

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Magang ini dilaksanakan dengan supervisi dari Dr. Ir. Winarno, M.Kom. yang membimbing dengan memberikan masukan terhadap progress, diskusi mengenai keperluan proyek, memberi dan menjawab pertanyaan mengenai proses magang yang dilaksanakan. Pekerjaan dilaksanakan dengan tanggung jawab sebagai seorang *Unity Developer* atas setiap aspek pembangunan aplikasi. Pekerjaan yang dilaksanakan merupakan tugas yang berbentuk perancangan, mengimplementasikan, dan melakukan pengujian terhadap aplikasi yang dikembangkan selama masa magang.

#### 3.2 Tugas yang Dilakukan

Seiringnya pelaksanaan magang ini, terdapat beberapa tugas yang diberikan oleh supervisor untuk diselesaikan. Beberapa tugas yang diberikan oleh supervisor adalah sebagai berikut:

Tabel 3.1. Pekerjaan yang ditugaskan selama masa magang

Tugas No -	Pekerjaan yang ditugaskan
1	Memahami dampak pameran <i>virtual</i>
2	Memahami cara kerja <i>Virtual Reality</i> di dalam <i>Engine Unity</i>
3	Menerapkan fungsi interaksi <i>Virtual Reality</i> ke dalam pameran
4	Menerapkan perumahan <i>virtual</i> sebagai tempat pameran
5	Implementasi fungsi interaksi pada furnitur rumah
6	Implementasi UI/UX untuk interaksi <i>user</i> dengan <i>chatbot AI</i>
7	Implementasi <i>chatbot AI</i> untuk menjelaskan detail pameran kepada <i>user</i>

Tugas yang berbentuk pemrograman dilakukan menggunakan bahasa pemrograman *CSharp (C#)*[10]. Proyek akan dikembangkan dalam bentuk 3D dan 2D. Dimana UI akan berbentuk 2D dan seluruh pameran akan berbentuk 3D.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kegiatan magang berlangsung selama lima bulan, terhitung sejak tanggal 3 Februari 2025 hingga 11 Juni 2025. Meskipun periode magang yang direncanakan adalah selama empat bulan, adanya sejumlah hari libur nasional menyebabkan jadwal kegiatan mengalami penyesuaian, sehingga durasi pelaksanaan diperpanjang menjadi lima bulan agar memenuhi ketentuan kontrak dan jam kerja.

Berikut adalah uraian pelaksanaan kerja magang:



Tabel 3.2. Pekerjaan yang dikerjakan selama masa magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Diskusi job desk, studi banding pameran digital & fisik, belajar konsep <i>Generative AI</i> , dan persiapan project <i>Unity</i> .
2	Peminjaman VR, penyesuaian project lama ke <i>Unity</i> , dan pembelajaran tentang kerja VR.
3	Mencari aset perumahan, pengujian VR, dan integrasi VR ke project.
4	Penyesuaian VR dengan <i>Unity</i> , playtesting, dan bug fixing awal.
5	Bug fixing dan pembuatan fungsi interaksi objek dalam <i>Unity</i> .
6	Lanjutan pembuatan fungsi interaksi dan pencarian aset.
7	Fungsi interaksi lanjutan dan bug fixing.
8	Mulai membuat map perumahan dan mencari aset rumah.
9	Libur Nasional
10	Fungsi interaksi & bug fixing, pencarian serta penambahan aset ke pameran.
11	Lanjutan pembuatan fungsi interaksi, diskusi aplikasi, dan bug fixing.
12	Bug fixing lanjutan dan pencarian aset perumahan.
13	Mulai implementasi <i>chatbot AI</i> ke <i>Unity</i> dan testing.
14	Bug fixing dan pengembangan UI untuk project VR.
15	Implementasi dan penyesuaian <i>chatbot AI</i> dan UI dengan VR.
16	Penyesuaian UI & VR, pencarian aset kecil, bug fixing.
17	Implementasi dan penyesuaian integrasi AI dengan UI.
18	Pertemuan supervisor dan bug fixing lanjutan pada AI & UI.
19	Merapikan dan finalisasi proyek.

Pekerjaan ini dilakukan menggunakan Laptop pribadi dan VR yang dipinjam dari *Lab 3D Game Development* yang terdapat pada *Gedung B Lt 5, Ruang 510* di kampus *Universitas Multimedia Nusantara*.

### 3.3.1 Mempelajari *Virtual Reality* dan *Generative AI* menggunakan *Unity 3D*

Selama pelaksanaan magang ini, pengembangan aplikasi pameran dilakukan menggunakan game engine *Unity 3D* sebagai platform utama. *Unity* merupakan sebuah perangkat lunak pengembangan lintas platform yang banyak digunakan untuk membuat aplikasi interaktif, terutama game dan pengalaman berbasis VR, karena kemampuannya dalam menghasilkan visualisasi 3D yang realistis serta dukungan terhadap berbagai perangkat dan sistem operasi.[11]

Pada minggu pertama pelaksanaan magang, sebagian besar waktu difokuskan untuk mempelajari dasar-dasar teknologi yang akan digunakan dalam pengembangan proyek, yaitu teknologi *Virtual Reality* dan konsep *Generative AI*. Kegiatan diawali dengan diskusi bersama supervisor guna memperoleh pemahaman yang lebih jelas terkait ruang lingkup pekerjaan, tujuan proyek, serta pembagian tugas yang akan dijalankan selama masa magang. Diskusi ini juga menjadi dasar dalam menyusun rencana kerja yang terstruktur dan sesuai dengan kebutuhan proyek.

