

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Kegiatan dalam bekerja dalam perusahaan FX Media pada divisi *Game Developer* yang bertugas membangun aplikasi gamifikasi menggunakan program aplikasi *Unity Engine* ataupun *Unreal Engine* sesuai kebutuhan klien. Selama pelaksanaan magang, berada dibawah pengawasan *XR Lead* dan *XR Senior Dev* sebagai mentor yang memberikan tugas, arahan, dan memastikan dapat memahami aplikasi atau fitur kebutuhan klien dengan jelas.

3.2 Tugas yang Dilakukan

Selama bekerja di FX Media sebagai *Game Developer*, dimulai dengan proses pelatihan sampai dengan pengerjaan pengerjaan proyek menggunakan aplikasi program *Unity*. Berikut adalah tugas yang diberikan selama magang :

- Mempelajari materi *Unity Programming Patterns* menggunakan bahasa *C-sharp* untuk membuat kodingan di dalam *Game Engine Unity* lebih efisien, mudah dilihat, dan mudah di-organisasi.
- Mempelajari penerapan *multiplayer*, dan AI dalam aplikasi *Unity*.
- Membuat proyek kecil dengan menggunakan teknologi aplikasi terbaru yaitu *WebGPU* dalam aplikasi *Unity*.
- Melakukan penerapan, konversi, dan menambahkan fitur-fitur baru pada proyek X.
- Melakukan pencarian dan perbaikan bug pada proyek X.

3.3 Uraian Pelaksanaan Magang

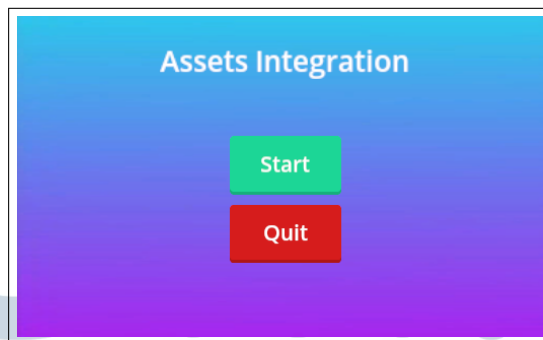
Berikut merupakan paparan kegiatan selama melaksanakan kegiatan magang di FX Media sebagai *Game Developer* di tim XR :

Tabel 3.1. Tugas dan Pekerjaan yang dilaksanakan selama pelaksanaan magang

Minggu	Tugas dan Pekerjaan yang dilaksanakan
1	Mempelajari dan mempraktik latihan penggunaan <i>asset and plugin</i> dalam proyek baru.
2	Mempelajari penerapan <i>multiplayer</i> dalam <i>Unity</i> dan mempelajari <i>Unity Programming Patterns</i> .
3	Mempelajari dan penerapan teknologi terbaru <i>WebGPU</i> dalam aplikasi <i>Unity</i> .
4	Mempelajari aplikasi <i>Meta Spark Studio</i> dan perenapan <i>Large Language Model (LLM) ChatGPT</i> dalam <i>Unity WebGPU</i> .
5	Penerapan <i>Unity WebGPU, Large Language Model (LLM) ChatGPT</i> pada proyek C.
6	Memperbaiki bug dan mengganti <i>asset</i> yang tidak berfungsi dengan aplikasi <i>Unity</i> terbaru pada proyek C.
7	Membuat Proyek A baru dengan penerapan <i>UI/UX, 3D model, Large Language Model (LLM) ChatGPT, Text-to-Speech, Speech-to-Text</i> dan <i>WebGPU</i> .
8-12	Mempelajari <i>plugin Unity Render Streaming</i> dan <i>Unreal Pixel Streaming</i> . Membuat Proyek A yang sama seperti dua minggu lalu dengan aplikasi <i>game engine Unreal</i> .
13-14	Memperbaiki bug dan mengganti <i>asset</i> yang tidak berfungsi dengan aplikasi <i>Unity</i> terbaru pada proyek N.
15	Mempelajari dan menerapkan <i>plugin WebXR</i> pada proyek A.

