan Penelitian .....	5
<b>BAB II STUDI PUSTAKA</b> .....	<b>6</b>
2.1 Penelitian Terkait .....	6
2.2 Landasan Teori.....	9
2.2.1 Simulasi .....	9
2.2.2 <i>Virtual reality</i> .....	10
2.2.3 <i>Path</i> .....	11
2.2.4 Sistem Rekomendasi.....	12
2.2.5 <i>Collaborative Filtering</i> .....	14
2.2.6 <i>Cosine similarity</i> .....	15
2.2.7 <i>Weighted sum</i> .....	16
2.2.8 Evaluasi Sistem.....	17
<b>BAB III METODE PENELITIAN</b> .....	<b>20</b>
3.1 Desain Penelitian.....	20
3.2 Desain Game .....	22
3.3 Desain Sistem.....	31
3.3.1 <i>Data Rating</i> .....	32
3.3.2 Perhitungan <i>Similarity</i> .....	33
3.3.3 Perhitungan Prediksi .....	34
3.3.4 Rencana Pengujian.....	35
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>38</b>
4.1 Implementasi .....	38
4.1.1 Implementasi Perangkat.....	38
4.1.2 Implementasi User Interface .....	39
4.1.3 Implementasi <i>Cosine similarity</i> .....	43
4.1.4 Implementasi <i>Weighted sum</i> .....	48

4.2 Implementasi Sistem .....	50
4.2.1 Proses Implementasi Pada Game .....	50
4.2.2 Pengujian Sistem Rekomendasi .....	54
4.3 Integrasi Islam .....	58
<b>BAB V PENUTUP.....</b>	<b>62</b>
5.1 Kesimpulan .....	62
5.2 Saran.....	63
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 3.1 Alur Penelitian .....	20
Gambar 3.2 Alur Game.....	22
Gambar 3.3 <i>Flowchart</i> Jalan Level Pada Game.....	23
Gambar 3.4 <i>Mindmap</i> Desain Game.....	23
Gambar 3.5 <i>Gameplay</i> .....	24
Gambar 3.6 FSM <i>Window</i> Pertanyaan <i>Gameobject</i> Pada Titik <i>Quest</i> .....	25
Gambar 3.7 Empat <i>Path</i> Yang Dapat Dilalui Untuk Pergi Ke Gedung Saintek.....	27
Gambar 3.8 <i>Path</i> 1 .....	28
Gambar 3.9 <i>Path</i> 2 .....	29
Gambar 3.10 <i>Path</i> 3 .....	30
Gambar 3.11 <i>Path</i> 4 .....	30
Gambar 3.12 <i>Flowchart</i> Jalan Sistem.....	31
Gambar 4.1 Tampilan <i>Main Menu</i> .....	39
Gambar 4.2 Tampilan <i>Loading Screen</i> .....	40
Gambar 4.3 <i>Window</i> Rekomendasi <i>Path</i> Yang Diberikan Kepada <i>Player</i> .....	40
Gambar 4.4 Tampilan Jika <i>Path</i> 4 Dipilih Dalam <i>Window</i> Rekomendasi.....	41
Gambar 4.5 Tampilan Jika <i>Path</i> 1 Dipilih Dalam <i>Window</i> Rekomendasi.....	41
Gambar 4.6 <i>Game Object</i> Yang Perlu Diinteraksikan Oleh <i>Player</i> .....	42
Gambar 4.7 Pertanyaan Quiz .....	42
Gambar 4.8 <i>Player</i> Menyelesaikan Quiz Dengan Nilai Lebih Atau Sama Dengan 80.....	43
Gambar 4.9 <i>Pseudo Code</i> Untuk Mengambil Data Dari Excel.....	44
Gambar 4.10 Data Excel yang Telah Dikonversi Menjadi <i>Scriptable Object</i> .....	45
Gambar 4.11 <i>Pseudo Code</i> Perhitungan <i>Cosine Similarity</i> Antara 2 Vektor.....	46
Gambar 4.12 <i>Pseudo Code</i> Perhitungan <i>Cosine Similarity</i> Dengan Seluruh Data .....	47
Gambar 4.13 Perhitungan <i>Similarity</i> Antara <i>Player</i> Dengan Setiap <i>User</i> .....	48
Gambar 4.14 <i>Pseudo Code</i> Perhitungan Nilai Prediksi Menggunakan <i>Weighted Sum</i> ....	49
Gambar 4.15 Pemberian Nilai Pada <i>Path</i> Menggunakan <i>Weighted Sum</i> .....	49
Gambar 4.16 Data Yang Telah Diurutkan Berdasarkan <i>Similaritynya</i> .....	51
Gambar 4.17 <i>Log</i> Yang Menunjukkan Nilai <i>Rating</i> Baru <i>Player</i> Terhadap Tiap <i>Path</i> ....	52
Gambar 4.18 Tampilan Ketika Rekomendasi <i>Path</i> Dipilih .....	52
Gambar 4.19 Penempatan <i>Game Object Waypoint</i> Dalam Unity .....	53
Gambar 4.20 <i>Pseudo Code</i> Untuk Menyambungkan <i>Trail</i> Dari <i>Player</i> Ke Titik Target .	53
Gambar 4.21 Pengujian Sistem Rekomendasi Berdasarkan 10, 20 Dan 40 Data Uji.....	58

## DAFTAR TABEL

Tabel 3.1 <i>Storyboard</i> Contoh Jalan Level .....	26
Tabel 3.2 Contoh Tabel Data <i>Rating</i> .....	33
Tabel 3.3 Contoh Perhitungan <i>Similarity</i> Antar <i>User</i> .....	34
Tabel 3.4 Contoh Perbandingan Nilai .....	36
Tabel 3.5 Contoh <i>Confusion Matrix</i> .....	36
Tabel 4.1 Data Mentah Yang Disimpan Dalam Microsoft Excel .....	45
Tabel 4.2 Pengujian Berdasarkan 10 Data Acuan.....	54
Tabel 4.3 <i>Confusion Matrix</i> .....	56
Tabel 4.4 Tabel Perhitungan <i>Confusion Matrix</i> Berdasarkan 10 Baris Data.....	56
Tabel 4.5 Tabel Perhitungan <i>Confusion Matrix</i> Berdasarkan 20 Baris Data.....	57
Tabel 4.6 Tabel Perhitungan <i>Confusion Matrix</i> Berdasarkan 40 Baris Data.....	57

## ABSTRAK

Kurniawan, Rashad Fathin. 2024. **Sistem Rekomendasi Collaborative Filtering Multi Criteria Untuk Rekomendasi Skenario Tour Pada Game 3D Pengenalan Kampus**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Yunifa Miftachul Arif, M. T (II) Hani Nurhayati, M.T.

**Kata kunci:** *Sistem Rekomendasi, Simulasi, Jalur*

Berbanding lurus dengan perkembangan teknologi yang terus berjalan, salah satu dari cara untuk mempelajari sesuatu tersebut adalah dengan game, khususnya game simulasi. Game merupakan kegiatan yang biasanya bertujuan sebagai media hiburan. Seiring dengan perkembangan zaman, game saat ini juga dikembangkan sebagai media pembelajaran. Penelitian ini dilakukan bertujuan untuk membuat game 3D untuk pengenalan kampus UIN Malang yang mengimplementasikan sistem rekomendasi *collaborative filtering multicriteria* untuk pemilihan jalur yang perlu digunakan untuk mencapai lokasi yang dituju. Pendekatan yang digunakan dalam sistem rekomendasi *collaborative filtering multicriteria* tersebut adalah pendekatan *cosine similarity* untuk mencari kemiripan antar *user* dan *weighted sum* untuk pemberian nilai *rating*-nya. Pada pengujian sistem, dengan menggunakan *confusion matrix* dihasilkan nilai akurasi, presisi, *recall* dan *F-score* yang berbeda-beda yang bergantung kepada jumlah data ujinya. Dengan menggunakan 10, 20, dan 40 data uji, diperoleh performa terbaik dengan menggunakan 40 data uji dengan nilai akurasi sebesar 90%, nilai presisi sebesar 87,5%, nilai *recall* sebesar 87,5% dan nilai *f-score* sebesar 87,5%.

## ABSTRACT

Kurniawan, Rashad Fathin. 2024. *Collaborative Filtering Multi Criteria Recommender System For Tour Recommendation In Campus Introduction 3D Game*. Undergraduate Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Yunifa Miftachul Arif, M. T (II) Hani Nurhayati, M.T.

**Keywords:** *Recommender System, Simulation, Path*

In direct proportion to ongoing technological developments, one way to learn something is through games, especially simulation games. Games are activities that are usually intended as a medium of entertainment. Along with the times, games are now also being developed as learning media. This research was carried out with the aim of creating a 3D game for introducing the UIN Malang campus which implements a collaborative filtering multicriteria recommendation system for selecting the path that needs to be used to reach the destination location. The approach used in the multicriteria collaborative filtering recommendation system is the cosine similarity approach to find similarities between users and the weighted sum approach to provide the rating value. In system testing, using the confusion matrix, different accuracy, precision, recall and F-score values are produced depending on the amount of test data. Using 10, 20, and 40 test data, the best performance was obtained using 40 test data with an accuracy value of 90%, a precision value of 87.5%, a recall value of 87.5% and an f-score value of 87.5%.

## مستخلص البحث

كورنيوان، رشاد فتحين. ٢٠٢٤. نظام التصفية التعاوني للتوصية متعدد المعايير لتوصيات سيناريو الجولة في ألعاب مقدمة الحرم الجامعي ثلاثية الأبعاد. أطروحة. برنامج دراسة الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرف: (١) د. يونيفة مفتاح عارف، م.ت (٢) هاني نورحياتي، م.ت  
الكلمات المفتاحية: نظام التوصية، المحاكاة، المسار

بما يتناسب بشكل مباشر مع التطورات التكنولوجية المستمرة، فإن إحدى طرق تعلم شيء ما هي من خلال الألعاب، وخاصة ألعاب المحاكاة. الألعاب هي أنشطة تهدف عادةً إلى أن تكون وسيلة للترفيه. ومع مرور الوقت، يتم أيضًا تطوير الألعاب كوسيلة تعليمية. تم إجراء هذا البحث بهدف إنشاء لعبة ثلاثية الأبعاد للتعريف بالحرم الجامعي للجامعة الإسلامية الحكومية في مالانج والذي يطبق نظام ترشيح تعاوني متعدد المعايير للتوصية لاختيار المسار الذي يجب استخدامه للوصول إلى موقع الوجهة. النهج المستخدم في نظام توصيات التصفية التعاونية متعدد المعايير هو نهج تشابه جيب التمام للعثور على أوجه التشابه بين المستخدمين والمجموع المرجح لتوفير قيمة التصنيف. في اختبار النظام، يؤدي استخدام مصفوفة الارتباك إلى إنتاج قيم مختلفة للدقة والدقة والاستدعاء ودرجة  $f$  والتي تعتمد على كمية بيانات الاختبار. باستخدام ١٠ و ٢٠ و ٤٠ بيانات اختبار، تم الحصول على أفضل أداء باستخدام ٤٠ بيانات اختبار بقيمة دقة ٩٠٪، بقيمة دقة ٨٧,٥٪، بقيمة استدعاء ٨٧,٥٪، بقيمة دقة  $f$  ٨٧,٥٪.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Belajar merupakan salah satu aspek hidup yang sangat penting bagi umat manusia, khususnya bagi umat muslim. Dengan adanya ilmu, manusia dapat mengetahui apa yang sebelumnya diketahui dan dapat menjadi orang yang lebih bijaksana. Pada surah an-Nahl ayat ke-78 menceritakan tentang potensi pada diri manusia yang harus digunakan dalam kegiatan belajar dan pembelajaran :

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَرَ وَالْأَفْئِدَةَ ۚ لَعَلَّكُمْ تَشْكُرُونَ

*“Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatupun, dan dia memberi kamu pendengaran, penglihatan dan hati, agar kamu bersyukur.” (QS. An-Nahl: 78)*

Ayat di atas menunjukkan bahwa ada tiga potensi yang terlibat dalam proses pembelajaran: *as-Sam'u* (mendengar), *al-Bashar* (melihat), dan *Fu'ad* (qalbu) (Wakka, 2020). Ada banyak cara bagi seseorang untuk mempelajari sesuatu. Berbanding lurus dengan perkembangan teknologi yang terus berjalan, salah satu dari cara untuk mempelajari sesuatu tersebut adalah dengan game, khususnya game simulasi. Game merupakan kegiatan yang biasanya bertujuan sebagai media hiburan. Namun, seiring berkembangnya zaman game saat ini juga dikembangkan sebagai media pembelajaran (Anggraini *et al.*, 2021). Simulasi adalah metode pelatihan yang memperagakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya, yang dimana dalam konteks ini penjelajahan lingkungan kampus akan divirtualisasikan. Dengan adanya simulasi *virtual reality*,

pengguna akan merasa seperti berada di lingkungan yang asli (Mambu *et al.*, 2019). Dengan adanya pembelajaran berbentuk simulasi game, pengguna diharapkan untuk dapat mempelajari sesuatu dengan lebih menarik sekaligus untuk menghindari resiko yang mungkin didapatkan dari lingkungan asli.

*Virtual reality* (VR) adalah teknologi yang memungkinkan penggunanya untuk menyelam ke dalam dunia (*reality*) buatan (*virtual*), baik dunia buatan yang alamnya sepenuhnya imajiner ataupun pembuatan ulang dari apa yang ada di dunia nyata (Elmqaddem, 2019). Dengan adanya VR, pengguna akan merasa seolah menyatu dengan lingkungan buatan tersebut dan dapat berinteraksi dengan objek-objek yang terdapat di sana (Saurik *et al.*, 2019). *Virtual reality* biasanya diaplikasikan kepada beberapa hal, salah satunya adalah untuk hiburan dan pendidikan. *Hardware* penting untuk menjalankan aplikasi VR antara lain adalah VR headset dan *motion controller*. Kedua hal tersebut diperlukan agar pengguna dapat mensimulasikan kehadiran fisiknya di dunia virtual tersebut, seperti melihat sekitar, bergerak dan berinteraksi dengan *item* yang ada.

Sistem rekomendasi merupakan sistem yang bertujuan untuk memperkirakan informasi yang menarik bagi pengguna dan juga membantu pengguna dalam menentukan pilihannya (Nugroho & Ismu Rahayu, 2020). Sistem rekomendasi adalah subkelas dari sistem penyaringan informasi yang memberikan saran untuk item yang paling relevan bagi pengguna tertentu. Sistem rekomendasi akan memberikan rekomendasi secara personal kepada setiap pengguna, bukan sekadar memberikan daftar item yang paling diminati dan rekomendasi akan diberikan kepada pengguna terkait item-item yang mungkin sesuai dengan ketertarikan

pengguna (Mondi *et al.*, 2019). Terdapat dua pendekatan utama dalam mengembangkan sistem rekomendasi yaitu *content-based filtering* dan *collaborative filtering*.

*Collaborative filtering* adalah suatu konsep dimana data dari pengguna lain yang ada digunakan untuk memprediksi item yang mungkin disukai / diminati oleh pengguna lainnya. Pendekatan ini merekomendasikan item untuk menargetkan pengguna berdasarkan apa yang sebelumnya disukai pengguna yang serupa (Haruna *et al.*, 2017). Hasil rekomendasi yang diberikan dengan menggunakan pendekatan *collaborative filtering* sangat bergantung pada data penilaian dari *neighbor* (pengguna lain) terhadap suatu *item* (Wijaya & Alfian, 2018). Pendekatan *Collaborative filtering* dalam sistem rekomendasi akan merekomendasikan pilihan berdasarkan dari gerombolan *rating user* yang ada terhadap *item* tertentu. Terdapat 2 tipe utama dari *collaborative filtering*, yaitu *item-based* dan *user-based* (Rokhim & Saikhu, 2016).

*Collaborative Filtering multicriteria* adalah sistem rekomendasi multi-kriteria yang menggunakan pendekatan *collaborative filtering*. *Collaborative Filtering multicriteria* memprediksi *rating* keseluruhan untuk suatu *item* berdasarkan *rating* sebelumnya mengenai kedua *item* secara keseluruhan dan kriteria individu, serta merekomendasikan kepada pengguna *item* dengan skor keseluruhan terbaik. Dengan demikian, algoritma untuk sistem rekomendasi *collaborative filtering multicriteria* dapat dikembangkan dari sistem rekomendasi kriteria tunggal (Nilashi *et al.*, 2016). Contohnya dalam sistem rekomendasi film, jika seorang pengguna menyukai sebuah film, kriteria yang disukai dari satu

pengguna dengan pengguna lain tidak selalu sama (pengguna 1 mungkin menyukai aktingnya dan pengguna 2 mungkin menyukai plotnya). Jadi, Sistem Rekomendasi *multicriteria* yang memanfaatkan *rating* multi-kriteria untuk mengevaluasi atribut *item* yang berbeda dapat meningkatkan akurasi rekomendasi (Nassar *et al.*, 2020).