Setelah itu, dilakukan studi banding terhadap berbagai bentuk pameran digital dan fisik yang telah ada sebelumnya, baik dalam maupun luar negeri, untuk memperoleh referensi dan inspirasi mengenai bagaimana sebuah pameran dapat dikemas secara menarik dan interaktif. Studi ini menjadi penting dalam memahami kelebihan dan kekurangan dari masing-masing pendekatan, serta bagaimana integrasi teknologi dapat meningkatkan kualitas pengalaman pengguna.

Selain itu, kegiatan magang pada minggu pertama juga difokuskan untuk memperdalam pemahaman mengenai cara kerja teknologi VR dalam konteks pengembangan aplikasi pameran, termasuk bagaimana VR dapat digunakan untuk menciptakan lingkungan virtual yang imersif dan menarik. Di sisi lain, juga dilakukan eksplorasi awal terhadap konsep *Generative AI*, khususnya dalam hal bagaimana teknologi ini dapat digunakan untuk memperkaya konten pameran secara dinamis dan kontekstual. Seluruh proses pembelajaran dan eksplorasi ini dilakukan sebagai fondasi awal dalam merancang dan mempersiapkan proyek pameran perumahan berbasis VR yang akan dikembangkan secara bertahap selama masa magang berlangsung.

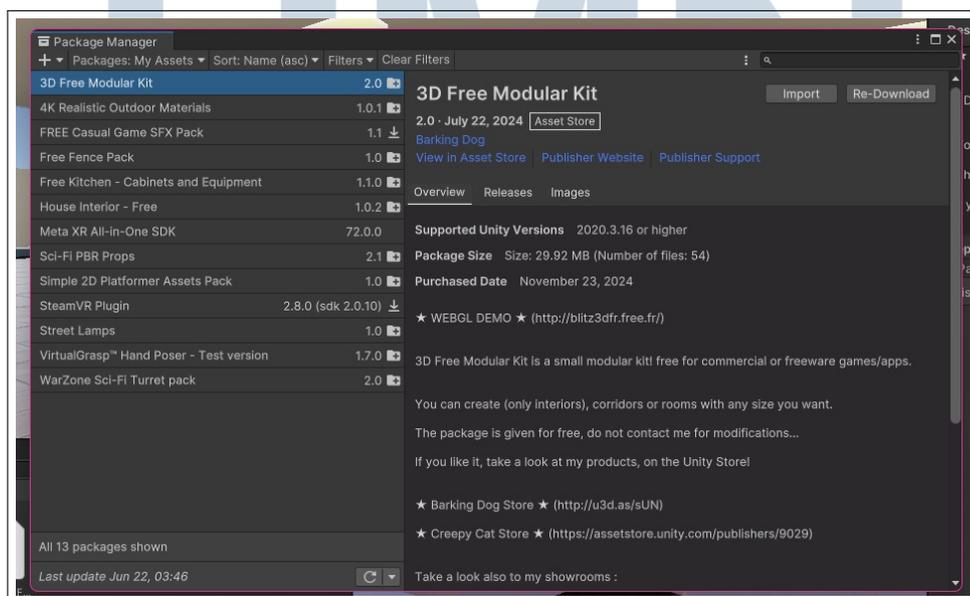
### 3.3.2 Pengembangan Aplikasi

Selama proses perancangan dalam kegiatan magang ini, dilakukan penambahan fitur pada aplikasi yang telah ada sebelumnya untuk menambahkan tipe pameran baru. Proses pengembangan ini melalui beberapa tahap iteratif yang disesuaikan dengan perubahan pada rancangan aplikasi. Selama masa magang, pekerjaan dilakukan secara bertahap pada berbagai komponen aplikasi yang berbeda sesuai kebutuhan pengembangan.

#### A Inkorporasi Aset 3D

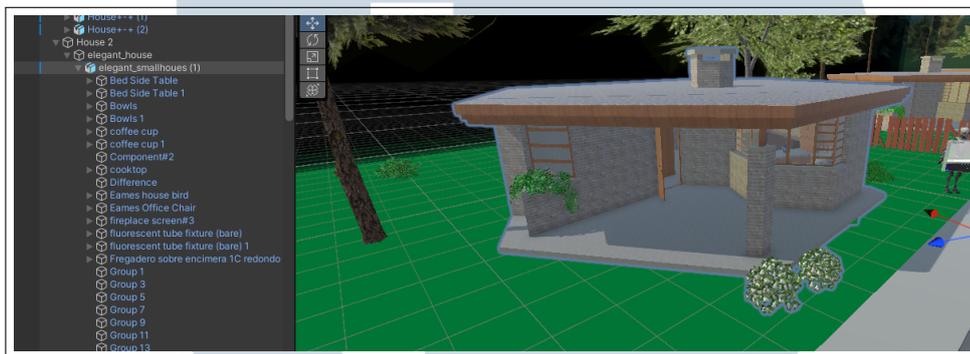
Aset-aset 3D yang digunakan dalam proyek ini diperoleh melalui berbagai platform daring seperti *Unity Asset Store*, *Sketchfab*, dan *TurboSquid*. Situs-situs tersebut menyediakan beragam aset 3D yang dapat dimanfaatkan untuk keperluan pengembangan aplikasi, khususnya dalam mendukung proses pembuatan konten visual yang lebih realistis dan interaktif.

Aset-aset yang diunduh difokuskan pada elemen-elemen yang relevan dengan tema perumahan, terutama model 3D rumah. Model-model ini digunakan sebagai komponen utama dalam pameran virtual yang dikembangkan, guna menciptakan representasi visual yang sesuai dengan tujuan proyek serta memberikan pengalaman pengguna yang lebih imersif. Semua aset yang ditemukan akan di import ke dalam proyek dan akan digunakan untuk fitur-fitur lainnya.