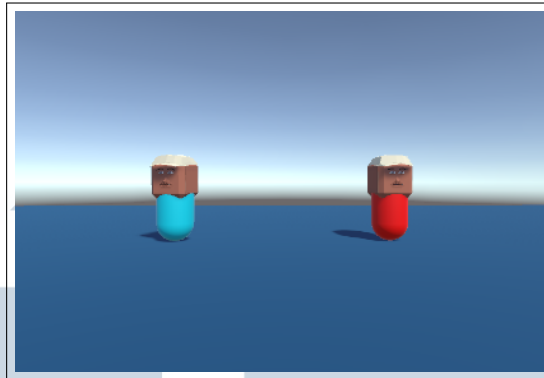
3.3.1 Latihan Praktik

Pada minggu pertama pelaksanaan magang, diberikan latihan praktik untuk membuat gim dalam *Unity*, tujuan praktik tersebut adalah untuk mengetahui kemampuan penggunaan *Game Engine Unity* seperti ; penggunaan *asset*, *multiplayer system*, *inventory system*, *waypoint system*, *minimap system*, *Audio system*, *text-to-speech*, *lipsync*, *C-Sharp Programming*, *scene manager*, *UI/UX*, dan *input system*. Pada tampilan pada Gambar 3.1 untuk aplikasi latihan memiliki *title*, *background*, *music*, tombol *Start*, tombol *Quit*. pada *scene* ini musik akan dipilih secara acak dan dimainkan. Apabila kursor pengguna berada di atas atau menekan salah satu tombol *Start* atau *Quit* maka akan memainkan suara FX. Tombol *Start* akan memindahkan pengguna kedalam gim, dan Tombol *Quit* akan menutup aplikasi gim. Pada saat tombol *Start* di tekan maka dari *scene* menu utama akan berpindah ke *game scene* seperti pada Gambar 3.2. Pada *game scene* terdapat dua *Non Player Control (NPC)*, *item*, dan pemain lain. NPC dapat menghadap ke pemain dan NPC dapat berbicara menggunakan *Speech-to-Text plugin* saat pemain ber-interaksi dengan NPC. Pemain dapat ber-interaksi dengan *item* yang berada di lantai dan *item* tersebut akan hilang an otomatis berada di *inventory* pemain. Pemain juga dapan menjatuhkan *item* ke lantai dan *item* di *inventory* pemain akan di-*update* kembali.

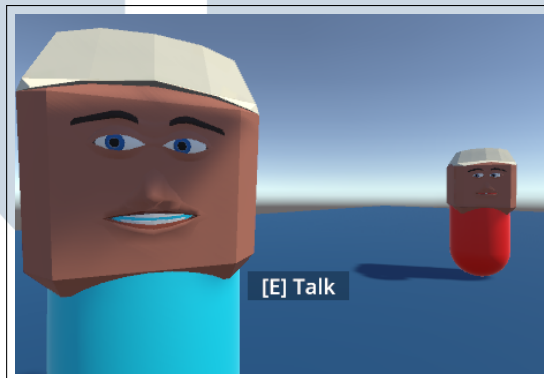


Gambar 3.1. Tampilan Menu utama

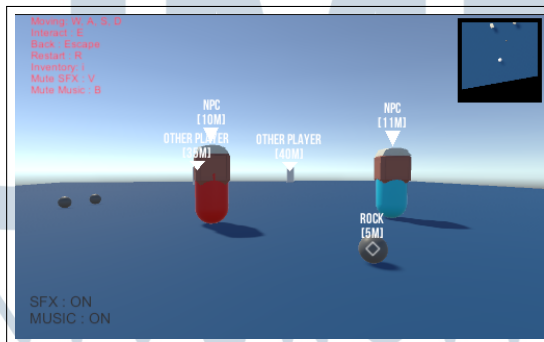
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.2. Tampilan dalam gim