Berdasarkan latar belakang di atas, maka penelitian ini dilakukan bertujuan untuk membuat game *virtual reality* untuk pengenalan kampus UIN Malang yang mengimplementasikan sistem rekomendasi *collaborative filtering multicriteria* untuk pemilihan jalur yang perlu digunakan untuk mencapai lokasi yang dituju. Dengan game *virtual reality* yang dibuat, mahasiswa baru ataupun warga luar yang belum mengenal area kampus UIN Malang dapat mengetahui, menghafal dan mengalami area kampus UIN Malang secara langsung dengan *immersive*. Hal tersebut dilakukan dengan memberikan *player* yang nantinya akan menjalankan game melakukan beberapa misi yang mengharuskan *player* untuk mengelilingi kampus.

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang masalah di atas, maka ditetapkan bahwa masalah dari penelitian adalah seberapa akurat sistem rekomendasi *collaborative filtering multicriteria* dalam memberikan rekomendasi jalur pada game *virtual reality* pengenalan kampus UIN Malang.

## **1.3 Tujuan Penelitian**

Dari rumusan masalah tersebut, ditetapkan bahwa tujuan dari penelitian ini adalah untuk menganalisa keakuratan *collaborative filtering multicriteria* dalam

pemberian rekomendasi jalur yang perlu digunakan di dalam game *virtual reality* pengenalan kampus UIN Malang.

#### **1.4 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberi sarana pengenalan kampus UIN Malang bagi mahasiswa baru ataupun orang luar yang belum mengenal lingkungan kampus.
2. Membantu *player* mendapatkan jalur yang terbaik untuk pergi ke gedung saintek dari ma'had laki-laki

#### **1.5 Batasan Penelitian**

Batasan penelitian dari penelitian adalah sebagai berikut:

1. Kemiripan atau *similarity* dari pengguna diukur menggunakan metode *cosine similarity*.
2. Perhitungan prediksi dilakukan dengan menggunakan metode *weighted sum*.
3. Pengevaluasian sistem rekomendasi dilakukan dengan menghitung akurasi, presisi, *recall* dan *F-score*.
4. *User Segmentation*: Mahasiswa UIN Malang tahun ajaran baru fakultas sains & teknologi.

## BAB II

### STUDI PUSTAKA

#### 2.1 Penelitian Terkait

Virgantara Putra, *et al.* (2022) dengan penelitiannya yang berjudul “Simulasi 3 Dimensi Rute Evakuasi Bencana di Gontor Putri Kampus 3 Berbasis *Virtual reality* (VR)” menyatakan bahwa mitigasi melalui simulasi dapat dilakukan sebagai untuk mengantisipasi bencana. Penelitian tersebut juga berkesimpulan bahwa simulasi yang dilakukan dengan menggunakan *Virtual reality* lebih efektif dibandingkan simulasi dengan menggunakan video karena pengguna dapat merasakan seolah-olah masuk dan terbenam (*immerse*) ke dalam lingkungan virtual (Virgantara Putra *et al.*, 2022).

Ma'ruf & Qoiriah (2022) membandingkan algoritma *cosine similarity* dengan *euclidean distance* dalam mencari nilai kemiripan *item* pada penelitian mereka. Perbandingan dilakukan dengan menghitung nilai *error* untuk mengevaluasi hasil akurasi pada tiap algoritma yang digunakan. Pada percobaan yang dijalankan, algoritma *cosine similarity* menghasilkan nilai *Mean Absolute Error* sebesar 2,21 serta nilai *Root Mean Square Error* sebesar 2,51. Sedangkan pada percobaan dengan menggunakan algoritma *Euclidean Distance* menghasilkan nilai *Mean Absolute Error* sebesar 2,24 serta nilai *Root Mean Square Error* sebesar 2,55. Dari hasil penelitian yang telah dilakukan, algoritma *Cosine similarity* memiliki nilai *error* yang lebih rendah dibandingkan dengan algoritma *Euclidean Distance*. Hal tersebut dapat dilihat dari nilai *Mean Absolute Error* dan juga *Root Mean Square*

*Error* pada algoritma *Cosine similarity* yang lebih memiliki nilai lebih kecil dari algoritma Euclidean Distance (Ma'ruf & Qoiriah, 2022).

Berdasarkan *review* literatur yang dilakukan oleh Intan Hervianda Putri *et al.*, (2022) yang berfokus pada sistem rekomendasi buku dan film, peneliti berkesimpulan bahwa metode *collaborative filtering* adalah metode yang paling populer digunakan. Hal tersebut dikarenakan metode *collaborative filtering* bisa digunakan secara langsung maupun dikombinasikan dengan algoritma lain ataupun mengimprovisasi algoritma di dalamnya, dengan tujuan untuk mengembangkan sistem rekomendasi agar menjadi lebih efisien dan akurat (Intan Hervianda Putri *et al.*, 2022).

Janan (2020) melakukan penelitian terhadap sistem rekomendasi *User-Based Collaborative Filtering* untuk game rekomendasi tempat wisata di Kota Batu. *Rating* wisata yang wisatawan pernah dikunjungi sebelumnya diperlukan untuk menentukan rekomendasi tersebut, yang dimana data inputan *rating* tersebut akan tersimpan di database. Dengan menggunakan metode *collaborative filtering* dalam membangun sistem rekomendasi tersebut, sebesar 70% dari total responden menyetujui bahwa hasil rekomendasi memenuhi ekspektasi, sebesar 81.5% responden juga menyetujui bahwa game dapat ikut andil dalam peningkatan pengetahuan khalayak ramai mengenai destinasi wisata di Kota Batu. Pengukuran kesuksesan dilakukan dengan cara pemberian kuesioner yang diberikan kepada 50 responden. Berdasarkan pengujian *Precision*, *Recall* dan F-Measure berturut-turut didapatkan hasil 0.8, 1, dan 0.889. Dari hasil tersebut dapat disimpulkan bahwa

metode *User Based Collaborative Filtering* cukup akurat dalam memberikan rekomendasi (Janan, 2020).

Aisha (2022) dalam penelitiannya yang berjudul “Sistem Rekomendasi Toko Online menggunakan Algoritma *Collaborative Filtering* dan *Content Based Filtering*” berkesimpulan bahwa algoritma *content based filtering* maupun *collaborative filtering* memiliki perbedaan tersendiri dalam merekomendasikan hasilnya. *Content based filtering* dapat memberikan rekomendasi terhadap item yang belum pernah *dirating* dari user-nya sekalipun, namun *collaborative filtering* merupakan kebalikannya yang tidak dapat memberikan rekomendasi jika yang *dirating* belum ada. Menggunakan *Mean Absolute Error*, didapatkan nilai *error* pada algoritma *collaborative filtering* sebesar 2,31, sedangkan nilai *error* pada *content based filtering* 3,8. Algoritma *collaborative filtering* memiliki nilai *error* yang lebih rendah, sehingga dapat disimpulkan bahwa algoritma *collaborative filtering* lebih akurat jika dibandingkan dengan algoritma *content based filtering* (Aisha, 2022).

Penelitian yang dilakukan oleh Yusmar et al. (2021) yang berjudul “*Restaurant Recommender System Using Item Based Collaborative Filtering and Adjusted Cosine Algorithm Similarity*” menjelaskan bahwa banyaknya informasi yang tersedia terkadang membuat orang *over-informed*, sehingga sulit untuk memilih restoran berdasarkan preferensi mereka. Oleh karena itu, diperlukanlah sebuah sistem rekomendasi untuk membantu pengguna dalam pemilihan restoran. Penelitian ini mengimplementasikan sistem rekomendasi *item-based collaborative filtering* dengan menggunakan metode *adjusted cosine similarity*. Pengetesan akurasi dilakukan dengan 3 skenario: 5 pengguna, 20 pengguna dan 40 pengguna.

Skenario ke-3 memberikan presisi yang lebih baik sehingga peneliti berkesimpulan bahwa sistem rekomendasi *item-based collaborative filtering* memiliki nilai akurasi dan error paling baik jika data penggunaannya lebih banyak (Yusmar *et al.*, 2021).

## **2.2 Landasan Teori**

Landasan teori merupakan bagian dari kerangka teoritis yang mendukung dan menjelaskan konsep-konsep, teori, atau ide-ide yang menjadi dasar penelitian. Landasan teori membantu peneliti untuk memahami fenomena yang diteliti dan memberikan dasar logis untuk hipotesis yang diajukan. Pada landasan teori penelitian ini akan membahas tentang simulasi, *virtual reality*, *path*, sistem rekomendasi, *collaborative filtering*, *cosine similarity*, *weighted sum* dan evaluasi sistem.

### **2.2.1 Simulasi**

Simulasi adalah kegiatan yang dilakukan untuk meminimalkan resiko atau kerugian bagi manusia (Virgantara Putra *et al.*, 2022). Selain pengertian tersebut, terdapat beberapa pendapat ahli mengenai pengertian dari simulasi. Menurut Emsoff & Simun (1970), simulasi adalah sebagai suatu model sistem dimana komponennya di presentasikan oleh prosesor-prosesor aritmetika dan logika yang dijalankan komputer untuk memperkirakan sifat-sifat dinamis system tersebut. Menurut Banks & Carson (1984), simulasi adalah tiruan sistem nyata yang dikerjakan secara manual atau komputer, yang kemudian di observasi dan disimpulkan untuk mempelajari karakterisasi sistem. Menurut Law & Kelton (1991), simulasi didefinisikan sebagai sekumpulan metode dan aplikasi untuk

menirukan atau mereprestasikan perilaku dari suatu sistem nyata, yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu. Dari beberapa pendapat diatas, bisa disimpulkan bahwa simulasi adalah suatu teknik numerik untuk melakukan percobaan-percobaan pada suatu komputer digital, yang melibatkan bentuk-bentuk fungsi matematika dan logika tertentu untuk menjelaskan tingkah laku dan struktur suatu nyata.

Game simulasi merupakan game yang umumnya dirancang untuk mensimulasikan aktivitas dunia nyata secara dekat. Simulasi game berusaha untuk meniru berbagai aktivitas dari kehidupan nyata dalam bentuk game untuk berbagai keperluan seperti pelatihan, analisis, atau prediksi. Biasanya tidak ada tujuan yang ditentukan secara ketat dalam permainan, dengan pemain diizinkan untuk mengontrol karakter atau lingkungan dengan bebas. Pada penelitian yang akan dilakukan, lingkungan kampus UIN Malang akan disimulasikan dalam bentuk game *virtual reality* sehingga pengguna bisa menjelajahi lingkungan kampus tanpa perlu ke lokasi kampus tersebut.

### **2.2.2 *Virtual reality***

*Virtual Reality* adalah teknologi yang membuat pengguna bisa berinteraksi dengan dunia *virtual* (Mambu *et al.*, 2019). *Virtual reality* (VR) adalah teknologi *interface* manusia-komputer canggih yang mensimulasikan lingkungan yang realistis. Para pengguna dapat bergerak di dunia maya. Mereka bisa melihatnya dari berbagai sudut, bergerak dan juga berinteraksi di dalamnya.

Konsep *virtual reality* telah ada selama beberapa dekade, tetapi baru pada tahun 1970-an teknologinya mulai terbentuk. Pada tahun 1990-an, sejumlah

perusahaan mulai mengeksplorasi potensi VR untuk game dan hiburan, dan perangkat VR umum pertama dirilis. Namun, teknologinya masih sangat terbatas, dan baru pada tahun 2010-an VR mulai terlihat mencapai potensinya. Saat ini, VR digunakan untuk berbagai aplikasi, mulai dari game dan hiburan hingga pendidikan, pelatihan dan perawatan kesehatan. Dengan terus berkembangnya *hardware* dan *software* baru, kemungkinan besar teknologi *virtual reality* akan memberikan dampak yang semakin signifikan bagi kehidupan kita di tahun-tahun mendatang.

Dengan adanya *virtual reality*, pengguna dimungkinkan untuk menjelajahi dan berinteraksi dengan lingkungan virtual yang mendekati kenyataan, dengan cara mempersepsikan indra pengguna. Lingkungan dibuat dengan perangkat keras dan perangkat lunak komputer, meskipun pengguna mungkin juga perlu memakai perangkat seperti helm atau kacamata untuk berinteraksi dengan lingkungan. Semakin dalam pengguna dapat membenamkan diri dalam lingkungan VR dan memblokir lingkungan fisik mereka semakin mereka mampu menanggukkan keyakinan mereka dan menerimanya sebagai nyata, bahkan jika lingkungannya bersifat fiktif. Pada penelitian yang akan dilakukan, *virtual reality* akan digunakan untuk mensimulasikan lingkungan kampus tersebut.

### **2.2.3 Path**

*Path* dalam video game mengacu pada rute atau lintasan yang telah ditentukan sebelumnya yang diikuti oleh pemain atau karakter dalam dunia game (Rogers, 2014). Jalur ini dapat mengambil berbagai bentuk, seperti koridor, jalan, *trail*, atau rute khusus apa pun yang dibuat oleh perancang tingkat. *Path* merupakan hal mendasar dalam memandu pergerakan dan perilaku entitas di dunia game.

Dalam penelitian ini, aspek *path* game yang digunakan adalah *path* untuk pergerakan karakter, yang dimana *path* tersebut digunakan untuk menentukan pergerakan karakter dalam game. Hal tersebut dapat mencakup rute yang diambil karakter saat menavigasi lingkungan, seperti berjalan di sepanjang jalur atau *trail* yang telah ditentukan atau mengikuti lintasan tertentu. *Path* dalam video game adalah rute yang telah ditentukan atau dihitung secara dinamis yang dapat diikuti oleh sebuah entitas dalam dunia game. *Path* sangat penting untuk menciptakan pola dan perilaku gerakan yang realistis dan menarik.

#### **2.2.4 Sistem Rekomendasi**

Sistem rekomendasi adalah peralatan perangkat lunak dan suatu teknik yang menyediakan rekomendasi *item* yang berguna untuk pengguna (Mondi *et al.*, 2019). Sistem rekomendasi adalah sistem pemfilteran informasi yang memberikan rekomendasi yang dipersonalisasi kepada pengguna berdasarkan preferensi, perilaku, dan data relevan lainnya. Tujuan dari sistem rekomendasi adalah untuk membantu pengguna menemukan produk, layanan, atau konten baru yang mungkin mereka minati, berdasarkan perilaku mereka di masa lalu atau perilaku pengguna serupa.

Terdapat dua tipe utama sistem rekomendasi, diantaranya adalah sistem rekomendasi *content-based*, dan sistem rekomendasi *collaborative filtering*. Sistem rekomendasi *content-based* akan merekomendasikan *item* yang mirip dengan *item* yang sebelumnya disukai atau memiliki interaksi dengan pengguna. Sistem rekomendasi *collaborative filtering* di sisi lain, merekomendasikan *item*

berdasarkan perilaku pengguna lain yang memiliki preferensi serupa dengan pengguna yang bersangkutan.

Sistem rekomendasi digunakan dalam berbagai aplikasi, termasuk *e-commerce*, media sosial, hiburan, *streaming*, dll. Sistem rekomendasi dapat digunakan untuk merekomendasikan produk, film, musik, artikel berita, dan banyak lagi. Sistem rekomendasi menjadi semakin penting dalam beberapa tahun terakhir karena jumlah data yang tersedia telah bertambah, sehingga akan menjadi semakin sulit bagi pengguna untuk menemukan konten yang relevan seorang diri. Dengan memberikan rekomendasi yang dipersonalisasi, sistem rekomendasi membantu pengguna menemukan item baru dan memberikan pengalaman baik kepada pengguna secara keseluruhan.

Sistem rekomendasi *multicriteria* adalah jenis sistem rekomendasi yang memperhitungkan banyak kriteria saat membuat rekomendasi kepada pengguna. Dalam sistem rekomendasi tradisional, rekomendasi didasarkan pada satu kriteria, seperti perilaku pengguna di masa lalu atau *rating* yang mereka berikan pada *item*. Namun, dalam sistem rekomendasi *multicriteria*, rekomendasi didasarkan pada kombinasi faktor, seperti perilaku pengguna di masa lalu, informasi demografis, dan preferensi pengguna lainnya.