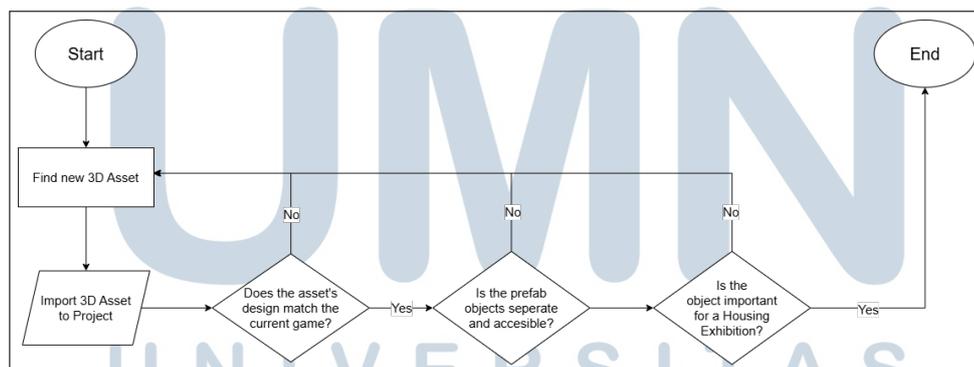
Gambar 3.1. Package Manager untuk Import Asset

Proses pemilihan aset 3D yang akan digunakan dalam proyek dilakukan melalui tahap evaluasi singkat namun cermat untuk memastikan bahwa setiap objek yang dipilih memenuhi kriteria teknis dan visual yang dibutuhkan. Beberapa aspek yang menjadi pertimbangan dalam proses ini antara lain adalah aksesibilitas isi prefab dari objek, kelengkapan struktur model (seperti adanya bagian interior dan eksterior), serta kesesuaian desain objek dengan tema dan kebutuhan pameran virtual yang sedang dikembangkan.



Gambar 3.2. Contoh Aset 3D yang sesuai

Jika aset yang ditemukan tidak sesuai secara teknis atau desain, maka ada memulai pencarian aset baru yang lebih sesuai. Alur dari proses pencarian dan evaluasi untuk aset 3D baru bisa terlihat pada Gambar 3.3.



Gambar 3.3. Flowchart Pencarian dan Evaluasi Aset 3D

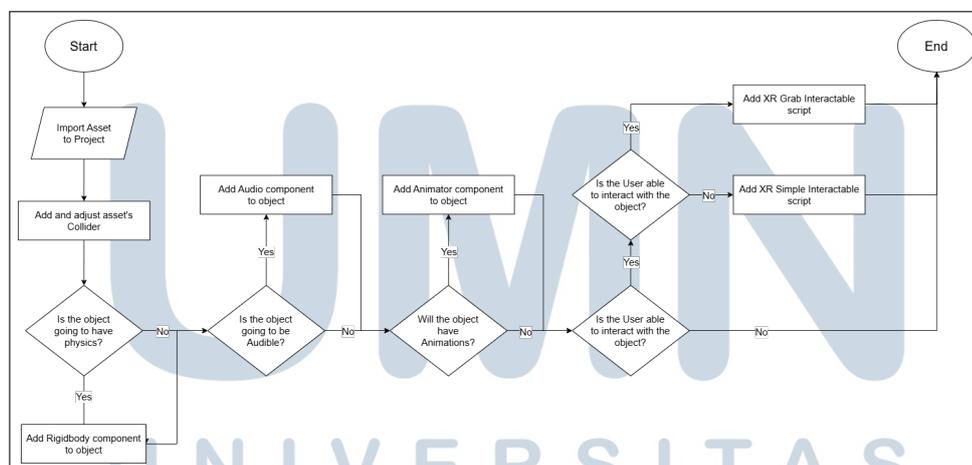
## B Fitur Interaksi Object

Setelah aset dinilai layak dan sesuai dengan kebutuhan proyek, setiap objek 3D akan melalui serangkaian proses penyesuaian serta penambahan fitur tambahan yang bersifat teknis dan fungsional. Proses penyesuaian ini penting untuk memastikan bahwa setiap aset tidak hanya tampil secara visual sesuai dengan desain

pameran, tetapi juga dapat berfungsi dengan baik di dalam lingkungan aplikasi VR yang sedang dikembangkan.

Penyesuaian yang dilakukan meliputi integrasi berbagai fitur interaktif yang dapat memperkaya pengalaman pengguna, seperti penambahan animasi, pengaturan trigger untuk interaksi, serta penyesuaian posisi, rotasi, dan skala objek agar sesuai dengan tata letak lingkungan virtual. Dalam beberapa kasus, aset juga memerlukan pemisahan struktur prefab agar dapat dikendalikan secara individual melalui skrip, terutama jika objek tersebut memerlukan interaksi spesifik seperti membuka pintu, menampilkan informasi tambahan, atau merespons tindakan pengguna secara real-time.

Selain itu, aset yang telah diimplementasikan ke dalam Unity juga diuji kembali untuk memastikan bahwa tidak terjadi konflik dengan komponen lain di dalam proyek, serta untuk menjaga performa aplikasi agar tetap optimal saat dijalankan menggunakan perangkat VR. Proses ini dilakukan secara berulang dan berkelanjutan, mengikuti prinsip iteratif dalam pengembangan aplikasi, hingga seluruh aset dapat berfungsi dengan baik dan memenuhi standar kualitas yang telah ditentukan. Alur lengkap dari proses pencarian, penyesuaian, dan implementasi aset 3D ke dalam aplikasi dapat dilihat pada Gambar 3.4.