Gambar 3.3. Tampilan NPC

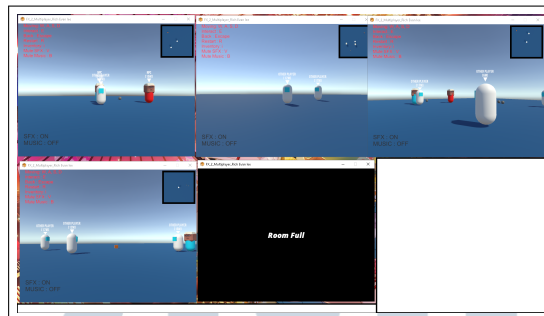


Gambar 3.4. Tampilan UI dalam gim



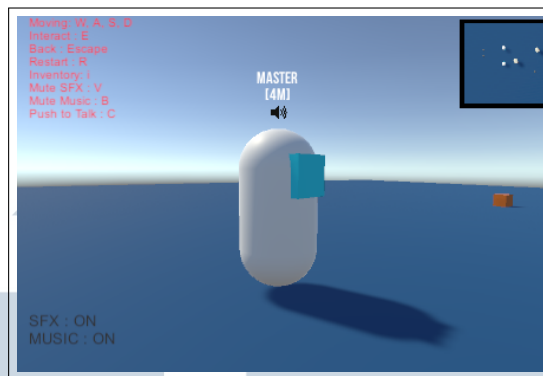
Gambar 3.5. Tampilan *inventory*

Pada minggu ke-dua menambahkan sistem *multiplayer* (Gambar 3.6) dan *Voice Chat* (Gambar 3.7) dalam gim. Pemain lain dapat bergabung kedalam game dan bermain bersama pemain yang lain. Pemain yang pertama masuk ke ruangan akan memiliki nama "*Master*" dan pemain yang lain dinamakan "*Other-Player*". jika ruangan telah memiliki lebih dari batas pemain (pada Gambar 3.6 memiliki lebih dari 4 pemain) maka pemain tersebut tidak diberikan izin masuk ke ruangan tersebut. Pemain dapat menggunakan sistem *Voice-Chat*, sistem ini untuk berkomunikasi dengan pemain lain di sekitar dan jika pemain berbicara maka akan ada indikator ikon suara seperti pada Gambar 3.7.



Gambar 3.6. Tampilan *multiplayer*

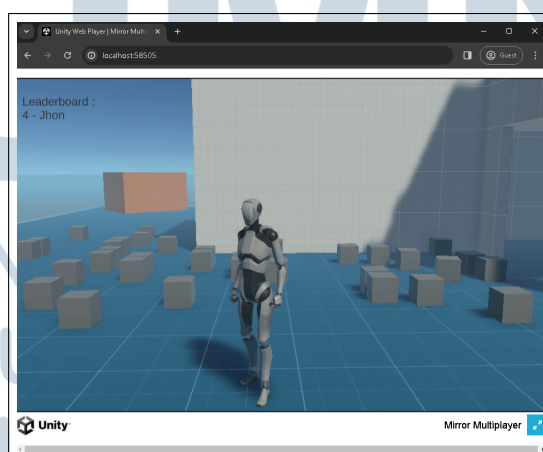
UNIVERSITAS
MULTIMEDIA
NUSANTARA



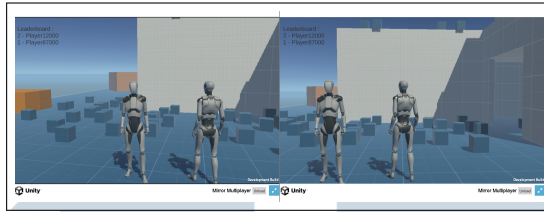
Gambar 3.7. Tampilan *Voice Chat* dan Pemain lain