Gagasan di balik sistem rekomendasi *multicriteria* adalah untuk memberikan rekomendasi yang lebih personal dan akurat dengan mempertimbangkan berbagai faktor yang lebih luas. Hal ini bisa sangat berguna dalam situasi di mana pengguna memiliki preferensi yang kompleks dan bervariasi, atau di mana ada sejumlah besar data untuk dianalisis. Sistem rekomendasi *multicriteria* umumnya digunakan dalam

*e-commerce*, di mana mereka dapat membantu pengguna menemukan produk yang sesuai dengan preferensi dan kebutuhan unik mereka. Sama seperti sebelumnya juga, sistem rekomendasi *multicriteria* juga baik untuk digunakan di industri lain, seperti hiburan dan *streaming*, di mana rekomendasi dapat didasarkan pada riwayat tontonan pengguna, genre yang disukai, dan faktor lainnya. Secara keseluruhan, tujuan dari sistem pemberi rekomendasi *multicriteria* adalah untuk memberikan rekomendasi yang lebih dipersonalisasi, akurat, dan relevan kepada pengguna, yang menghasilkan kepuasan dan keterlibatan pengguna yang lebih tinggi. Untuk menjalankan fungsinya, MCRS bekerja dalam 2 bagian, yaitu bagian prediksi juga bagian rekomendasi. Bagian prediksi adalah bagian dari sistem yang akan memprediksi nilai seorang *user* terhadap item tertentu dengan membandingkan preferensi *user* tersebut, sedangkan bagian rekomendasi adalah bagian dimana pengguna mendapatkan item rekomendasi (Arif *et al.*, 2022).

### **2.2.5 Collaborative Filtering**

*Collaborative filtering* adalah teknik yang digunakan dalam sistem rekomendasi untuk membuat prediksi tentang minat pengguna berdasarkan preferensi sekelompok pengguna lain (Intan Hervianda Putri *et al.*, 2022). *Collaborative filtering* mengasumsikan bahwa jika dua pengguna memiliki preferensi yang sama untuk beberapa *item*, maka mereka juga cenderung memiliki preferensi yang sama untuk item lainnya.

Ada dua jenis utama dari *collaborative filtering*: *user-based* dan *item-based*. *User-based collaborative filtering* memprediksi preferensi pengguna dengan menemukan pengguna lain yang memiliki *rating item* serupa dan menggunakan

*rating* mereka untuk membuat rekomendasi. *Item-based collaborative filtering* di sisi lain, memprediksi preferensi pengguna dengan menemukan *item* lain yang mirip dengan *item* yang dinilai tinggi oleh pengguna dan merekomendasikan *item* tersebut.

Algoritma *collaborative filtering* bekerja dengan cara membuat matriks dari *rating item-pengguna*, di mana setiap baris mewakili pengguna dan setiap kolom mewakili *item* atau kriteria. Matriksnya keseringan menyebar, karena terdapat kemungkinan bahwa pengguna hanya akan menilai sebagian kecil dari *item* yang tersedia. Algoritmanya-pun kemudian menggunakan berbagai pengukuran kesamaan untuk mengidentifikasi pengguna atau *item* yang mirip satu sama lain dan membuat rekomendasi berdasarkan kesamaan tersebut.

*Collaborative filtering* banyak digunakan dalam sistem rekomendasi di *e-commerce*, media sosial, dan area lain di mana rekomendasi yang dipersonalisasi dapat membantu pengguna menemukan produk atau layanan baru yang mungkin mereka minati. Salah satu keuntungan *collaborative filtering* adalah sistem rekomendasinya dapat bekerja dengan baik bahkan ketika ada sedikit atau tidak ada informasi tentang item yang direkomendasikan, selama ada kumpulan data *rating item pengguna* yang cukup besar. Berdasarkan alasan tersebut, maka penelitian ini akan menggunakan *collaborative filtering* daripada *content-based filtering*.

### **2.2.6 Cosine similarity**

*Cosine similarity* adalah pengukuran *similarity* yang umum digunakan dalam *collaborative filtering*. *Cosine similarity* adalah pengukuran kesamaan antara dua vektor bukan nol yang mengukur cosinus sudut di antara keduanya

(Mondi *et al.*, 2019). Dalam konteks sistem rekomendasi *collaborative filtering*, biasanya digunakan untuk mengukur kesamaan antara dua pengguna atau antara pengguna dan *item*, berdasarkan vektor *rating* mereka.

*Cosine similarity* berkisar dari 0 hingga 1, di mana 1 berarti bahwa dua vektor menunjuk ke arah yang sama (yaitu, memiliki *rating* yang identik untuk semua *item*) sedangkan 0 berarti vektor tersebut tidak memiliki kesamaan. *Cosine similarity* adalah ukuran tingkat tumpang tindih antara dua vektor dan digunakan untuk menemukan pengguna atau item yang paling mirip dengan pengguna atau item tertentu. Adapun rumus dari algoritma *cosine similarity* adalah sebagai berikut:

$$\cos(\theta) = \text{sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.1)$$

Keterangan:

$\text{sim}(A, B)$  = *similarity* antara user A dan user B

$A_i$  = Nilai *rating* yang diberikan user A

$B_i$  = Nilai *rating* yang diberikan user B

### 2.2.7 Weighted sum

Model *weighted sum* adalah jenis model matematika yang digunakan di berbagai bidang, termasuk *machine learning* statistik, dan decision making. Dalam konteks *machine learning* dan *decision making*, model *weighted sum* dapat digunakan untuk membuat prediksi atau keputusan berdasarkan beberapa fitur masukan / *rating* (Yusmar *et al.*, 2021). Misalnya, dalam sistem rekomendasi, model *weighted sum* dapat digunakan untuk menggabungkan *rating* pengguna yang serupa untuk memprediksi *rating* pengguna target untuk *item target*. Dalam konteks

pengambilan keputusan, model *weighted sum* dapat digunakan untuk menggabungkan skor beberapa kriteria untuk mengevaluasi keinginan dari pilihan yang berbeda. Adapun rumus dari *weighted sum* adalah sebagai berikut: (2.2)

$$P(a, j) = \frac{\sum_{i \in I} (R_{a,i} * S_{i,j})}{\sum_{i \in I} |S_{i,j}|}$$

Keterangan :

P(a,j) = Nilai Prediksi untuk item a terhadap user j

R(a,i) = Penilaian untuk item a oleh user i

S(i,j) = Nilai *similarity* antar user i dan user j

### 2.2.8 Evaluasi Sistem

Evaluasi sistem dilakukan untuk mengetahui seberapa tinggi tingkat akurasi dari hasil sistem rekomendasi pada setiap *user*. Dalam penelitian ini, akan digunakan *confusion matrix* sebagai standar untuk pengevaluasian sistem. Dalam konteks sistem rekomendasi, *confusion matrix* dapat digunakan untuk mengukur kemampuan sistem dalam memprediksi preferensi pengguna secara akurat. Baris matriks mewakili *rating* sebenarnya dari pengguna, sedangkan kolom mewakili perkiraan *rating* sistem. Elemen matriks mewakili jumlah *item* yang diklasifikasikan ke dalam setiap kombinasi *rating* aktual dan prediksi. *Confusion matrix* biasanya memiliki empat kategori:

- a. *True Positives* (TP): jumlah item yang diklasifikasikan dengan benar dan bernilai positif.
- b. *True Negatives* (TN): jumlah item yang diklasifikasikan dengan benar dan bernilai negatif.
- c. *False Positives* (FP): jumlah item yang salah diklasifikasikan dan sistem memprediksi bahwa nilainya positif.

- d. *False Negatives* (FN): jumlah item yang salah diklasifikasikan dan sistem memprediksi bahwa nilainya negatif.

Dari *confusion matrix*, beberapa metrik dapat dihitung untuk mengevaluasi kinerja sistem, seperti akurasi, presisi, *recall*, dan skor F1. *Confusion matrix* adalah alat yang berguna untuk memvisualisasikan kinerja model klasifikasi dan untuk mengidentifikasi area di mana model perlu diperbaiki. Dari *confusion matrix* tersebut, dapat dihasilkanlah metrik untuk menghitung evaluasi sistem berikut:

- Akurasi

Akurasi mengukur proporsi semua rekomendasi yang benar. Akurasi dihitung sebagai jumlah rekomendasi yang benar dibagi dengan jumlah total rekomendasi yang dibuat oleh sistem. Rumus dari Akurasi adalah sebagai berikut:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.3)$$

- Presisi

Presisi mengukur proporsi positif sejati dari semua item yang diprediksi positif. Rumus dari presisi adalah sebagai berikut:

$$Precision = \frac{TP}{(TP + FP)} \quad (2.4)$$

- *Recall*

*Recall* mengukur proporsi positif sejati dari semua item yang benar-benar positif. Rumus dari *recall* adalah sebagai berikut:

$$Recall = \frac{TP}{(TP + FN)} \quad (2.5)$$

- *F-Score*

*F-Score* adalah rata-rata dari presisi dan *recall*, sehingga menyeimbangkan presisi dan daya ingat. Rumus dari *F-Score* adalah sebagai berikut:

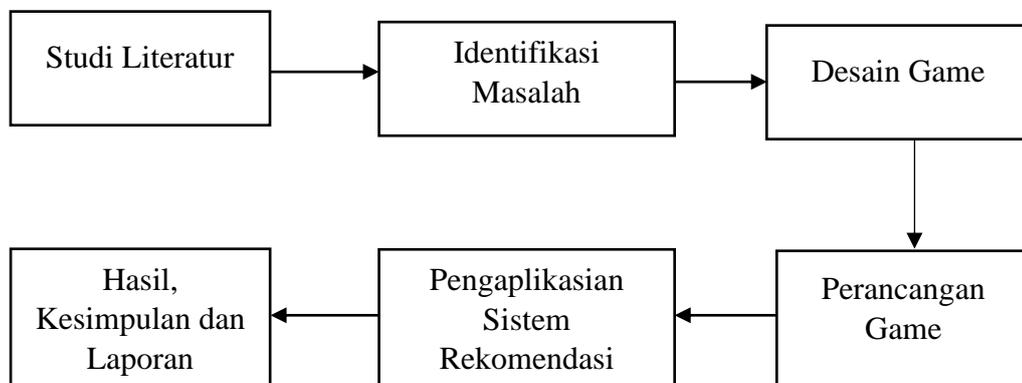
$$F - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (2.6)$$

## BAB III

### METODE PENELITIAN

#### 3.1 Desain Penelitian

Desain penelitian adalah rencana atau strategi yang digunakan untuk melakukan penelitian dan menyelidiki pertanyaan penelitian atau hipotesis. Penelitian dilakukan dengan pengumpulan referensi / studi literatur terhadap topik yang ingin disentuh, yang dimana topiknya adalah *virtual reality* untuk edukasi. Berhubungan dengan topik tersebut, ditetapkanlah pertanyaan penelitian yang berupa bagaimana untuk mengaplikasikan sistem rekomendasi *collaborative filtering multicriteria* kepada simulasi yang ingin dibangun. Alur penelitiannya ditunjukkan oleh Gambar 3.1 berikut:



Gambar 3.1 Alur Penelitian

#### - Studi Literatur

Pada tahap ini, dilakukan studi literatur terhadap topik-topik yang berhubungan penelitian untuk memahami topik-topik tersebut lebih dalam. Topik-topik tersebut antara lain adalah: perancangan game 3D dengan menggunakan *unity*

*game engine*, sistem rekomendasi, sistem rekomendasi *multicriteria*, sistem rekomendasi *collaborative filtering* dan juga penelitian terkait yang berhubungan.

- Identifikasi Masalah

Pada tahap ini, ditetapkanlah masalah penelitian yang berupa bagaimana untuk mengaplikasikan sistem rekomendasi *collaborative filtering multicriteria* kepada game simulasi pengenalan kampus.

- Desain Game

Pada tahap ini, game yang nantinya akan dirancang didesain terlebih dahulu. Hal-hal yang perlu di-design antara lain adalah seperti model game, alur game serta cara kerja game secara keseluruhan.

- Perancangan Game

Pada tahap ini, desain game yang telah dibuat ditranslasikan kedalam bentuk game (dirancang). Simulasi game akan dirancang dengan menggunakan *game engine unity 3D*.

- Pengaplikasian Sistem Rekomendasi

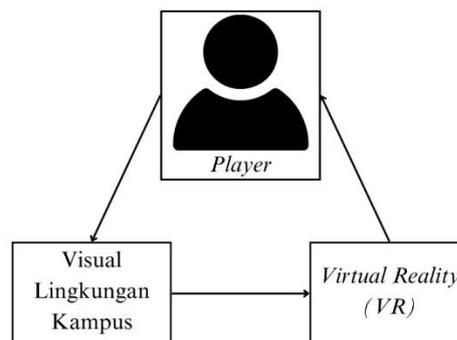
Pada tahap ini, sistem rekomendasi *collaborative filtering multicriteria* yang telah disusun diimplementasikan ke dalam game yang telah dirancang.

- Hasil, Kesimpulan dan Laporan

Setelah semuanya selesai dilakukan, kemudian hasil dan kesimpulan dari penelitian dapat diambil dan dilaporkan dalam bentuk laporan. Pengukuran akurasi dari sistem rekomendasi yang telah dibuat juga dimasukkan di sini.

### 3.2 Desain Game

Game didesain dengan tujuan untuk mengenalkan *player* dengan lingkungan serta keadaan atau informasi terkait kampus. Saat game dimulai, pertama-tama yang *player* lakukan adalah memilih level yang dimana nilai *rating player* tersebut yang akan dimasukkan ke dalam sistem rekomendasi didapat dari level yang dipilih *player*. Dari level yang dipilih, *player* akan diberikan sejumlah misi. Sistem rekomendasi yang ada akan memberitahu *player* untuk sebaiknya menyelesaikan misi yang mana terlebih dahulu dari titik lokasi *player* yang sekarang. Waktu yang diperlukan level untuk menyelesaikan suatu level tersebut-pun akan tersimpan.



Gambar 3.2 Alur Game

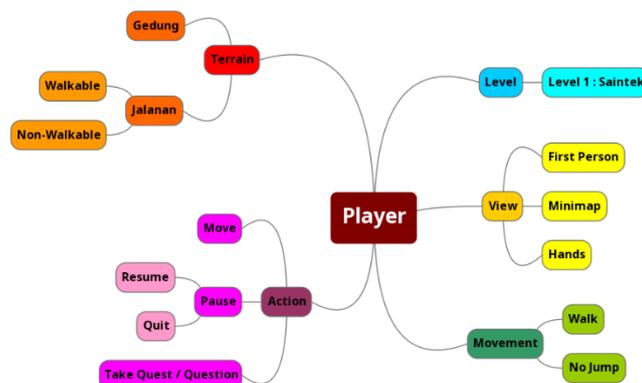
Gambar 3.2 menunjukkan alur saat pilihan *start* dijalankan. Pada *main menu* saat *player* memilih tombol *start game*, *player* akan masuk ke dalam level 1 dan dalam level 1 tersebut pun, *similarity player* akan diukur dengan sistem rekomendasi yang telah disusun menggunakan *collaborative filtering*. Saat proses *loading environment* selesai, *player* akan memulai level dan lingkungan kampus

akan divisualisasikan VR kembali kepada *player*. Berikut ini merupakan *flowchart* yang menunjukkan apa yang terjadi ketika suatu level dijalankan.



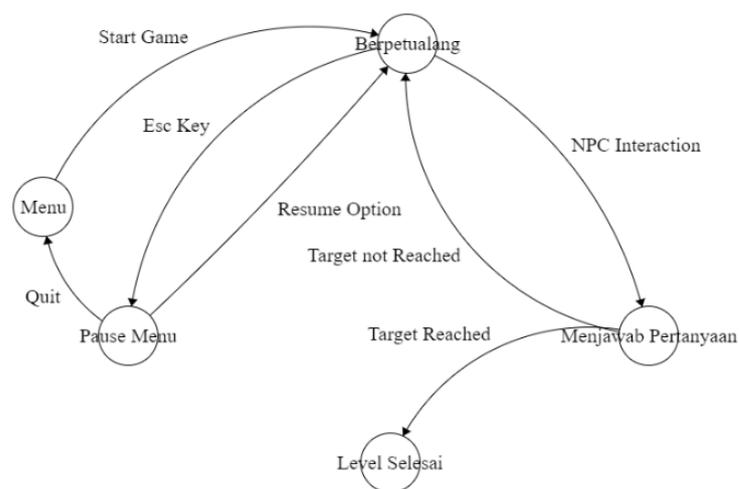
Gambar 3.3 *Flowchart* Jalan Level Pada Game

Gambar 3.3 menunjukkan bahwa ketika suatu level dimulai, *player* diperlukan mendatangi titik target dimana quiz bisa dijalankan. Lokasi quiz tiap level berbeda-beda. *Player* perlu mendapatkan skor 80 atau lebih tinggi untuk menyelesaikan level dan masuk ke level berikutnya sehingga game diselesaikan. Berikut ini merupakan gambar 3.4 yang menunjukkan *mind map* desain game.