Gambar 3.4. Flowchart Implementasi Aset 3D

Salah satu fitur interaktif yang diimplementasikan dalam aplikasi ini adalah kemampuan bagi pengguna untuk mengambil dan memindahkan objek, seperti furnitur atau elemen properti lainnya yang terdapat dalam lingkungan pameran virtual. Fitur ini dirancang untuk memberikan kebebasan eksplorasi yang lebih tinggi kepada pengguna, sehingga mereka dapat berinteraksi langsung dengan objek-objek 3D secara intuitif, seperti mengangkat, memutar, atau memindahkan

posisi objek sesuai keinginan.

Dalam penerapannya, objek-objek tersebut dilengkapi dengan skrip khusus, yang mengatur perilaku saat objek dipegang maupun setelah dilepaskan. Ketika objek dilepaskan, sistem akan memulai timer untuk mereset posisi objek kembali ke tempat semula setelah jangka waktu tertentu, untuk menjaga keteraturan lingkungan virtual. Selain itu, efek suara tumbukan juga ditambahkan untuk meningkatkan kesan realisme ketika objek bersentuhan dengan permukaan lain. Dengan adanya sistem ini, pengguna tidak hanya dapat melihat detail furnitur secara dekat, tetapi juga merasakan pengalaman interaktif yang mendekati situasi nyata dalam ruang pameran fisik.

```
1
2 public class Props : MonoBehaviour
3 {
4     public AudioClip collisionAudio;
5
6     private AudioSource audioSource;
7     private Rigidbody rigidBody;
8
9     private Vector3 startingPosition;
10    private Quaternion startingRotation;
11    // Start is called before the first frame update
12    void Start()
13    {
14        audioSource = this.GetComponent<AudioSource>();
15        rigidBody = this.GetComponent<Rigidbody>();
16        Invoke("SetVolume", 2f);
17
18        startingPosition = transform.position;
19        startingRotation = transform.rotation;
20    }
21
22    void SetVolume(){
23        audioSource.volume = 1f;
24    }
25
26    void OnCollisionEnter(Collision col){
27        float velocity = Mathf.Clamp(Vector3.Magnitude(rigidBody.
28        velocity) / 5f, 0.1f, 1f);
29        audioSource.PlayOneShot(collisionAudio, velocity);
30    }
```

```

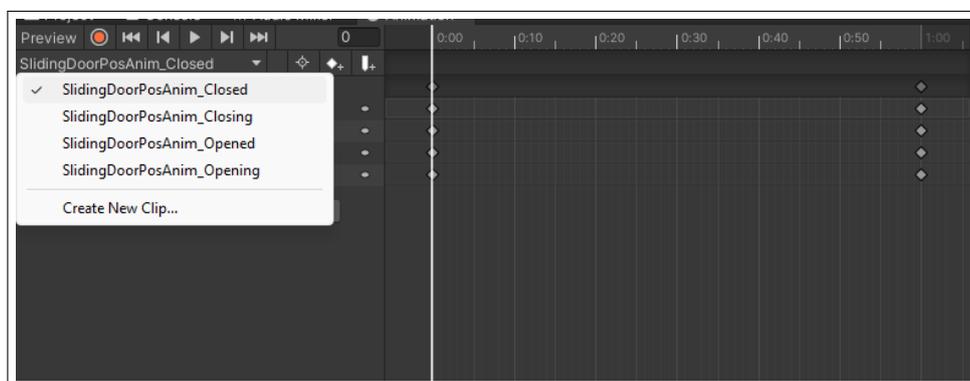
31 void Reset(){
32     Instantiate(this.gameObject, startingPosition,
startingRotation);
33     Destroy(this.gameObject);
34 }
35
36 public void SetHeld(bool enterSelect){
37     if(enterSelect){
38         CancelInvoke();
39     }
40     else{
41         Invoke("Reset", 30f);
42     }
43 }
44 }

```

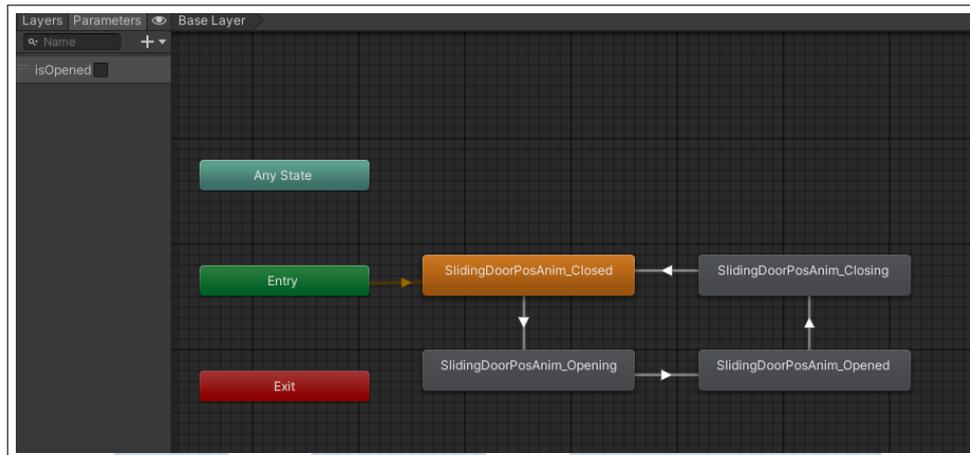
Kode 3.1: Kode Props.cs

Untuk mendukung interaksi yang lebih realistis dalam aplikasi pameran berbasis Virtual Reality ini, penting bahwa beberapa objek, seperti pintu, dilengkapi dengan animasi yang memungkinkan pengguna melihat transisi visual saat pintu dibuka atau ditutup. Animasi ini dirancang agar sesuai dengan konteks interaksi pengguna di lingkungan virtual, sehingga meningkatkan kesan imersif selama eksplorasi pameran.