3.3.2 Integrasi *WebGPU* dan AI

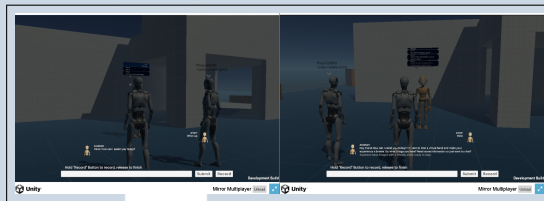
Pada minggu ketiga sampai keempat supervisor menyarankan untuk mempelajari *Unity Programming Pattern* dapat membuat kodingan di dalam *Game Engine Unity* lebih efisien, mudah dilihat, dan mudah di-organisasi. Setelah itu *supervisor* menyarankan untuk mempelajari integrasi *WebGPU* pada *Unity WebGL* seperti pada Gambar 3.8. *WebGPU* merupakan penerus dari *WebGL* dan bertujuan untuk menawarkan merupakan *API* grafis modern yang dirancang untuk menyediakan akses ke perangkat keras *Graphics Processing Unit (GPU)* untuk aplikasi web. Setelah mempelajari integrasi *WebGPU* supervisor memberikan tugas untuk menambahkan sistem *multiplayer* (Gambar 3.9), bersama dengan sistem *ChatBot AI* (Gambar 3.10)



Gambar 3.8. Tampilan gim dalam *WebGPU*



Gambar 3.9. *Multiplayer game dalam WebGPU*



Gambar 3.10. *Tampilan Chat bot AI dalam permainan multiplayer dalam WebGPU*

Pada minggu ke lima dan seterusnya di tugaskan ke mentor lain dan diberi tugas integrasi *WebGPU* pada projek C. Projek C merupakan projek gamifikasi yang mensimulasikan kegiatan asisten dokter rumah sakit. Projek C dibangun dengan versi *Unity* yang sudah cukup lama, dikarenakan *WebGPU* merupakan fitur baru dalam *Unity* maka projek C perlu di *upgrade* agar *WebGPU* dapat di-integrasi ke projek C. Pada projek C banyak *plugin* yang tidak dapat digunakan di *WebGPU* dan harus di ganti dengan *plugin* yang baru atau dapat digunakan oleh *WebGPU*. Setelah projek C dapat di integrasikan dengan *WebGPU*, diberikan tugas untuk menambahkan AI *ChatGPT* sebagai dokter yang membantu pemain menyelesaikan tugasnya dalam gim seperti pada Gambar 3.11. Code 3.1 merupakan potongan dari sebagian kode system *ChatBot* untuk me-requesting *API ChatGPT* dan mengembalikan respon AI ke dalam gim seperti pada Gambar 3.11.

```

1 public class OpenAIRequest
2 {
3     private OpenAIApi openai;
4     public OpenAIRequest(string API_Key, string Org_Key
5     )
6     {
7         openai = new OpenAIApi(API_Key, Org_Key);
8     }
9     public enum who
10    {

```

```

10     user , assistant , system
11 }
12 public void NewChat()
13 {
14     request = new CreateChatCompletionRequest
15     {
16         Model = "gpt-3.5-turbo" ,
17         Messages = new List<ChatMessage>()
18     };
19     OnMessageChange?.Invoke();
20 }
21 public async void SendMessage(string content)
22 {
23     // add user msg
24     AddMessage(content);
25
26     // get req
27     var response = await openai.
CreateChatCompletion(request);
28
29     // add bot msg
30     AddMessage(response.Choices.First().Message ,
who.assistant);
31 }
32 public void AddedMessage(ChatMessage message)
33 {
34     request.Messages.Add(message);
35 }
36 public void AddMessage(string message , who role =
who.user)
37 {
38     ChatMessage chatMessage = new ChatMessage
39     {
40         Content = message ,
41         Role = role.ToString()
42     };