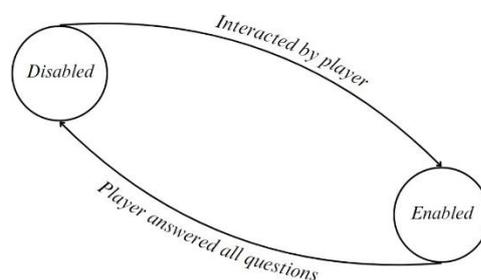
Gambar 3.4 *Mindmap* Desain Game

Gambar 3.4 menunjukkan tentang mindmap mengenai apa yang dilihat, dialami dan apa yang dapat dilakukan *player*. Untuk *child terrain* atau lingkungan dari game terdapat gedung-gedung kampus sekaligus jalanan-jalanan dalam kampus. Untuk *child action* atau apa yang dapat dilakukan *player* dalam game, terdapat *action* untuk bergerak (*move*), *action pause* untuk menghentikan game dan *action take question* untuk mengambil pertanyaan. Untuk *child movement* memberitahukan bahwa dalam pergerakan *player*, *player* dapat berjalan namun tidak dapat melompat. Untuk *child view* atau apa yang dapat dilihat *player* dalam *screen* ketika game sedang berjalan adalah *view first person* (sudut pandang pertama), minimap dari lingkungan sesungguhnya yang di *center*-kan terhadap *player*, dan tangan *player* yang digunakan untuk berinteraksi dengan lingkungan game. Untuk *child level* memberitahu bahwa terdapat 1 level dalam game, yang dimana pada level pertama titik target terdapat di Gedung Saintek. Selain gambar 3.4 tersebut, berikut ini adalah gambar yang menunjukkan *gameplay player* ketika menjalankan game pada suatu level.



Gambar 3.5 *Gameplay*

Gambar 3.5 menggambarkan tentang apa yang perlu dilakukan *player* (apa yang akan ditampilkan *screen*) saat berada di dalam game. Saat game dijalankan, *player* akan bermula dari *scene menu* yang kemudian dapat memilih tombol ‘*start game*’. Saat level dijalankan *player* akan bertempat di depan parkirannya ma’had putra, yang dimana *player* akan diperlukan untuk berpetualang lingkungan kampus untuk mencari titik target yang telah ditentukan pada level. Terdapat *option pause* untuk menghentikan game sesaat ataupun keluar dari game jika *player* ingin melakukannya ditengah-tengah jalan game. Ketika *player* menemukan titik target *quest*, *player* dapat berinteraksi dengan *game object* yang berada pada titik *quest* tersebut untuk menjawab pertanyaan-pertanyaan yang telah ditentukan dalam setiap level. Jika skor *player* dalam menjawab pertanyaan-pertanyaan yang ada melebihi 80, *player* dinyatakan lulus level dan dapat lanjut ke level berikutnya. Jika skor level *player* tidak mencapai 80, *player* akan memulai ulang level tersebut dari lokasi awal *player* berada. Gambar berikut ini merupakan FSM dari window pertanyaan dari *game object* pada titik lokasi untuk lebih menjelaskan gambar *gameplay* di atas.

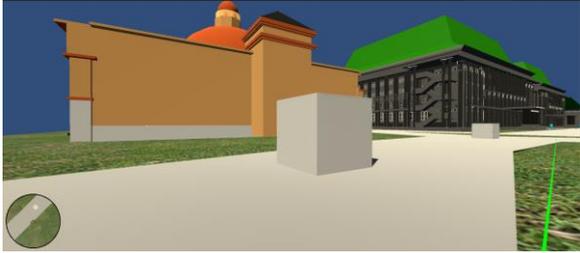


Gambar 3.6 FSM *Window* Pertanyaan *Gameobject* Pada Titik *Quest*

Gambar 3.6 di atas menunjukkan 2 *state* yang dimiliki oleh window pertanyaan quiz yang berhubungan dengan *gameobject* yang terdapat pada titik target / *quest*. *Gameobject* yang tidak terlihat pada *screen* atau *disabled* dari awal game dimulai akan tampak atau *enabled* ketika player berinteraksi dengan *gameobject* yang terdapat pada titik target, memberikan pertanyaan-pertanyaan seputar gedung sains teknologi.

Berikut ini merupakan tabel yang menunjukkan *storyboard* dari contoh jalan level setelah level dijalankan.

Tabel 3.1 *Storyboard* Contoh Jalan Level

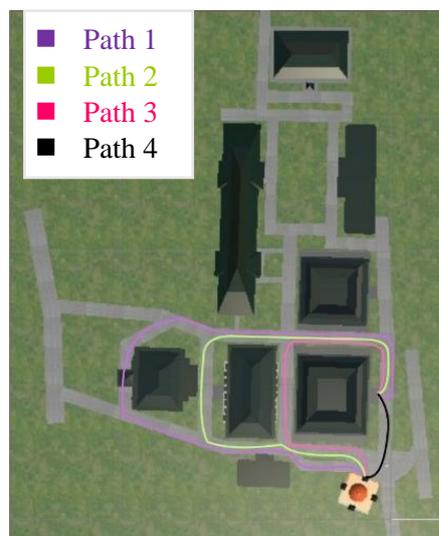
Gambar	Keterangan
	<p><i>Player</i> memulai level</p>
	<p><i>Player</i> bertempat di depan lingkungan ma'had</p>
	<p><i>Player</i> mendapat misi sesuai level</p>
	<p><i>Player</i> pergi ke tempat dimana misi perlu diselesaikan</p>

Gambar	Keterangan
	<p><i>Player menyelesaikan misi</i></p>
	<p>Level selesai</p>

Berikut ini adalah beberapa indikator untuk menandakan keberhasilan game:

- *Player* mengetahui jalan kampus dari ma'had laki-laki hingga gedung saintek
- *Player* mengetahui gedung-gedung yang perlu dilewati jika ingin pergi ke gedung saintek dari ma'had laki-laki
- *Player* mengetahui informasi seputar fakultas sains & teknologi

Untuk *path-path* yang dapat dilalui pada level 1 (dari lokasi *player* yang berada di sebelah parkir ma'had hingga ke fakultas saintek) adalah sebagai berikut:

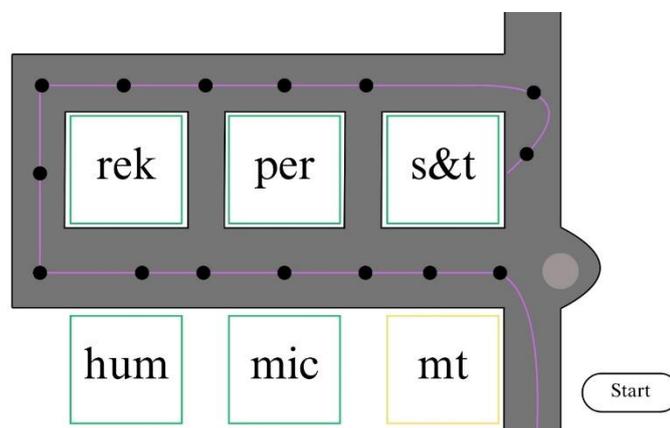


Gambar 3.7 Empat *Path* Yang Dapat Dilalui Untuk Pergi Ke Gedung Saintek

Gambar 3.7 menunjukkan 4 *path* utama yang dapat dilalui *player* jika *player* ingin pergi ke fakultas sains dan teknologi dari lingkungan mahad. Dalam tiap *path*, diberikan beberapa *waypoint* atau titik jalan yang terbentang dari lokasi titik awal hingga lokasi target. Informasi di bawah merupakan informasi lebih lanjut dari *path-path* tersebut. Kode rek, per, s&t, hum, mic dan mt masing-masing merupakan kode-kode untuk menunjukkan bangunan rektorat, perpustakaan, saintek, humaniora, *microteaching* dan masjid tarbiyah.

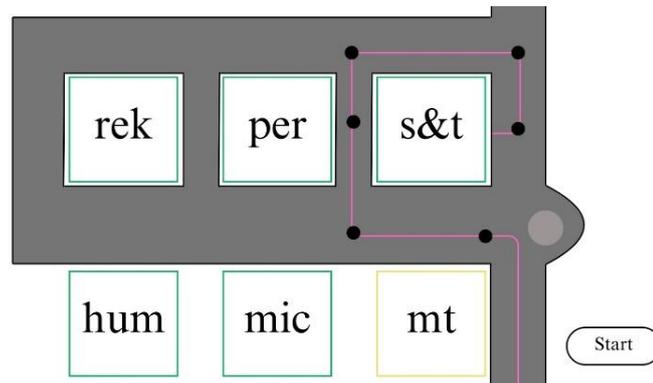
a. *Path 1*

*Path 1* dimulai dari lokasi titik awal, melewati masjid tarbiyah, gedung *micro teaching*, gedung fakultas humaniora lalu belok ke depan bangunan rektorat UIN Malang, lalu melingkarinya untuk mencapai gedung saintek lewat sisi selatan melewati gedung rektorat, perpustakaan dan gedung saintek bagian belakang. *Waypoint* yang digunakan ketika melewati *path* ini adalah sebanyak 15 *waypoint*. Berikut ini adalah gambar dari *path 1* (ditandai dengan warna ungu) beserta *waypoint-waypointnya*:



Gambar 3.8 *Path 1*

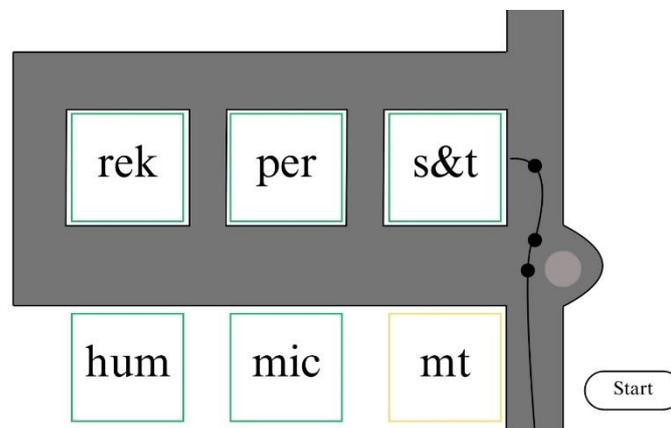




Gambar 3.10 Path 3

## d. Path 4

*Path 4* dimulai dari lokasi titik awal, melewati menara UIN Malang dan lurus hingga mencapai bagian depan gedung saintek. *Waypoint* yang digunakan ketika melewati *path* ini adalah sebanyak 3 *waypoint*. Berikut ini adalah gambar dari *path 4* (ditandai dengan warna hitam) beserta *waypoint-waypoint*nya:



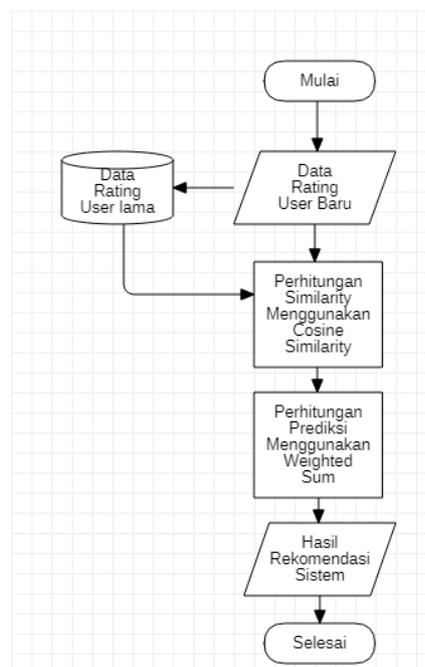
Gambar 3.11 Path 4

Pada setiap level, *path-path* yang akan diambil akan dinilai berdasarkan kriteria-kriteria berikut: *crowdiness* untuk nilai keramaian *path*, *distance* untuk menilai jarak dari titik awal ke titik akhir dari *path* yang diambil, *sighting* untuk menilai apa saja yang dapat dilihat dari *path* yang diambil untuk menambah

pengetahuan selingkup kampus dan *accessibility* untuk menilai keuntungan tambahan ketika *path* tersebut diambil (contoh: apakah melewati toko kecil, perpustakaan, dsb).

### 3.3 Desain Sistem

Sistem yang akan dirancang dalam penelitian dimulai dengan input-an *rating* dari user baru yang dimana *rating* dari user baru tersebut akan masuk ke tahap *collaborative filtering* untuk menghitung nilai *similarity*-nya dengan user lain dengan menggunakan metode *cosine similarity* (untuk *player* baru akan diberikan nilai netral semua untuk *matrix rating*-nya, diberikan nilai 1 tiap kriteria), setelah itu nilai prediksinya dihitung dengan menggunakan metode *weighted sum*. Setelah nilai prediksinya didapatkan, rekomendasi dari sistem akan diukur akurasi, presisi, *recall* dan *F-Score*-nya. Berikut ini merupakan *flowchart* yang menunjukkan jalannya sistem.



Gambar 3.12 *Flowchart* Jalan Sistem

Gambar 3.12 menunjukkan langkah-langkah sebagaimana sistem rekomendasi yang digunakan berjalan. *Player* akan memiliki *rating* dari tiap *path* yang ada yang dimana nilai dari tiap *rating* tersebut akan digunakan untuk membandingkan kemiripan *player* dengan tiap *user* yang ada. *Player* akan diberikan pilihan apabila ingin memasukkan *rating* atau tidak (*player* boleh memberikan *rating* jika *player* mengetahui tentang jalanan dalam lingkungan kampus). Jika *player* memilih untuk tidak memasukkan *rating*, *value* untuk tiap *path* pada *player* akan diberikan nilai satu. Setelah itu, kemiripan antara *player* dengan setiap *user* yang terdapat pada data akan diukur menggunakan *cosine similarity*. Setelah kemiripan *player* dengan setiap *user* yang ada didapatkan, *rating* *player* terhadap tiap *path* akan dihitung ulang menggunakan *weighted sum*, yang dimana 2 nilai tertinggi dari *weighted sum* akan diberikan kembali kepada *player* sebagai rekomendasi pilihan *path*.

### 3.3.1 Data Rating

Sesuai dengan gambar 3.12, tahap pertama dalam jalannya sistem rekomendasi *collaborative filtering multicriteria* adalah pemberian *rating* yang dilakukan oleh *user* baru, yang dimana *rating* yang diberikan oleh *user* baru tersebut akan dibandingkan *similarity*-nya dengan *rating user* yang sudah ada. Untuk memahami jalan sistem dengan lebih baik, berikut ini merupakan tabel berisi data *rating dummy* yang akan digunakan dalam perhitungan sistem pada subbab ini.

Tabel 3.2 Contoh Tabel Data *Rating*

<b>Kriteria</b> <b>User</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>
User1	5	5	1	1	1
User2	5	4	2	2	2
User3	2	2	4	3	3
User4	1	2	5	5	4
User5	4	3	3	2	?

Tabel di atas merupakan contoh data yang akan digunakan untuk percobaan perhitungan *similarity* antar *user* pada subbab ini, yang dimana dengan data tersebut akan dilakukan percobaan perhitungan prediksi untuk C5 User5 menggunakan *weighted sum*, hingga pengevaluasian perhitungan prediksinya.

### 3.3.2 Perhitungan *Similarity*

Tahap berikutnya adalah untuk menghitung *similarity* antar *user*. Perhitungan *Similarity* mengacu pada proses komputasi untuk mencari kesamaan antara 2 *user*, berdasarkan *rating* yang sudah diberikan. *User* yang perhitungan *similarity*-nya lebih mirip satu sama lain cenderung memiliki *rating* atau preferensi yang serupa, dan oleh karena itu lebih mungkin untuk direkomendasikan satu sama lain. Salah satu metode perhitungan *similarity* yang sering digunakan oleh sistem rekomendasi *collaborative filtering* adalah metode *cosine similarity*. Berikut ini adalah contoh perhitungan *cosine similarity* diantara 2 *user* sesuai dengan persamaan (2.1).

$$\begin{aligned}
 Sim(User1, User2) &= \frac{(5 * 5) + (5 * 4) + (1 * 2) + (1 * 2)}{\sqrt{5^2 + 5^2 + 1^2 + 1^2} \sqrt{5^2 + 4^2 + 2^2 + 2^2}} \\
 &= 0.970725343
 \end{aligned}$$

Seperti yang sudah dibahas di atas, nilai *similarity* memiliki kisaran antara 0 dan 1, yang dimana jika nilainya mendekati 1 maka kedua *user* memiliki selera yang sangat mirip dan sebaliknya. Berikut ini adalah contoh tabel *similarity* yang dimiliki oleh *user* yang ada.

Tabel 3.3 Contoh Perhitungan *Similarity* Antar *User*

	User1	User2	User3	User4	User5
User1	1	0.970725343	0.65178614	0.467473495	0.899842541
User2	0.970725343	1	0.79578357	0.635674156	0.973328527
User3	0.65178614	0.79578357	1	0.96237768	0.903651992
User4	0.467473495	0.635674156	0.96237768	1	0.765587619
User5	0.899842541	0.973328527	0.903651992	0.765587619	1

Tabel di atas merupakan percobaan perhitungan *similarity* yang telah dilakukan terhadap tiap data yang terdapat pada tabel 3.2 menggunakan persamaan (2.1). Dari tabel tersebut, dapat dilihat bahwa User1 memiliki *similarity* tertinggi dengan User2 daripada user-user lain dengan *similarity* sebesar 0.970725343. User2 memiliki *similarity* tertinggi dengan User5 daripada *user-user* lain dengan *similarity* sebesar 0.973328527. User3 memiliki *similarity* tertinggi dengan User4 daripada *user-user* lain dengan *similarity* sebesar 0.96237768 dan seterusnya.