Proses implementasi animasi dilakukan dengan menambahkan component *Animator* pada objek yang bersangkutan. Setiap object yang memerlukan animasi seperti pintu, dirancang memiliki beberapa state animasi, seperti tertutup, terbuka, sedang terbuka, dan sedang tertutup. Animasi untuk masing-masing state dibuat secara manual menggunakan *Unity Animation*, sehingga setiap transisi dapat dikontrol secara detail sesuai dengan kebutuhan interaksi pengguna.



Gambar 3.5. Window *Animation*



Gambar 3.6. Window *Animator*

Untuk memindah dan memainkan animasi yang sudah dibuat, object yang akan memiliki animasi tersebut juga memiliki *script* yang akan menyalakan kondisi dari animasi. Contoh kode untuk menyalakan animasi untuk onject seperti pintu dapat dilihat pada sebagai berikut.

```

1
2 using UnityEngine.XR.Interaction.Toolkit;
3
4 public class DoorTrigger : MonoBehaviour
5 {
6     public Animator anim;
7     public bool isOpened;
8     public XRSimpleInteractable simpleInteractable;
9     private void Start()
10    {
11        anim = this.GetComponent<Animator>();
12    }
13
14    public void DoorTriggered()
15    {
16        Debug.Log("Triggered");
17        if (anim.GetBool("isOpened") == true)
18        {
19            Debug.Log("Opening");
20            anim.SetBool("isOpened", false);
21        }
22        else
23        {
24            Debug.Log("Closing");

```

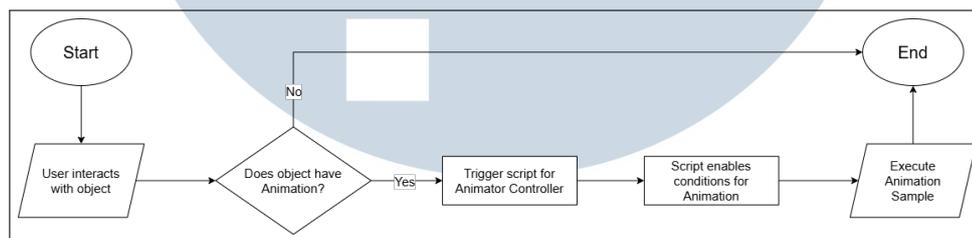
```

25     anim . SetBool ( "isOpened" , true );
26     }
27 }
28 }

```

Kode 3.2: Kode Trigger Animasi Pintu

Seluruh state animasi tersebut kemudian dihubungkan dan diatur melalui *Animator Controller*, yang memungkinkan transisi antar-state dijalankan berdasarkan kondisi logis tertentu, seperti input dari pengguna atau pemicu interaksi lainnya. Dengan pendekatan ini, pengguna dapat merasakan perubahan kondisi objek secara visual dan fungsional, misalnya ketika mereka menekan tombol interaksi atau mendekati pintu. Hal ini selain meningkatkan realisme dalam simulasi, akan tetapi juga memberikan pengalaman yang lebih responsif dan intuitif di dalam aplikasi. Alur untuk animasi pada sebuah object dapat dilihat pada Gambar 3.7.



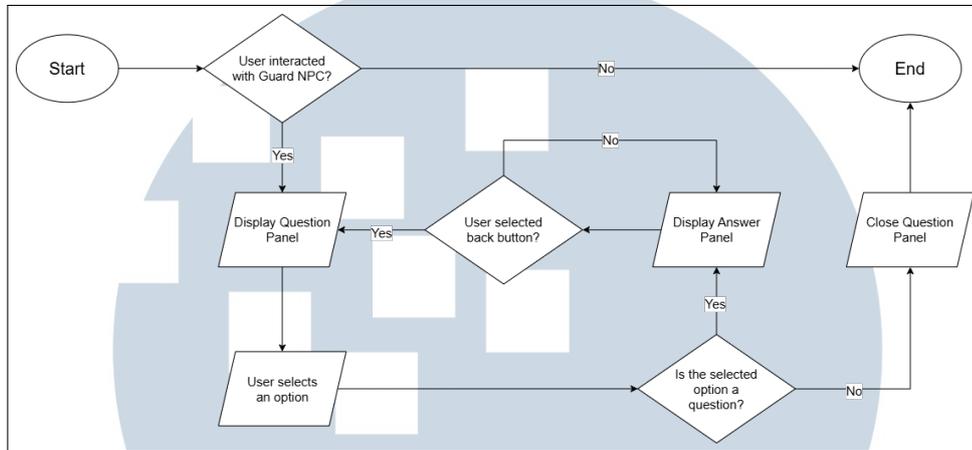
Gambar 3.7. Flowchart Trigger Animasi

### C *Non-Playable Characters (NPC) dan User Interface (UI)*

Dalam pengembangan aplikasi pameran berbasis Virtual Reality ini, dirancang beberapa *Non-Playable Characters (NPC)* atau Karakter Non-Pemain yang berperan sebagai elemen interaktif untuk meningkatkan pengalaman pengguna di dalam lingkungan virtual. NPC ini ditempatkan secara strategis di dalam area pameran dan memiliki fungsi tertentu yang mendukung penyampaian informasi secara imersif dan terarah.

Salah satu NPC utama yang dikembangkan adalah *Guard NPC*, yang berfungsi sebagai pemandu atau narasumber umum bagi pengunjung pameran. *Guard NPC* ini dirancang untuk memberikan jawaban atas sejumlah pertanyaan umum terkait isi dan tujuan pameran perumahan. Pengguna dapat memilih pertanyaan yang tersedia dan menerima jawaban yang telah diprogram sebelumnya. Antarmuka interaksi diaktifkan melalui tombol *Grip* pada kontroler VR ketika

pengguna mengarahkan pandangannya ke NPC dan menekan tombol tersebut, maka akan muncul tampilan UI yang menampilkan daftar pertanyaan yang dapat dipilih.



Gambar 3.8. Flowchart Alur UI NPC *Guard*

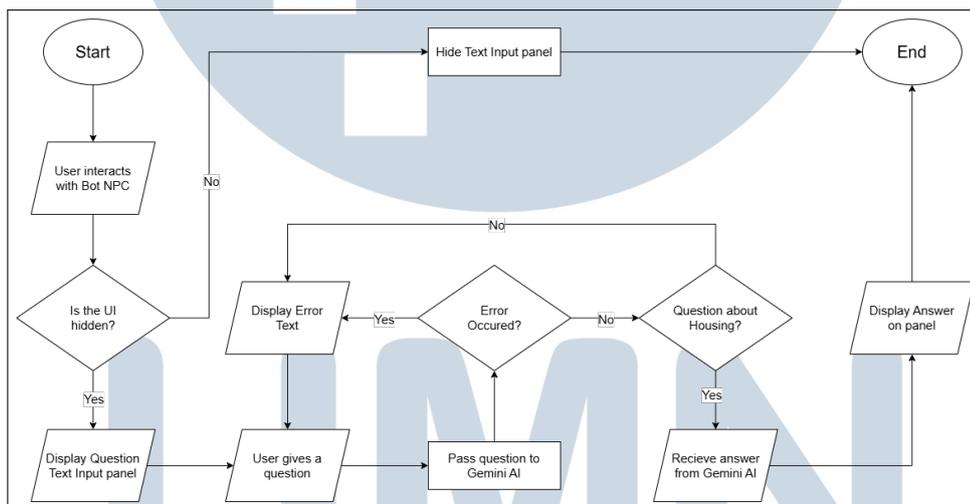
Fitur ini dirancang untuk memberikan nuansa interaktif yang lebih alami dan mendukung proses eksplorasi pameran secara mandiri oleh pengguna. Dengan adanya NPC seperti ini, diharapkan pengguna dapat memperoleh informasi secara langsung tanpa harus keluar dari lingkungan VR, sehingga menciptakan pengalaman yang lebih kohesif dan mendalam dalam menjelajahi pameran virtual. Contoh *Guard NPC* di dalam pameran terdapat pada Gambar 3.9.



Gambar 3.9. Contoh NPC *Guard* pada Perumahan

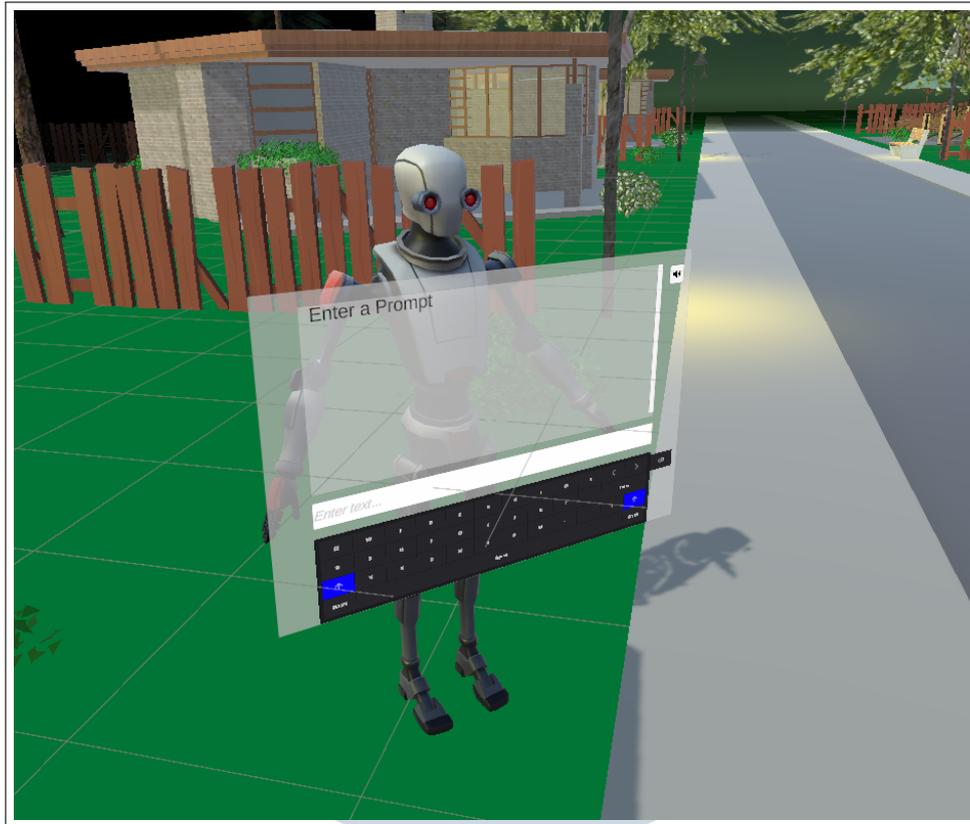
Selain *Guard NPC*, aplikasi ini juga dilengkapi dengan *Bot NPC* yang memiliki kemampuan lebih dinamis dalam memberikan respons terhadap pertanyaan pengguna. Tidak seperti *Guard NPC* yang hanya memberikan jawaban terhadap pertanyaan-pertanyaan yang telah disiapkan sebelumnya, *Bot NPC* dirancang untuk dapat menjawab berbagai pertanyaan secara fleksibel dengan memanfaatkan *Artificial Intelligence* buatan, yaitu *AI Gemini*.