### 3.3.3 Perhitungan Prediksi

Langkah berikutnya adalah untuk melakukan perhitungan prediksi. Perhitungan prediksi dalam sistem rekomendasi mengacu pada proses memprediksi *rating* atau preferensi pengguna untuk *item* yang belum diberi *rating* atau di-interaksikan. Perhitungan prediksi merupakan perhitungan kunci dalam banyak sistem rekomendasi, karena memungkinkan sistem membuat rekomendasi yang dipersonalisasi kepada pengguna. Metode yang digunakan untuk perhitungan

prediksi dalam penelitian ini adalah metode *weighted sum*. Berikut ini adalah contoh perhitungan *weighted sum* sesuai dengan persamaan (2.2).

$$P(C5, user5) = \frac{(1 * 0,899) + (2 * 0,973) + (3 * 0,903) + (4 * 0,765)}{|0,899| + |0,973| + |0,903| + |0,765|}$$

$$= 2,433$$

Perhitungan di atas merupakan contoh perhitungan untuk mencari nilai C5 User5 menggunakan *weighted sum*. Sebagai *numerator*-nya adalah nilai *rating* dari tiap *user* yang dikalikan dengan *similarity user* tersebut dengan User5 yang dimana 4 hasil dari perkalian tersebut akan ditambahkan. Sebagai *denominator*-nya adalah *similarity* dari setiap *user* yang ada terhadap User5 yang dimana 4 dari *similarity* tersebut akan ditambahkan. Setelah itu, nilai *numerator* yang didapatkan akan dibagi dengan nilai *denominator* yang didapatkan dan hasilnya adalah C5 User5 yang didapatkan menggunakan *weighted sum*. Karena pada percobaan hanya terdapat 5 baris data, perhitungan *weighted sum* dilakukan menggunakan semua baris data yang ada. Pada implementasi, *similarity* dari *player* terhadap *user-user* yang ada akan di-*ranking* terlebih dahulu dan 20% *user* dengan *rank* tertinggi yang akan dimasukkan kedalam perhitungan *weighted sum*.

### 3.3.4 Rencana Pengujian

Dan langkah yang terakhir untuk dilakukan adalah mengukur keakuratan dari sistem rekomendasi *collaborative filtering* yang digunakan terhadap rekomendasi jalur dengan pengujian akurasi. Pengujian akurasi dalam sistem rekomendasi dilakukan dengan cara membandingkan *rating* atau rekomendasi yang diprediksi dengan *rating* atau preferensi pengguna yang sebenarnya. Hal ini dapat dilakukan

dengan menggunakan berbagai metrik evaluasi, diantaranya seperti akurasi, presisi, *recall*, dan *F-score*. Presisi dan *recall* matrik pada sistem rekomendasi telah digunakan oleh banyak peneliti untuk evaluasi sistem rekomendasi, presisi diartikan sebagai rasio rekomendasi yang relevan dengan jumlah total item yang direkomendasikan. Perhitungan akurasi, presisi, *recall* dan *F-Score* akan dilakukan sesuai dengan persamaan (2.3), (2.4), (2.5), dan (2.6). Berikut ini adalah contoh hasil pengujian terhadap 2 *user* yang menggunakan data acuan sebanyak 5 *user*:

Tabel 3.4 Contoh Perbandingan Nilai

User	Nilai Asli		Nilai Prediksi	
U1	C1	5	C1	4
	C2	5	C2	3
	C3	1	C3	3
	C4	1	C4	2
	C5	1	C5	2
U2	C1	5	C1	3
	C2	4	C2	3
	C3	2	C3	3
	C4	2	C4	2
	C5	2	C5	2

Tabel 3.4 diatas merupakan perbandingan-perbandingan nilai dari data nilai asli yang sesuai dengan tabel 3.2 dan nilai prediksi yang masing-masing nilainya dihitung menggunakan pendekatan *weighted sum* sesuai dengan persamaan (2.2).

Tabel 3.5 Contoh *Confusion Matrix*

	Asli Positif	Asli Negatif
Prediksi Positif	1	2
Prediksi Negatif	2	5

Setelah tabel perbandingan nilai didapatkan, lalulah dapat diketahui nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN) dalam *confusion matrix*. TP adalah nilai yang nilai prediksi dan nilai aslinya sama-sama positif. TN adalah nilai yang nilai prediksi dan nilai aslinya sama-sama negatif. FP

adalah nilai yang nilai aslinya negatif namun nilai prediksinya positif. FN adalah nilai yang nilai aslinya positif namun nilai prediksinya negatif. Dalam konteks *rating* dalam penelitian ini, nilai 4 dan 5 akan diklasifikasikan sebagai positif dan nilai 1,2 dan 3 kebawah akan dikategorikan sebagai *negative*. Selain itu, nilai prediksi tidak akan digolongkan *true* jika memiliki perbedaan lebih dari 1 dari nilai asli. Setelah tabel *confusion matrix* didapatkan, maka contoh nilai akurasi presisi, *recall* dan *F-Score* dapat dihitung. Berikut ini adalah perhitungan akurasi, presisi, *recall* dan *F-Score* pada contoh data yang digunakan:

- Perhitungan Akurasi

$$Accuracy = \frac{(1 + 5)}{(1 + 5 + 2 + 2)} = 0,6$$

- Perhitungan Presisi

$$Precision = \frac{1}{(1 + 2)} = 0,33$$

- Perhitungan *Recall*

$$Recall = \frac{1}{(1 + 2)} = 0,33$$

- Perhitungan *F-Score*

$$FScore = 2 * \frac{(0,33 * 0,33)}{(0,33 + 0,33)} = 0,33$$

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Implementasi**

Disini rancangan yang telah dibicarakan dalam bab III diterapkan. Sistem dibuat dengan beberapa tahapan yang ditetapkan sebelumnya agar sesuai dengan apa yang dibutuhkan.

##### **4.1.1 Implementasi Perangkat**

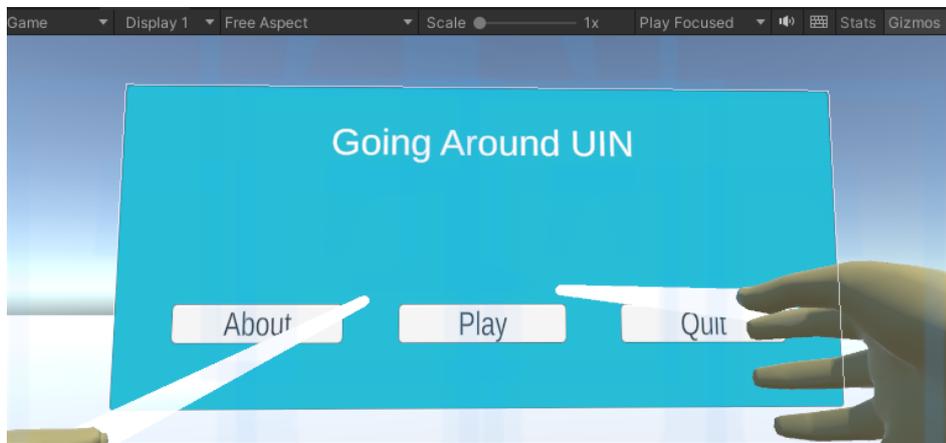
Berikut ini merupakan keterangan tentang perangkat yang digunakan dalam proses penelitian:

- a. Perangkat Keras
  - Intel® Core™ i3-7020U CPU
  - RAM 8 GB
  - NVIDIA® GeForce® MX230
  - Keyboard
  - Mouse
- b. Perangkat Lunak
  - Windows 10 Home Single Language
  - Unity 3D
  - Visual Studio 2022
  - Blender
  - Google Chrome

### 4.1.2 Implementasi User Interface

*User Interface* (UI) merupakan tampilan game yang dengan kata lain adalah penghubung antara *player* dengan sistem yang memiliki bentuk berupa tampilan game. Berikut ini merupakan tampilan-tampilan yang akan ditampilkan kepada *player* dari memulai game hingga *player* masuk ke level 1.

- a. Tampilan yang pertama kali ditampilkan kepada *player* ketika *player* memulai game adalah *Main menu*. Terdapat 3 tombol dalam tampilan main menu, yakni tombol *play*, *about* dan *quit*. Tombol *play* akan mengarahkan *player* untuk bermain game ke level 1, tombol *about* akan mengarahkan *player* untuk melihat deskripsi dari game dan tombol *quit* akan mengarahkan *player* untuk keluar dari / menutup game.



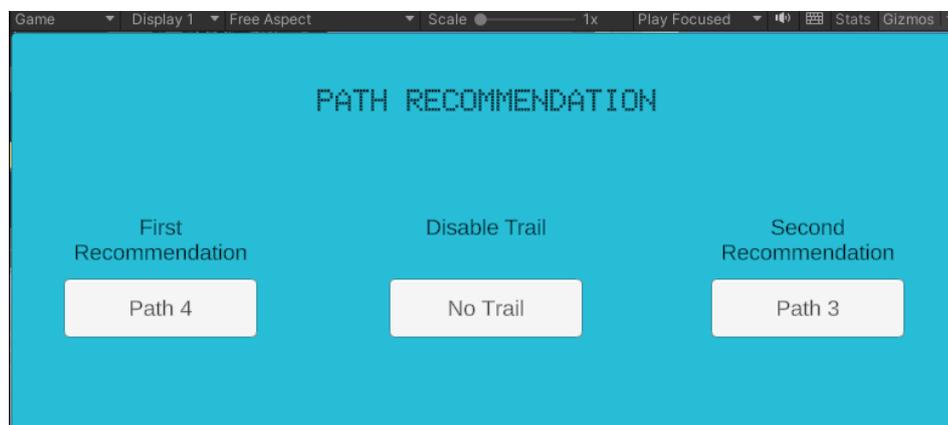
Gambar 4.1 Tampilan *Main Menu*

- b. Tampilan kedua / tampilan yang muncul ketika *player* memilih *play* adalah tampilan *loading screen*. *Loading screen* akan menunjukkan level berapa yang sedang *player* mainkan serta lokasi target *player* dari level tersebut.



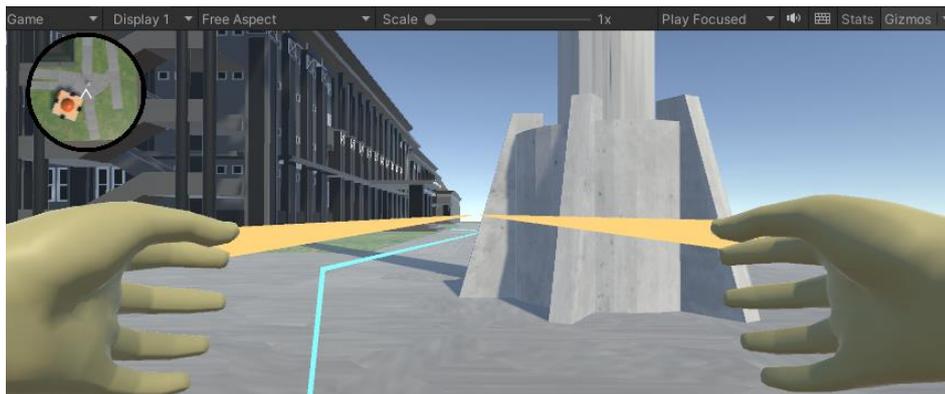
Gambar 4.2 Tampilan *Loading Screen*

- c. Setelah level 1 berhasil di-load, barulah *player* memasuki level 1. Namun sebelum *player* berpetualang, *player* akan diberikan pilihan 3 pilihan terkait *path* yang akan dilalui. 2 pilihan diambil dari 2 nilai tertinggi yang didapatkan oleh sistem rekomendasi yang terhubung dengan sistem. Jika salah satu dari 2 pilihan ini dipilih, trail dari lokasi *player* ke lokasi target akan bersinar sesuai dengan rekomendasi *path* yang dipilih *player*. Pilihan ke-3 adalah pilihan untuk tidak menggunakan sistem rekomendasi. Jika pilihan ini dipilih, maka tidak ada trail yang bersinar dan *player* perlu sampai ke target tujuan tanpa tuntutan arah.



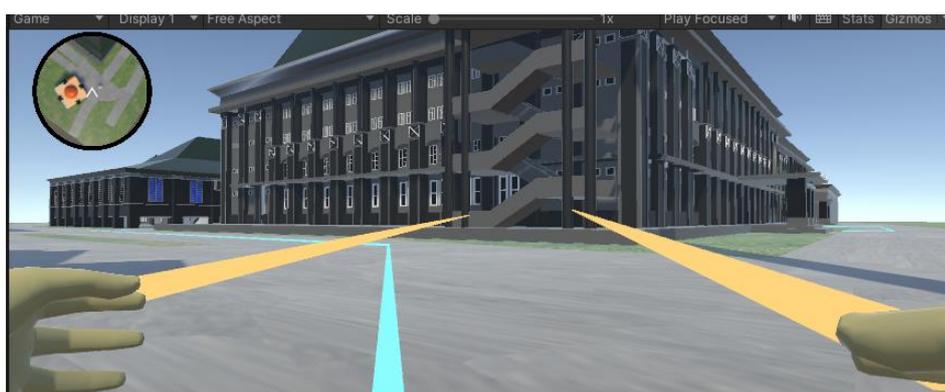
Gambar 4.3 Window Rekomendasi *Path* Yang Diberikan Kepada *Player*

- d. Tampilan berikutnya merupakan tampilan ketika *player* sudah memasuki lingkungan kampus. *Player* bertitik mula di depan parkirannya ma'had laki-laki (di sebelah masjid tarbiyah). *Trail* yang akan menuntun *player* akan bersinar sesuai dengan *path* mana yang *player* pilih di *window* rekomendasi sebelumnya. Berikut ini merupakan 2 contoh sinar *trail* berdasarkan *path* rekomendasi yang dipilih *player*.



Gambar 4.4 Tampilan Jika *Path* 4 Dipilih Dalam *Window* Rekomendasi

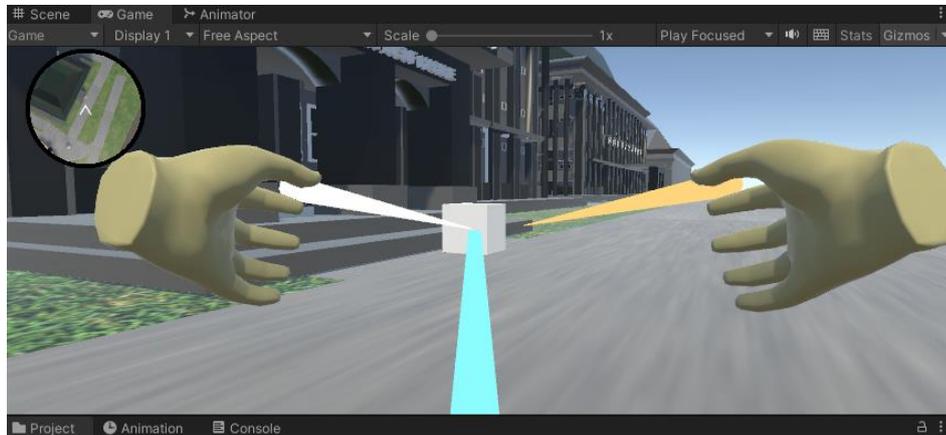
Gambar 4.4 merupakan *trail* yang menyinari *path* 4 ketika *player* memilih *path* 4 dalam *window* rekomendasi. Gambar selanjutnya merupakan *trail* yang menyinari *path* 1 ketika *player* memilih *path* 1 dalam *window* rekomendasi.



Gambar 4.5 Tampilan Jika *Path* 1 Dipilih Dalam *Window* Rekomendasi

Setelah *player* sampai pada titik lokasi target tiap level, *game object* yang dapat diinteraksikan oleh *player* akan menunggu. *Game object* tersebut berfungsi

bagi *player* untuk menjalankan quiz tiap level yang perlu diselesaikan. Berikut ini merupakan lokasi titik target level 1.