Interaksi dengan *Bot NPC* difasilitasi melalui tampilan UI berupa sebuah kotak input teks. Melalui UI ini, pengguna dapat mengetikkan pertanyaan secara bebas terkait topik perumahan, seperti informasi mengenai tipe rumah, lokasi, fasilitas, hingga saran-saran umum. Pertanyaan yang dikirimkan akan diproses oleh sistem dan diteruskan ke *AI Gemini* untuk dianalisis. Hasil jawaban dari *AI* kemudian akan ditampilkan kembali kepada pengguna melalui layar dialog yang tersedia di dalam lingkungan virtual.



Gambar 3.10. Flowchart Alur UI NPC *Bot*

Penggunaan *Bot NPC* ini bertujuan untuk menghadirkan interaksi yang lebih realistis dan personal, serta memberikan keleluasaan bagi pengguna untuk menggali informasi lebih mendalam sesuai kebutuhan masing-masing. Integrasi *AI* dalam elemen *NPC* ini juga menjadi nilai tambah dalam inovasi pengembangan aplikasi, karena mampu menciptakan pengalaman pameran yang tidak hanya pasif, tetapi juga responsif dan adaptif terhadap pengguna. Contoh *Bot NPC* di dalam pameran bisa dilihat pada Gambar 3.11.



Gambar 3.11. Contoh NPC Bot pada Perumahan

#### D Chatbot AI

*NPC Bot* adalah Salah satu komponen interaktif penting yang dikembangkan dalam aplikasi pameran virtual ini. *NPC Bot* ini yang berperan sebagai perantara antara pengguna dan sistem *Artificial Intelligence*. NPC ini ditempatkan secara strategis di area pameran dan dirancang untuk menjadi titik informasi dinamis, di mana pengguna dapat mengajukan berbagai pertanyaan yang berkaitan dengan topik perumahan secara langsung. *API Key* yang digunakan merupakan *API AI Gemini* oleh karena Gemini menyediakan kunci API gratis yang dapat digunakan untuk sementara. Proyek Unity dan AI Gemini akan dihubungkan menggunakan *Google App Script* agar prompt yang diberikan peserta dari aplikasi dapat dikirimkan langsung ke AI Gemini.

*Google Apps Script* adalah platform berbasis cloud milik Google yang memungkinkan pengguna untuk menulis kode JavaScript untuk mengotomatisasi, memperluas, dan mengintegrasikan berbagai layanan *Google*, seperti *Google Sheets*, *Google Docs*, dan *Google Drive*. Dalam proyek ini, *Google Apps*

Script dimanfaatkan sebagai perantara yang menghubungkan aplikasi Unity dengan layanan AI Gemini melalui permintaan *HTTP*. Script ini akan menerima prompt dari aplikasi Unity, kemudian meneruskan permintaan tersebut ke endpoint API Gemini, dan akhirnya mengirimkan respons kembali ke aplikasi. Dengan pendekatan ini, komunikasi antara Unity dan AI Gemini dapat dilakukan secara efisien tanpa perlu membangun server backend secara terpisah[12].

Melalui antarmuka input yang tersedia pada NPC tersebut, pengguna dapat mengetikkan pertanyaan secara bebas dan tidak terbatas pada daftar pertanyaan yang telah ditentukan sebelumnya. Pertanyaan-pertanyaan ini kemudian dikirimkan ke *AI Gemini* yang akan memprosesnya dan menghasilkan jawaban yang relevan, informatif, dan kontekstual. Hasil jawaban dari AI ditampilkan kembali melalui UI dalam lingkungan virtual, sehingga pengguna dapat memperoleh informasi secara langsung tanpa perlu keluar dari pengalaman VR.

```
1
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using UnityEngine.UI;
6 using UnityEngine.Networking;
7 using System.Threading.Tasks;
8 using TMPro;
9
10 public class Gemini : MonoBehaviour
11 {
12     public struct TMP_Prompt{
13         public string prompt;
14         public TMP_Text tmpText;
15
16         public TMP_Prompt(string prompt, TMP_Text tmpText){
17             this.prompt = prompt;
18             this.tmpText = tmpText;
19         }
20     }
21
22     const string gasUrl = "https://script.google.com/macros/s/
AKfycbyBDjOG2gciYxtrFC80v5GyPPAPJb0cC8AruZ-60
J1Mx9S3_wOGQtLz7F4jgfdz9Ieyqg/exec";
23     const string imageRequestPrompt = "Provide a url for an image
of the following using the first image from google images and
make sure the url is valid:";
```

```

24
25 public Texture2D loadingImage;
26 public Texture2D errorImage;
27 private string inputPrompt;
28 private RawImage responseImage;
29 private List<TMP.Prompt> prompts = new List<TMP.Prompt>();
30
31 void Start()
32 {
33     StartCoroutine(SendDataToGasTMP());
34 }
35
36 private IEnumerator SendDataToGasGeminiInputField(
37 GeminiInputField geminiInputField)
38 {
39     Debug.Log("Running chat request.");
40     geminiInputField.SetResponseText("Running chat request.");
41
42     WWWForm form = new WWWForm();
43     form.AddField("parameter", inputPrompt);
44     UnityWebRequest www = UnityWebRequest.Post(gasUrl, form);
45
46     yield return www.SendWebRequest();
47     string response = "";
48
49     if(www.result == UnityWebRequest.Result.Success){
50         response = www.downloadHandler.text;
51     }
52     else{
53         response = "There was an error";
54     }
55     geminiInputField.SetResponseText(response);
56 }
57
58 private IEnumerator SendDataToGasTMP()
59 {
60     while(true){
61         if(prompts.Count == 0){
62             yield return new WaitForSeconds(1f);
63         }
64         else{
65             Debug.Log("Running chat request.");

```

```

66         WWWFom form = new WWWFom();
67         form.AddField("parameter", prompts[0].prompt);
68         UnityWebRequest www = UnityWebRequest.Post(gasUrl,
form);
69
70         yield return www.SendWebRequest();
71         string response = "";
72
73         if(www.result == UnityWebRequest.Result.Success){
74             response = www.downloadHandler.text;
75             response.Replace("*", "");
76         }
77         else{
78             response = "There was an error";
79         }
80
81         Debug.Log(response);
82
83         prompts[0].tmpText.text = response;
84         prompts.RemoveAt(0);
85     }
86 }
87
88 }
89
90 public void EnterInputFieldPrompt(string prompt,
GeminiInputField inputField)
91 {
92     inputPrompt = prompt;
93     StartCoroutine(SendDataToGasGeminiInputField(inputField));
94 }
95
96 public void EnterTextPrompt(string prompt, TMP_Text text)
97 {
98     prompts.Add(new TMP_Prompt(prompt, text));
99 }
100
101 private IEnumerator SendImageDataToGas()
102 {
103     Debug.Log("Running chat request.");
104     responseImage.texture = loadingImage;
105
106     WWWFom form = new WWWFom();

```

```

107     form.AddField("parameter", inputPrompt);
108     UnityWebRequest www = UnityWebRequest.Post(gasUrl, form);
109
110     yield return www.SendWebRequest();
111
112     string response = "";
113     if(www.result == UnityWebRequest.Result.Success){
114         response = www.downloadHandler.text;
115         StartCoroutine(GetTexture(response));
116     }
117     else{
118         responseImage.texture = errorImage;
119     }
120 }
121
122 private IEnumerator GetTexture(string url){
123     Debug.Log("Running image request: " + url);
124     UnityWebRequest request = UnityWebRequestTexture.
GetTexture(url);
125     yield return request.SendWebRequest();
126
127     if(request.result == UnityWebRequest.Result.
ConnectionError || request.result == UnityWebRequest.Result.
ProtocolError){
128         responseImage.texture = errorImage;
129         StartCoroutine(SendImageDataToGas());
130     }
131     else{
132         responseImage.texture = ((DownloadHandlerTexture)
request.downloadHandler).texture;
133     }
134 }
135
136 public void EnterImagePrompt(string prompt, RawImage image)
137 {
138     inputPrompt = imageRequestPrompt + prompt;
139     responseImage = image;
140     StartCoroutine(SendImageDataToGas());
141 }
142
143 }

```

Kode 3.3: Kode Gemini.cs

Integrasi fitur ini bertujuan untuk memberikan pengalaman pameran yang lebih interaktif, adaptif, dan personal, sekaligus memperkenalkan potensi penggunaan AI dalam mendukung layanan informasi berbasis virtual reality. *NPC Bot* tidak hanya menambah nilai inovatif pada aplikasi, tetapi juga memperkuat fungsi edukatif dari pameran dengan memberikan akses informasi yang responsif sesuai kebutuhan pengguna.

### **3.3.3 Testing dan Evaluasi**

Proses pengujian pada aplikasi pameran virtual ini dilakukan menggunakan metode *Exploratory Testing*, yaitu pendekatan pengujian yang bersifat fleksibel dan tidak bergantung pada skenario atau skrip pengujian yang kaku. Dalam metode ini, penguji secara aktif menjelajahi aplikasi, mencoba berbagai fitur dan skenario penggunaan secara spontan untuk menemukan bug, inkonsistensi, atau masalah pada alur interaksi pengguna. Pendekatan ini sangat sesuai digunakan dalam tahap pengembangan yang masih bersifat iteratif, karena memungkinkan deteksi dini terhadap kesalahan atau kekurangan yang mungkin belum teridentifikasi dalam perencanaan awal. Hasil dari eksplorasi ini kemudian digunakan untuk melakukan perbaikan secara langsung pada sistem[13].

Setelah pengujian internal oleh pengembang, tahap evaluasi dilanjutkan bersama supervisor proyek guna memperoleh masukan dan validasi terhadap fitur yang telah dikembangkan. Supervisor turut mengamati dan mencoba langsung aplikasi, serta memberikan umpan balik terkait alur interaksi, kualitas visual, dan kesesuaian konten dengan tujuan pameran. Berdasarkan observasi tersebut, dilakukan sejumlah penyesuaian tambahan, seperti pengoptimalan posisi objek, perbaikan waktu respons pada interaksi dengan NPC, serta penyederhanaan elemen UI agar lebih mudah digunakan dalam perangkat VR.

Secara keseluruhan, aplikasi menunjukkan performa yang stabil dan fitur-fitur utama berfungsi sesuai dengan rancangan. Kolaborasi antara pengembang dan supervisor dalam proses evaluasi memberikan kontribusi signifikan dalam penyempurnaan akhir aplikasi. Meskipun pengujian belum dilakukan secara luas kepada pengguna umum, hasil evaluasi menunjukkan bahwa aplikasi telah memenuhi kebutuhan dasar pameran virtual dan siap untuk digunakan dalam konteks akademik atau institusional sebagai media interaktif berbasis VR.

### 3.4 Kendala dan Solusi yang Ditemukan

Selama berjalannya magang ini, ditemukan beberapa kendala, seperti:

- Tidak adanya aset model rumah yang dapat dipakai untuk testing aplikasi
- Device yang digunakan untuk mengerjakan proyek magang kurang ideal untuk development VR di unity

Solusi yang dimanfaatkan untuk menghadapi kendala tersebut, adalah:

- Mencari aset melalui sumber-sumber online seperti *Unity Asset Store*, *SketchFab*, *TurboSquid* dan sebagainya.
- Memaksimalkan *modular* development dan melakukan optimasi pada aset-aset yang digunakan untuk mengurangi beban grafis saat pengujian. Selain itu, sebagian proses development, terutama scripting dan desain UI, dilakukan secara terpisah tanpa menjalankan simulasi VR secara penuh.