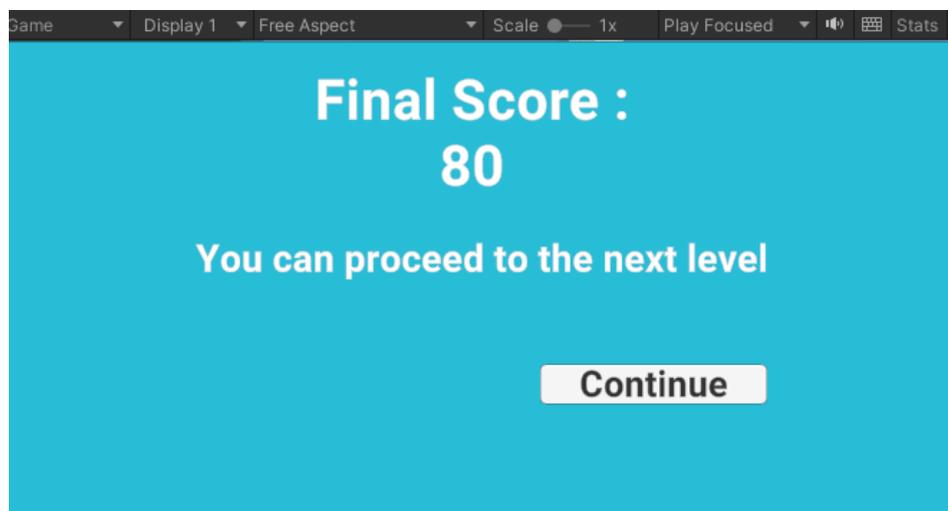
Gambar 4.6 *Game Object Yang Perlu Diinteraksikan Oleh Player*

Gambar di atas menunjukkan *ray* tangan kiri *player* yang bersinar ketika dihadapkan kepada *gameobject* target level. Ketika *player* berinteraksi dengan *game object target*, *window* pertanyaan akan muncul untuk menantang *player* menjawab beberapa pertanyaan yang berkaitan dengan lokasi level yang sedang *player* jalankan. Berikut ini merupakan salah satu contoh pertanyaannya.



Gambar 4.7 Pertanyaan Quiz

Gambar di atas menunjukkan salah satu dari 5 pertanyaan yang akan diberikan kepada *player* untuk menyelesaikan sebuah level. Dari kelima pertanyaan yang diberikan kepada *player*, *player* akan dapat mengakses level berikutnya jika pada level tersebut *player* mendapatkan skor lebih atau sama dengan 80%. Jika skor dari *player* pada quiz lebih sedikit dari 80%, *player* akan diperlukan untuk mengulangi level tersebut dari lokasi awal.



Gambar 4.8 *Player* Menyelesaikan Quiz Dengan Nilai Lebih Atau Sama Dengan 80

#### 4.1.3 Implementasi *Cosine similarity*

Implementasi sistem dilakukan dengan mengikuti persamaan (2.1) menggunakan bahasa C#. Namun sebelum melaksanakan persamaan tersebut, data dari *user* lama perlu di kumpulkan dan disimpan terlebih dahulu. Untuk mengumpulkan datanya, digunakan function `Random()` yang terdapat dalam package System C#. Dari data yang telah *di-generate*, data tersebut disimpan dalam file Microsoft Excel dan nantinya perlu diubah menjadi *scriptable object* agar dapat digunakan. Berikut ini merupakan *pseudo code* untuk mengambil data dari Microsoft Excel dan menyimpannya ke dalam *asset*.

```

method LoadCSV():

    string csvFilePath = Path.Combine(dataFolderPath,
    csvFileName)

    if File.Exists(csvFilePath):

        Log("Data : {csvFileName}: Terbaca")

        string[] lines = File.ReadAllLines(csvFilePath)

        for each line in lines:

            string[] fields = line.Split(',')

            if fields.Length >= 6:
                // Create a new PlayerData object
                PlayerData playerData = new PlayerData()

                TryParse(fields[0], playerData.id)
                playerData.user = fields[1]
                TryParse(fields[2], playerData.path1)
                TryParse(fields[3], playerData.path2)
                TryParse(fields[4], playerData.path3)
                TryParse(fields[5], playerData.path4)

                csvData.playerDataList.Add(playerData)

```

Gambar 4.9 Pseudo Code Untuk Mengambil Data Dari Excel

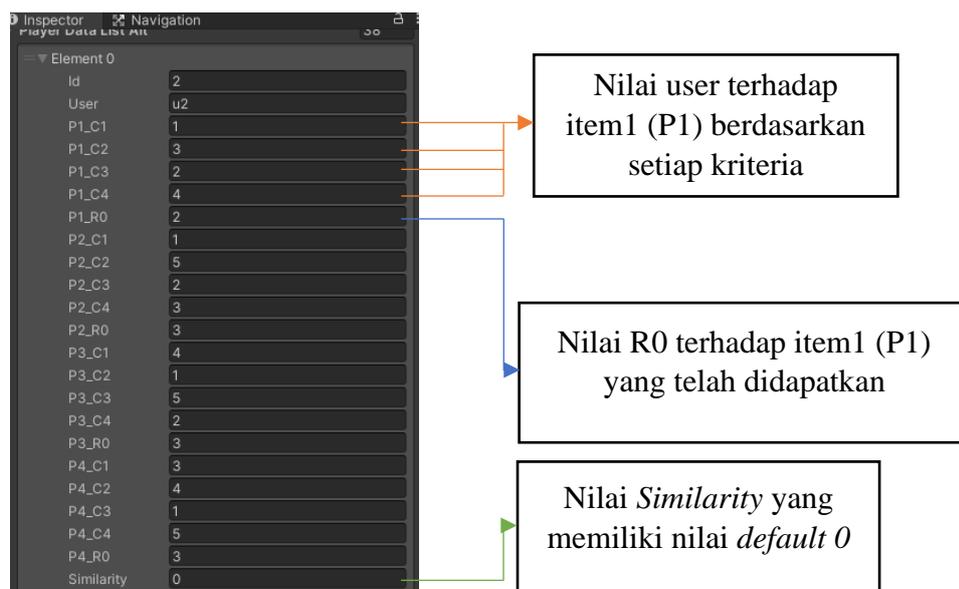
Pseudo code di atas merupakan metode LoadCSV() yang bertanggung jawab untuk membaca data dari file CSV dan memuatnya ke dalam struktur data yang sesuai. Pada awalnya, metode ini mengonstruksi jalur file CSV menggunakan lokasi folder data dan nama file CSV. Jika file ditemukan, metode membaca semua baris dari file, memisahkan setiap baris menjadi *array string* dengan menggunakan koma sebagai pemisah, dan mengonversi nilai-nilai dari setiap kolom ke tipe data yang sesuai. Selanjutnya, sebuah objek PlayerData baru dibuat dan diinisialisasi dengan nilai-nilai dari kolom-kolom yang sesuai. Objek PlayerData tersebut kemudian ditambahkan ke daftar playerDataList yang merupakan bagian dari struktur data yang menyimpan hasil bacaan dari file CSV. Proses ini memastikan bahwa data CSV dimuat dengan benar ke dalam program untuk digunakan lebih

lanjut. Gambar berikut merupakan empat baris pertama dari data file excel yang akan dikonversi menjadi *scriptable object unity*.

Tabel 4.1 Data Mentah Yang Disimpan Dalam Microsoft Excel

I d	U s e r	P1 C 1	P1 C 2	P1 C 3	P1 C 4	P2 C 1	P2 C 2	P2 C 3	P2 C 4	P3 C 1	P3 C 2	P3 C 3	P3 C 4	P4 C 1	P4 C 2	P4 C 3	P4 C 4
2	U 2	1	3	2	4	1	5	2	3	4	1	5	2	3	4	1	5
3	U 3	3	2	4	1	5	2	3	4	1	5	2	3	4	1	5	2
4	U 4	2	4	1	5	2	3	4	1	5	2	3	4	1	5	2	3
5	U 5	4	1	5	2	3	4	1	5	2	3	4	1	5	2	3	4

Tabel di atas merupakan 4 baris data dari dataset yang terdapat dalam sistem dan akan digunakan untuk menghitung nilai prediksi *player*. Dalam data tersebut, terdapat data *rating*-an tiap *user* terhadap item yang ada melalui beberapa kriteria. Untuk dapat digunakan dalam sistem, data di atas perlu diubah menjadi *scriptable object* terlebih dahulu.



Gambar 4.10 Data Excel yang Telah Dikonversi Menjadi *Scriptable Object*

Gambar di atas merupakan data dari tabel 4.1 yang telah dikonversikan menjadi *scriptable object* dalam unity. Dalam *scriptable object* tersebut, nilai R0 terhadap tiap item akan didapatkan dan nilai *similarity* yang memiliki nilai *default* 0 akan dihitung pada langkah berikutnya. Dari data yang sudah dikumpulkan tersebutlah dapat diukur kemiripan dari *player* baru dan *user* terdahulu. Untuk mengukur kemiripan tersebut digunakanlah *cosine similarity* sesuai dengan persamaan (2.1). Pseudo code untuk perhitungan *similarity* antara 2 buah vector dan pencarian *cosine similarity* antara *player* dan *user-user* yang ada adalah sebagai berikut:

```
function CosineSimilarity(vectorA, vectorB):
    // Calculate the dot product
    dotProduct = sum of (elementA * elementB) for each
    corresponding element in vectorA and vectorB

    // Calculate the magnitude of each vector
    magnitudeA = square root of sum of (element^2) for each
    element in vectorA
    magnitudeB = square root of sum of (element^2) for each
    element in vectorB

    // Calculate the cosine similarity
    if magnitudeA > 0 and magnitudeB > 0:
        similarity = dotProduct / (magnitudeA * magnitudeB)
        return similarity
    else:
        return 0 // If one of the vectors has zero magnitude,
        similarity is zero
```

Gambar 4.11 Pseudo Code Perhitungan Cosine Similarity Antara 2 Vektor

Fungsi *Cosine Similarity* menerima dua vektor, *vectorA* dan *vectorB*, dan menghitung kemiripan dari keduanya. Langkah-langkah perhitungan mencakup menghitung hasil perkalian titik (*dot product*) dari elemen-elemen yang sesuai dalam kedua vektor, serta menghitung *magnitude* masing-masing vektor dengan mengakumulasi kuadrat elemen-elemen. Selanjutnya, kemiripannya dihitung dengan membagi *dot product* oleh hasil perkalian *magnitude* keduanya. Fungsi ini

juga perlu memeriksa apakah *magnitude* kedua vektor tidak sama dengan nol sebelum melakukan perhitungan, untuk menghindari pembagian dengan nol. Jika salah satu dari *magnitude* vektor adalah nol, maka fungsi mengembalikan nilai kemiripan nol, mengindikasikan ketidakseragaman atau ketiadaan informasi pada salah satu vektor.

```
function CalculateCosineSimilarities (playerDataList,
referencePlayer):
    vectorCount = length(playerDataList)
    dimensionCount = 4

    // Create vectors for the reference player and other players
    referenceVector = [referencePlayer.path1,
referencePlayer.path2, referencePlayer.path3,
referencePlayer.path4]

    otherVectors = []

    // Populate otherVectors with vectors for other players
    for i = 1 to vectorCount - 1: // Skip the reference player
        otherVector = [playerDataList[i].path1,
playerDataList[i].path2, playerDataList[i].path3,
playerDataList[i].path4]
        add otherVector to otherVectors

    // Calculate cosine similarity for each player
    similarities = []

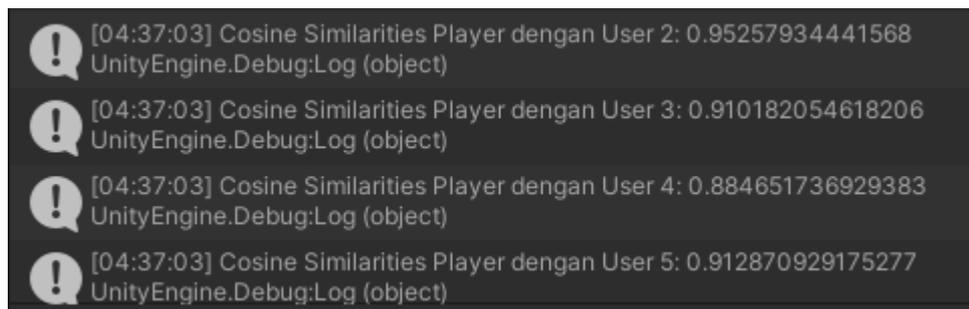
    for each vector in otherVectors:
        similarity = CosineSimilarity(referenceVector, vector)
        add similarity to similarities

    return similarities
```

Gambar 4.12 Pseudo Code Perhitungan Cosine Similarity Dengan Seluruh Data

Fungsi *Calculate Cosine Similarities* menerima daftar pemain (*playerDataList*) dan pemain referensi (*reference layer*), dan menghitung kemiripan kosinus antara pemain referensi dengan setiap pemain lain dalam daftar. Dalam proses ini, vektor referensi dan vektor pemain lainnya dibentuk dengan menggunakan nilai-nilai jalur (*path*) dari masing-masing pemain. Kemudian, untuk setiap pemain, kemiripan kosinus dihitung menggunakan fungsi *Cosine Similarity*

pada gambar 4.11 yang menerima vektor referensi dan vektor pemain tersebut sebagai argumen. Hasil kemiripan kemudian disimpan dalam daftar *similarities*, yang akhirnya dikembalikan sebagai hasil fungsi, memberikan informasi tentang sejauh mana setiap pemain dalam daftar serupa dengan pemain referensi berdasarkan jalur yang diukur.



Gambar 4.13 Perhitungan *Similarity* Antara *Player* Dengan Setiap *User*

Menggunakan fungsi dari gambar 4.12, *similarity* antara *player* dengan setiap *user* yang terdapat dalam dataset-pun dapat dihitung.

#### 4.1.4 Implementasi *Weighted sum*

Setelah kemiripan dari *player* dengan user yang ada diukur baru pemberian *path* pada *player* dapat direkomendasikan. Hal tersebut dilakukan menggunakan *weighted sum* sesuai dengan persamaan (2.2). Berikut ini merupakan *pseudo code* untuk memberikan rekomendasi *path player* menggunakan *weighted sum*:

```

Function CalculateWeightedSum(playerDataList, referencePlayer,
similarities):
    similaritiesCount = Length(similarities) / 5

    numerator = 0
    denominator = 0

    For i = 1 To similaritiesCount - 1:
        pre_numerator = playerDataList[i].path * similarities[i]
        numerator = numerator + pre_numerator
    Next i

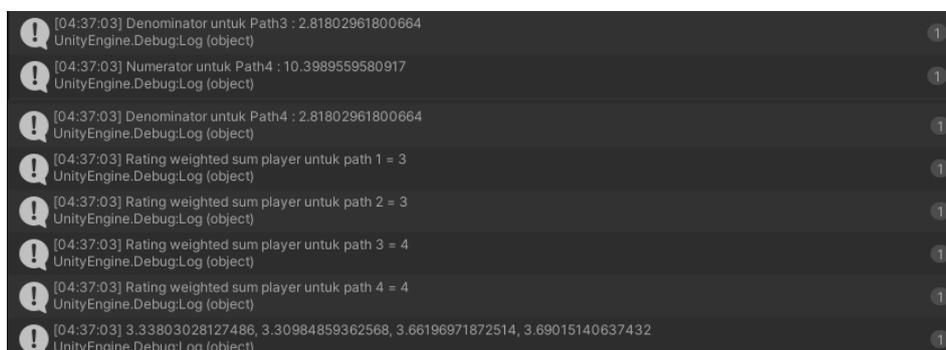
    For i = 1 To similaritiesCount - 1:
        similarity = similarities[i]
        denominator = denominator + similarity
    Next i

    Return numerator / denominator

```

Gambar 4.14 Pseudo Code Perhitungan Nilai Prediksi Menggunakan *Weighted Sum*

Fungsi *Calculate Weighted Sum* menggunakan tiga parameter: *player Data List*, *reference Player*, dan *similarities*. Fungsi ini kemudian menghitung nilai rata-rata tertimbang dari path pada data pemain dalam *player Data List*, di mana bobotnya ditentukan oleh tingkat kemiripan dengan pemain referensi. Proses perhitungan melibatkan iterasi terhadap *similarities* untuk menghitung pembilang dan penyebut, yang akhirnya digunakan untuk menghasilkan nilai rata-rata tertimbang (*weighted*) sebagai hasil fungsi.



```

[04:37:03] Denominator untuk Path3 : 2.81802961800664
UnityEngine.Debug:Log (object)
[04:37:03] Numerator untuk Path4 : 10.3989559580917
UnityEngine.Debug:Log (object)
[04:37:03] Denominator untuk Path4 : 2.81802961800664
UnityEngine.Debug:Log (object)
[04:37:03] Rating weighted sum player untuk path 1 = 3
UnityEngine.Debug:Log (object)
[04:37:03] Rating weighted sum player untuk path 2 = 3
UnityEngine.Debug:Log (object)
[04:37:03] Rating weighted sum player untuk path 3 = 4
UnityEngine.Debug:Log (object)
[04:37:03] Rating weighted sum player untuk path 4 = 4
UnityEngine.Debug:Log (object)
[04:37:03] 3.33803028127486, 3.30984859362568, 3.66196971872514, 3.69015140637432
UnityEngine.Debug:Log (object)

```

Gambar 4.15 Pemberian Nilai Pada *Path* Menggunakan *Weighted Sum*

Menggunakan fungsi dari gambar 4.14, *numerator* dan *denominator player* untuk tiap *path* didapatkan, sehingga nilai prediksi dari *player* terhadap tiap *path* tersebut-pun dapat dihitung dengan cara membulatkan nilai dari hasil pembagian *numerator* oleh *denominator*.

## 4.2 Implementasi Sistem

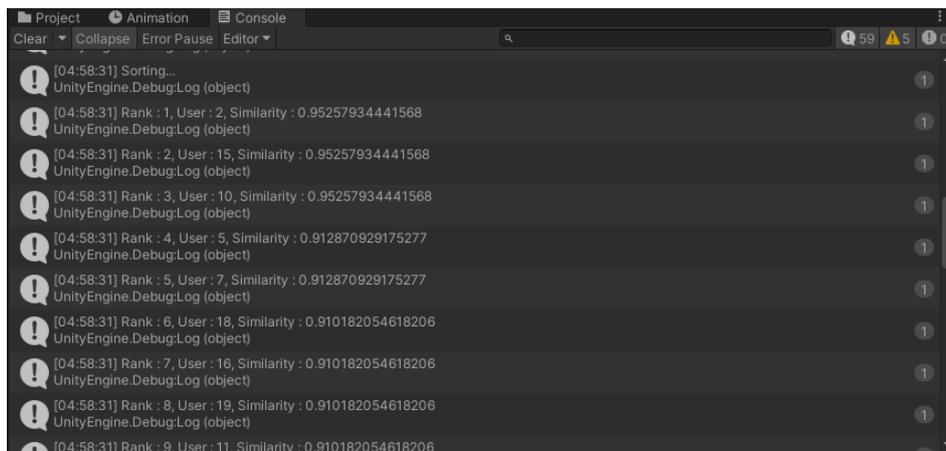
Berikut merupakan proses implementasi dan pengujian sistem *Collaborative Filtering Multi Criteria* untuk rekomendasi scenario *tour* pada Game 3D pengenalan kampus.

### 4.2.1 Proses Implementasi Pada Game

Saat level dijalankan, *player* akan langsung diberikan rekomendasi *path* ke tempat dimana misi berada pada level tersebut. Rekomendasi dari *path* tersebutpun didapatkan dari data yang ada / telah simpan di dalam Microsoft excel. Data yang telah disimpan tersebut akan digunakan dan dikalkulasikan dengan *player* untuk melihat *user* mana yang paling mirip dengan *player*. Simulasi ditujukan kepada *player* yang belum pernah mengelilingi lingkungan kampus, sehingga *player* disengaja untuk tidak dapat memberikan *rating* kepada setiap *path* yang ada. Namun, *similarity*-nya tetap dapat diukur menggunakan matriks yang bersifat netral (semua nilai dalam vektor matriks tersebut sama), yang dimana matriks netral tersebut adalah nilai yang akan diberikan kepada *player*. Langkah pertama yang dilakukan adalah mengambil data yang terdapat dalam Microsoft Excel.

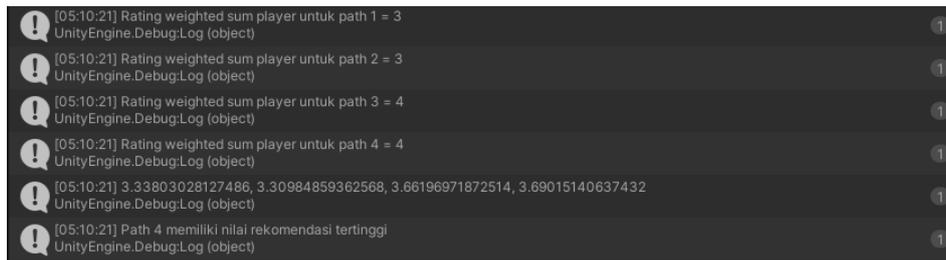
Setelah itu, data dari excel dikonversi menjadi *scriptable object* agar data tersebut dapat digunakan dalam *script unity* nantinya. Setelah itu, *similarity* antara *player* dan semua *user* yang ada diukur dengan menggunakan *cosine similarity*

mengikuti algoritma *pseudo code* yang sebelumnya. Setelah itu, *rating player* terhadap tiap *path* yang ada akan diberikan kembali yang Dimana nilainya dihitung menggunakan pendekatan *weighted sum*. Namun, data *similarity* yang ada perlu diurutkan terlebih dahulu sebelum perhitungan *weighted sum* dapat dilakukan. data yang ada perlu diurutkan terlebih dahulu berdasarkan *similarity*nya seperti berikut:



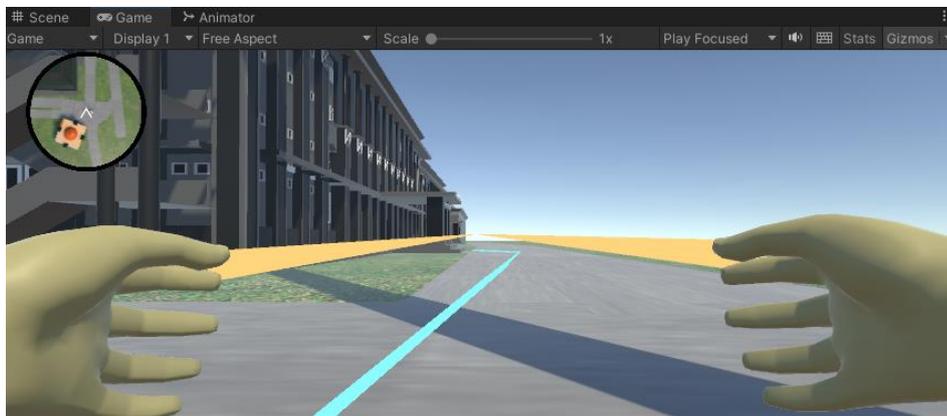
Gambar 4.16 Data Yang Telah Diurutkan Berdasarkan *Similarity*nya

Gambar 4.16 merupakan *similarity* antara *player* terhadap tiap user yang terdapat di dataset yang dimana datanya ditampilkan setelah *similarity*-nya di-*ranking*-kan. Setelah nilai *similarity* antara *player* dan tiap user yang ada di-*ranking*-kan, barulah nilai *rating* baru *player* terhadap tiap *path* dapat dihitung menggunakan *weighted sum*. Nilai *rating player* terhadap setiap *path* dihitung menggunakan *weighted sum* mengikuti yang ada sebelumnya. Dari keempat *path* yang ada, 2 *path* dengan nilai rekomendasi *weighted sum* paling tinggi akan menjadi pilihan direkomendasikan kepada *player* pada level tersebut. Berikut ini adalah contoh pemberian rekomendasi jalur berdasarkan nilai prediksi tertinggi yang diberikan oleh *weighted sum*.



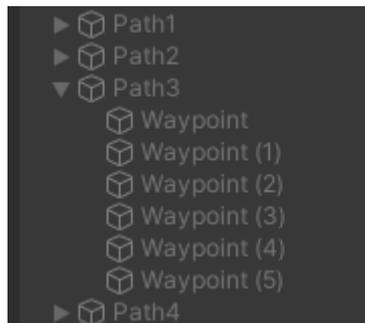
Gambar 4.17 Log Yang Menunjukkan Nilai *Rating* Baru *Player* Terhadap Tiap *Path*

Dalam gambar 4.17, karena *path* 3 dan 4 memiliki nilai rekomendasi tertinggi, maka jalan *path* 3 (bagian belakang perpustakaan sekaligus bagian belakang fakultas sains dan teknologi) dan jalan *path* 4 (bagian depan fakultas saintek) selaku jalan yang dilalui akan dimasukkan ke dalam window rekomendasi yang dimana jika diantara kedua *path* tersebut dipilih oleh *player* nantinya, *path* yang dipilih oleh *player* akan diberikan trail yang akan meng-*highlight*-kan *path* tersebut saat level berjalan.



Gambar 4.18 Tampilan Ketika Rekomendasi *Path* Dipilih

Pada gambar 4.18, peng-*highlight*-an tersebutpun dapat dilakukan dengan penempatan waypoint dalam map dari *player* ke titik tujuan yang dimana trail akan dihubungkan dari *player* ke *waypoint-waypoint* yang ada ke titik tujuan mengikuti *path* rekomendasi yang telah dipilih *player*.



Gambar 4.19 Penempatan *Game Object Waypoint* Dalam Unity

Gambar 4.19 menunjukkan *gameobject waypoint* atau titik yang terdapat pada tiap *path* untuk mencapai titik target tujuan. Setelah *waypoint-waypoint* yang diperlukan ditempatkan, *script trail* yang berfungsi untuk menyinari *path* rekomendasi yang dipilih-pun dapat berjalan. Berikut ini adalah pseudo code untuk function `start()` dan `update()` pada script trail:

```
function Start():
    // Add LineRenderer component
    lineRenderer = AddComponent<LineRenderer>()
    // Generate waypoints along the road's centerline
    GenerateWaypoints()
function Update():
    // Get the closest waypoints and calculate distances
    closestWaypoint = GetClosestWaypoint(transform.position)
    targetWaypoint = GetClosestWaypoint(targetLocation.position)
    distance = Distance(transform.position,
targetLocation.position)
    // Determine the path based on different scenarios
    if targetWaypoint is not null and closestWaypoint is not null
and distance <= closestWaypoint + targetWaypoint
        path = [transform.position, targetLocation.position]
    else if closestWaypoint is not null and targetWaypoint is not
null and closestWaypoint == targetWaypoint:
        path = [transform.position,
closestWaypoint.transform.position, targetLocation.position]
    else if closestWaypoint is not null and targetWaypoint is not
null:
        pathWaypoints = GetWaypointPath(closestWaypoint,
targetWaypoint)
        path = [transform.position] + pathWaypoints +
[targetLocation.position]
    else:
        path = []
    // Set the positions for the LineRenderer
    lineRenderer.positionCount = length(path)
    lineRenderer.SetPositions(path)
    lineRenderer.enabled = true
```

Gambar 4.20 *Pseudo Code* Untuk Menyambungkan *Trail* Dari *Player* Ke Titik Target

Function `start()` berfungsi untuk menginisialisasi *waypoint* yang digunakan, sekaligus mengaktifkan *linerenderer* yang berfungsi sebagai sinar yang menyinari *path* rekomendasi yang dipilih. Function `update()` berfungsi sebagai penghubung antara *player* dengan *waypoint-waypoint* tersebut menggunakan *linerenderer* yang telah diaktifkan dan menyesuaikan penghubungan tersebut berdasarkan *scenario* yang sedang terjadi.

#### 4.2.2 Pengujian Sistem Rekomendasi

Seperti yang telah dijelaskan dalam bab 3, langkah terakhir yang perlu dilakukan adalah untuk mengukur keakuratan dari sistem rekomendasi *collaborative filtering* yang digunakan terhadap rekomendasi *path* dengan pengujian akurasi. Hal tersebut akan dilakukan dengan cara membandingkan data prediksi dan data asli sesuai dengan persamaan (2.3), (2.4), (2.5), dan (2.6). Berikut ini merupakan 10 baris data yang akan digunakan sebagai pengujian menggunakan 10 baris data acuan. Berikut ini adalah perbandingan nilainya:

Tabel 4.2 Pengujian Berdasarkan 10 Data Acuan

User	Nilai Asli		Nilai Rekomendasi	
U2	P1	2	P1	1
	P2	3	P2	2
	P3	4	P3	5
	P4	5	P4	4
U3	P1	3	P1	3
	P2	4	P2	4
	P3	5	P3	4
	P4	1	P4	2
U4	P1	4	P1	4
	P2	5	P2	4
	P3	1	P3	3
	P4	2	P4	1

User	Nilai Asli		Nilai Rekomendasi	
U5	P1	2	P1	1
	P2	1	P2	3
	P3	4	P3	5
	P4	3	P4	4
U6	P1	3	P1	2
	P2	2	P2	2
	P3	1	P3	4
	P4	5	P4	4
U7	P1	4	P1	4
	P2	3	P2	5
	P3	2	P3	2
	P4	1	P4	2
U8	P1	1	P1	3
	P2	5	P2	5
	P3	3	P3	4
	P4	2	P4	2
U9	P1	4	P1	3
	P2	2	P2	3
	P3	5	P3	4
	P4	1	P4	2
U10	P1	3	P1	3
	P2	5	P2	5
	P3	4	P3	3
	P4	2	P4	2
U11	P1	1	P1	2
	P2	3	P2	3
	P3	5	P3	4
	P4	4	P4	4

Tabel di atas merupakan nilai asli dan nilai prediksi dari 10 *user* yang didapatkan oleh sistem. Dari data tabel diatas, lalulah dapat diketahui nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN) dalam *confusion matrix*. Dalam konteks *rating* dalam penelitian ini, nilai 4 dan 5 akan diklasifikasikan sebagai positif dan nilai 1,2 dan 3 kebawah akan dikategorikan sebagai *negative*. Selain itu, nilai prediksi tidak akan digolongkan *true* jika memiliki perbedaan lebih dari 1 dari nilai asli. Berikut ini adalah tabel *confusion matrix*-nya:

Tabel 4.3 *Confusion Matrix*

	Asli Positif	Asli Negatif
Prediksi Positif	14	6
Prediksi Negatif	3	17

Setelah tabel *confusion matrix* didapatkan, maka nilai akurasi presisi, *recall* dan *F-Score* dapat dihitung.

- Akurasi

$$Accuracy = \frac{(14 + 17)}{(14 + 17 + 6 + 3)} = 0,775$$

- Presisi

$$Precision = \frac{14}{(14 + 6)} = 0,7$$

- *Recall*

$$Recall = \frac{14}{(14 + 3)} = 0,823$$

- *F-Score*

$$FScore = 2 * \frac{(0,7 * 0,823)}{(0,7 + 0,823)} = 0,755$$

Tabel 4.4 Tabel Perhitungan *Confusion Matrix* Berdasarkan 10 Baris Data

Akurasi	0,775
Presisi	0,7
<i>Recall</i>	0,823
<i>F-Score</i>	0,755

Pengujian metode untuk 10 baris data yang ada dapat dilihat di tabel di atas. Berdasarkan hasil yang ditunjukkan oleh Tabel 4.4, diperoleh hasil akurasi sebesar

77,5% dan *F-Score* sebesar 75,5%. Sebagai perbandingan, akurasi, presisi, *recall*, dan *F-Score* akan dihitung kembali dalam log unity berdasarkan 20 baris data dan 40 baris data. Berikut ini adalah hasilnya untuk perhitungan berdasarkan 20 baris data:

Tabel 4.5 Tabel Perhitungan Confusion Matrix Berdasarkan 20 Baris Data

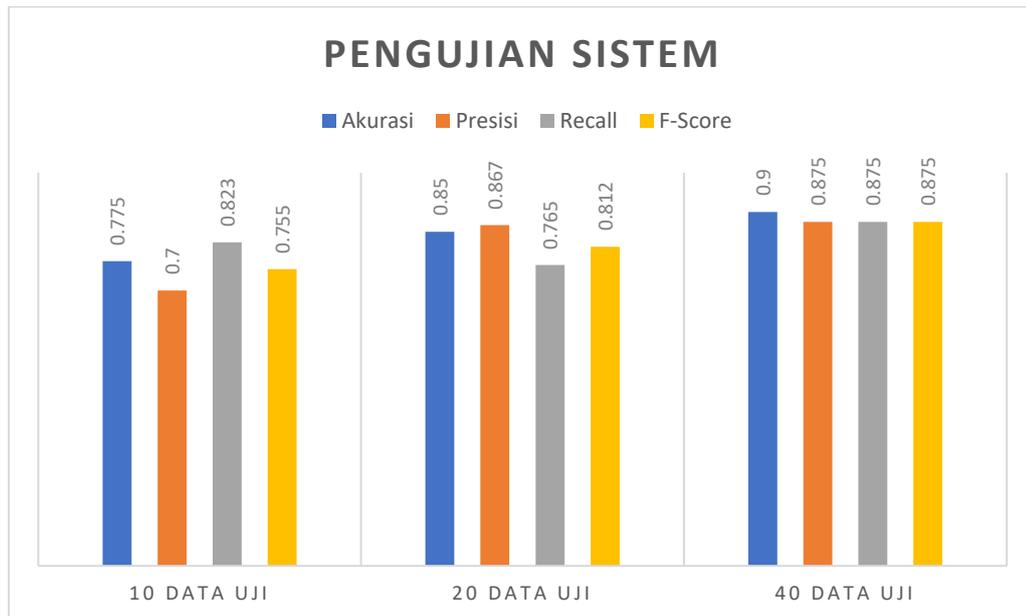
Akurasi	0,85
Presisi	0,867
<i>Recall</i>	0,765
<i>F-Score</i>	0,812

Pengujian metode untuk 20 baris data yang ada dapat dilihat di tabel 4.4 di atas. Berdasarkan hasil yang ditunjukkan oleh Tabel 4.4, diperoleh hasil akurasi sebesar 85% dan *F-Score* sebesar 81,2%. Setelah itu, berikut ini adalah hasilnya untuk perhitungan berdasarkan 40 baris data:

Tabel 4.6 Tabel Perhitungan Confusion Matrix Berdasarkan 40 Baris Data

Akurasi	0,9
Presisi	0,875
<i>Recall</i>	0,875
<i>F-Score</i>	0,875

Pengujian metode untuk 40 baris data yang ada dapat dilihat di tabel di atas. Berdasarkan hasil yang ditunjukkan oleh Tabel 4.6, diperoleh hasil akurasi sebesar 90% dan *F-Score* sebesar 87,5%. Perbandingan antara tabel 4.3, tabel 4.4 dan tabel 4.5 dapat dilihat pada grafik di bawah ini:



Gambar 4.21 Pengujian Sistem Rekomendasi Berdasarkan 10, 20 Dan 40 Data Uji

Dari ketiga percobaan pada gambar 4.21, dengan jumlah data uji yang berbeda-beda tersebut, dapat disimpulkan bahwa nilai *F-Score* dan akurasi berbanding lurus dengan jumlah data yang dimiliki, semakin banyak data uji yang digunakan, maka semakin tinggi pula akurasi dan *F-Score*-nya.

### 4.3 Integrasi Islam

Adanya kemajuan teknologi memungkinkan manusia untuk belajar menggunakan media-media yang bervariasi. Belajar merupakan salah satu aspek hidup yang sangat penting bagi umat manusia, khususnya bagi umat muslim. Dengan adanya ilmu, manusia dapat mengetahui apa yang sebelumnya diketahui dan dapat menjadi orang yang lebih bijaksana. Pada surah an-Nahl ayat ke-78 menceritakan tentang potensi pada diri manusia yang harus digunakan dalam kegiatan belajar dan pembelajaran:

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ وَالْأَفْئِدَةَ ۗ لَعَلَّكُمْ تَشْكُرُونَ

"Dan Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatupun, dan dia memberi kamu pendengaran, penglihatan dan hati, agar kamu bersyukur". (QS. An-Nahl: 78)

Ayat di atas menunjukkan bahwa ada tiga potensi yang terlibat dalam proses pembelajaran: *as-Sam'u* (mendengar), *al-Bashar* (melihat), dan *Fu'ad* (qalbu) (Wakka, 2020). Dalam Hadis Riwayat Imam Ahmad, Rasulullah S.A.W. bersabda:

مَنْ أَرَادَ الدُّنْيَا فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَ الْآخِرَةَ فَعَلَيْهِ بِالْعِلْمِ، وَمَنْ أَرَادَهُمَا فَعَلَيْهِ بِالْعِلْمِ

"Barang siapa yang hendak menginginkan dunia, maka hendaklah ia menguasai ilmu. Barang siapa menginginkan akhirat, hendaklah ia menguasai ilmu. Dan barang siapa yang menginginkan keduanya (dunia dan akhirat), hendaklah ia menguasai ilmu." (HR Ahmad)

Ada banyak cara bagi seseorang untuk mendapatkan ilmu. Berbanding lurus dengan perkembangan teknologi yang terus berjalan, salah satu dari cara untuk mempelajari sesuatu tersebut adalah dengan game, khususnya game simulasi. Simulasi adalah metode pelatihan yang meragakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya, yang dimana dalam konteks ini penjelajahan lingkungan kampus akan divirtualisasikan.

Selain itu dari Abu Hurairah, Rasulullah S.A.W. bersabda "Barangsiapa yang ditanya tentang sesuatu yang tidak diketahuinya, kemudian dia menutup mulutnya, maka itu adalah bagian dari ilmu (kebijaksanaan) baginya (HR. Tirmidzi)". Konsep yang terkandung dalam hadis tersebut berhubungan dengan ide-ide yang relevan dalam konteks teknologi, khususnya sistem rekomendasi. Salah

satu hubungannya adalah betapa pentingnya berbagi pengetahuan dan memberikan informasi kepada orang yang tidak mengetahui. Dalam konteks sistem rekomendasi, ide ini dapat dihubungkan dengan konsep berbagi informasi atau rekomendasi kepada mereka yang membutuhkannya. Hadis tersebut juga menyiratkan bahwa ilmu harus terus berkembang dan disebar. Dalam sistem rekomendasi, ide ini dapat dihubungkan dengan konsep pembaharuan dan peningkatan algoritma rekomendasi untuk menjaga keberlanjutan dalam memberikan rekomendasi yang relevan. Surah Al-Anbiya ayat 7 juga menyentuh tentang ketidaktahuan seperti berikut:

وَمَا أَرْسَلْنَا قَبْلَكَ إِلَّا رِجَالًا نُوْحِي إِلَيْهِمْ فَسَلُّوا أَهْلَ الذِّكْرِ إِنْ كُنْتُمْ لَا تَعْلَمُونَ

*“Dan Kami tidak mengutus (rasul-rasul) sebelum engkau (Muhammad), melainkan beberapa orang laki-laki yang Kami beri wahyu kepada mereka, maka tanyakanlah kepada orang yang berilmu, jika kamu tidak mengetahui.” (QS. Al-Anbiya: 7)*

Dari ayat tersebut dapat ditafsirkan dalam pentingnya konsultasi dan bertanya kepada orang yang berilmu jika seseorang tidak mengetahui atau memahami sesuatu, sama halnya seperti meminta rekomendasi dalam sistem rekomendasi. Selain itu, sebagaimana ayat tersebut menunjukkan bahwa Allah mengutus rasul-rasul yang diberi wahyu, sistem rekomendasi juga harus didasarkan pada sumber daya yang "berilmu" untuk memberikan rekomendasi yang akurat dan relevan. Surah Al-Ma'idah ayat 16 memiliki isi sebagai berikut:

يَهْدِي بِهِ اللَّهُ مَنِ اتَّبَعَ رِضْوَانَهُ سُبُلَ السَّلَامِ وَيُخْرِجُهُم مِّنَ الظُّلُمَاتِ إِلَى النُّورِ بِإِذْنِهِ وَيَهْدِيهِمْ إِلَى صِرَاطٍ مُسْتَقِيمٍ

*“Dengan kitab itulah Allah menunjuki orang-orang yang mengikuti keridhaan-Nya ke jalan keselamatan, dan (dengan kitab itu pula) Allah mengeluarkan orang-orang itu dari gelap gulita kepada cahaya yang terang benderang dengan seizin-Nya, dan menunjuki mereka ke jalan yang lurus.” (QS. Al-Maidah: 16)*

Ayat tersebut menjelaskan tentang Allah yang memberi petunjuk dengan kitab yang nyata kepada orang-orang yang mengikuti keridhaan Allah ke jalan-jalan keamanan dan keselamatan, dan mengeluarkan mereka dengan izinnya dari kegelapan-kegelapan kekafiran menuju cahaya keimanan, dan memberikan taufik bagi mereka menuju agamanya yang lurus. Dalam konteks pemberian rekomendasi *path*, sistem juga mengikuti esensi ayat tersebut yakni untuk membantu memberikan petunjuk ke jalan untuk sampai ke tujuan kepada *player* yang menggunakannya. Dari Abu Hurairah, Rasulullah S.A.W. bersabda:

مَنْ دَعَا إِلَى هُدًى، كَانَ لَهُ مِنَ الْأَجْرِ مِثْلُ أُجُورِ مَنْ تَبِعَهُ، لَا يَنْقُصُ ذَلِكَ مِنْ أُجُورِهِمْ شَيْئًا

*"Siapa yang mengajak kepada petunjuk, maka dia mendapatkan pahala seperti pahala orang yang mengikutinya, tanpa mengurangi pahalanya sedikitpun."* (HR. Muslim no. 2674, dari Abu Hurairah)

Dalam konteks ini, memberikan petunjuk kepada ilmu yang bermanfaat dapat diartikan sebagai memberikan saran atau rekomendasi kepada orang lain tentang sumber pengetahuan atau sebuah informasi. Dengan memberikan rekomendasi yang sesuai, seseorang dapat berkontribusi pada pertumbuhan pengetahuan dan kebaikan umum.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Seperti yang dijelaskan dalam hasil dan pembahasan pada penelitian, rekomendasi skenario *tour* dipasang pada game pengenalan kampus dari lingkungan ma'had hingga ke gedung sains dan teknologi dengan jumlah 4 *path*. Pemberian rekomendasi dilakukan dengan menggunakan Multi Criteria Recommender System (MCRS) dengan pendekatan *cosine similarity* untuk mencari kemiripan antar user dan *weighted sum* untuk pemberian nilai prediksi. Rekomendasi skenario *tour* diperoleh dari dataset yang ada yang dibandingkan dengan data user. Terdapat 4 *path* yang dapat diberikan nilai yang masing-masing *path* memiliki 4 kriteria sebagai penilaiannya. Nilai dari data *rating* berkisar dari 1 hingga 5. Dataset yang ada disimpan dalam file .csv dalam Microsoft Excel yang dimana data tersebut akan dikonversi menjadi *scriptable object* dalam unity. Dua nilai tertinggi dari perhitungan nilai prediksi akan dikembalikan kepada *player* untuk dipilih sebagai *path* yang akan diikuti. Pengujian sistem rekomendasi dilakukan dengan membandingkan perhitungan nilai prediksi pada sistem dengan data asli user yang sudah ada. Dengan menggunakan 10 data uji, diperoleh akurasi sebesar akurasi secara sebesar 77,5% dan F-Score sebesar 75,5%. Dengan menggunakan 20 data uji, diperoleh akurasi sebesar akurasi secara sebesar 85% dan F-Score sebesar 81,2%. Dengan menggunakan 40 data uji, diperoleh akurasi sebesar akurasi secara sebesar 90% dan F-Score sebesar 87,5%. Dari ketiga percobaan dengan jumlah data uji yang berbeda-beda tersebut, dapat disimpulkan

bahwa nilai F-Score dan akurasi berbanding lurus dengan jumlah data yang dimiliki, semakin banyak data uji yang digunakan dalam *Multi Criteria Recommender System* (MCRS) dengan pendekatan *cosine similarity* untuk penentu kemiripan dan pendekatan *weighted sum* untuk penentu nilainya, maka semakin tinggi pula akurasi dan F-Score-nya.

## 5.2 Saran

Saran untuk pengembangan penelitian lebih lanjut untuk menghasilkan penelitian yang lebih baik, diantaranya adalah sebagai berikut:

1. Menggunakan metode untuk mencari nilai prediksi yang lain ataupun menggunakan metode yang sama namun pendekatan yang lain untuk dijadikan sebagai bandingan untuk mencari hasil yang lebih baik.
2. Menambahkan jumlah data pengujian untuk mendapatkan nilai yang lebih baik.
3. Menambahkan kriteria-kriteria lain pada sistem yang nantinya dapat menggambarkan selera pengguna dengan lebih baik.

## DAFTAR PUSTAKA

- Aisha, D. (2022). *Sistem Rekomendasi Toko Online menggunakan Algoritma Collaborative Filtering dan Content Based Filtering*.
- Anggraini, A. N., Fadila, J. N., & Nugroho, F. (2021). Rancang Bangun Game 2D “Finding Tajwid” Dengan Metode Finite State Machine Menggunakan Software Unity Hub. *Jurnal Teknologi Informasi*, 5(1), 88–93. <https://doi.org/10.36294/jurti.v5i1.1782>
- Arif, Y. M., Nurhayati, H., Nugroho, S. M. S., & Hariadi, M. (2022). Destinations Ratings Based Multi-Criteria Recommender System for Indonesian Halal Tourism Game. *International Journal of Intelligent Engineering and Systems*, 15(1), 282–294. <https://doi.org/10.22266/IJIES2022.0228.26>
- Elmqaddem, N. (2019). Augmented Reality and Virtual Reality in education. Myth or reality? *International Journal of Emerging Technologies in Learning*, 14(3), 234–242. <https://doi.org/10.3991/ijet.v14i03.9289>
- Haruna, K., Ismail, M. A., Damiasih, D., Sutopo, J., & Herawan, T. (2017). A collaborative approach for research paper recommender system. *PLoS ONE*, 12(10), 1–17. <https://doi.org/10.1371/journal.pone.0184516>
- Intan Hervianda Putri, Siti Muslimah K. H. Nurakhmadyavi, & Erwin Eko Wahyudi. (2022). Sistem Rekomendasi untuk Buku dan Film. *Seminar Nasional Teknik Elektro, Sistem Informasi, Dan Teknik Informatika*, 75–81. <https://ejurnal.itats.ac.id/snestikdanhttps://snestik.itats.ac.id>
- Janan, A. (2020). Sistem Rekomendasi Pada Game Wisata Di Kota Batu Menggunakan Metode User Based Collaborative Filtering. *Skripsi. Sains Dan Teknologi. Teknik Informatika. Universitas Islam Negeri Maulana Malik Ibrahim. Malang*.
- Ma’ruf, M. A., & Qoiriah, A. (2022). Perbandingan Algoritma Cosine Similarity dan Euclidean Distance pada Sistem Rekomendasi Film dengan Metode Item-Based Collaborative Filtering. *Journal of Informatics and Computer Science (JINACS)*, 4(2), 160–168.
- Mambu, J. Y., Wahyudi, A. K., & Hezky, G. (2019). “Jump”: Game Simulasi Olahraga Berbasis Virtual Reality Dengan Sensor Accelerometer. *Nutrix Journal*, 3(2), 58. <https://doi.org/10.37771/nj.vol3.iss2.408>
- Mondi, R. H., Wijayanto, A., & Winarno. (2019). Recommendation System With Content-Based Filtering Method for Culinary Tourism in Mangan Application. *Jurnal Ilmiah Teknologi Dan Informasi*, 8(2), 65–72. <https://jurnal.uns.ac.id/itsmart/article/download/35008/27748>
- Nassar, N., Jafar, A., & Rahhal, Y. (2020). Multi-criteria collaborative filtering recommender by fusing deep neural network and matrix factorization. In

*Journal of Big Data* (Vol. 7, Issue 1). <https://doi.org/10.1186/s40537-020-00309-6>

- Nilashi, M., Esfahani, M. D., Roudbaraki, M. Z., Ramayah, T., & Ibrahim, O. (2016). A multi-criteria collaborative filtering recommender system using clustering and regression techniques. *Journal of Soft Computing and Decision Support Systems*, 3(5), 24–30.
- Nugroho, F., & Ismu Rahayu, M. (2020). Sistem Rekomendasi Produk Ukm Di Kota Bandung Menggunakan Algoritma Collaborative Filtering. *Jurnal Riset Sistem Informasi Dan Teknologi Informasi (JURSISTEKNI)*, 2(3), 23–31. <https://doi.org/10.52005/jursistekni.v2i3.63>
- Rogers, S. (2014). *Level Up! The Guide to Great Video Game Design* (2nd Editio). Wiley; 2nd edition (April 28, 2014). <https://www.amazon.com/Level-Guide-Great-Video-Design/dp/1118877160>
- Rokhim, A., & Saikhu, A. (2016). Sistem Rekomendasi Buku Pada Aplikasi Perpustakaan Menggunakan Metode Collaborative Filtering Pada Smkn 1 Bangil. *Spirit*, 8(2), 43–46. <http://download.garuda.kemdikbud.go.id/>
- Saurik, H. T. T., Purwanto, D. D., & Hadikusuma, J. I. (2019). Teknologi Virtual Reality untuk Media Informasi Kampus. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(1), 71. <https://doi.org/10.25126/jtiik.2019611238>
- Virgantara Putra, O., Reza Pradhana, F., & Shaleha, F. (2022). *Simulasi 3 Dimensi Rute Evakuasi Bencana Di Gontor Putri Kampus 3 Berbasis Virtual Reality (Vr)*. 1, 435–444.
- Wakka, A. (2020). Petunjuk Al-Qur'an Tentang Belajar Dan Pembelajaran (Pembahasan Materi, Metode, Media dan Teknologi Pembelajaran). *Education and Learning Journal*, 1(1), 83–84.
- Wijaya, A. E., & Alfian, D. (2018). Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering. *Jurnal Computech & Bisnis*, 12(1), 11–27.
- Yusmar, A., Wardhani, L. K., & Suseno, H. B. (2021). Restaurant Recommender System Using Item Based Collaborative Filtering and Adjusted Cosine Algorithm Similarity. *Jurnal Teknik Informatika*, 14(1), 93–100. <https://doi.org/10.15408/jti.v14i1.21102>