```



```

43     AddMessage(chatMessage , role );
44     }
45 }

```

Kode 3.1: Contoh potongan kode request ChatGPT API



Gambar 3.11. Tampilan gim Projek C dan ChatGPT

3.3.3 Virtual Avatar

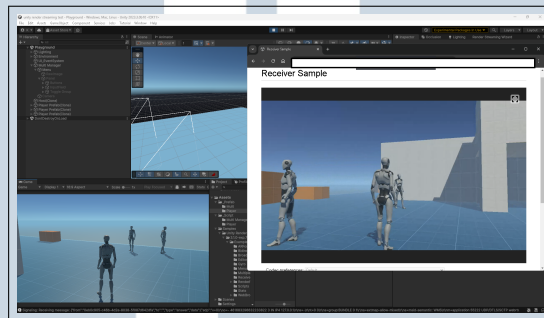
Pada Gambar 3.12 merupakan tampilan proyek terbaru berhubungan dengan *Virtual Assistant* untuk proyek A. *Virtual Assistant* merupakan istilah yang digunakan untuk representasi *avatar* atau representasi grafis yang sangat realistis seperti manusia dan menggunakan AI interaksi atau *ChatBot* untuk melayani pengguna. Dengan menggunakan potongan Kode 3.1 dari proyek sebelumnya, maka pengerjaan proyek A ini dengan mudah dan cepat.



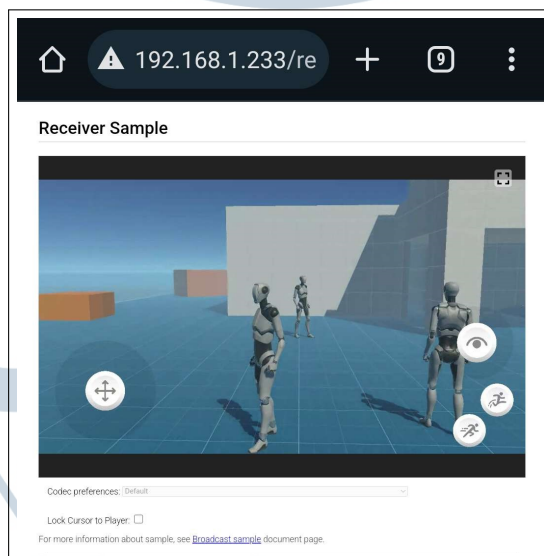
Gambar 3.12. Tampilan gim Projek A, Virtual Assistant

3.3.4 *Render Streaming*

Unity Render Streaming merupakan teknologi yang dikembangkan oleh Unity yang memungkinkan *rendering* grafis di *server* dan hasil *rendering* tersebut di-*streaming* ke berbagai perangkat klien, terlihat pada Gambar 3.13. Kegunaan *Unity Render Streaming* tersebut memungkinkan aplikasi yang membutuhkan grafis intensif untuk dijalankan pada perangkat yang kemampuan grafis terbatas seperti *smartphone*, atau *web browser* terlihat pada Gambar 3.14.



Gambar 3.13. Tampilan *Unity Render Streaming* server dan klien

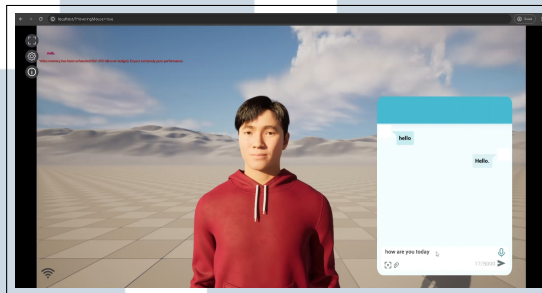


Gambar 3.14. Tampilan *Unity Render Streaming* di perangkat *mobile* klien

3.3.5 *Unreal Pixel Streaming*

Unreal Pixel Streaming sama seperti *Unity Render Streaming* dimana teknologi tersebut yang disediakan oleh *Unreal Engine*. Saat mengerjakan proyek

A, mentor memberikan tugas untuk membangun proyek A dalam *Unreal Engine* agar dapat menghasilkan *avatar* yang lebih realistik dan menggunakan *Unreal Pixel Streaming* agar klien dapat menggunakan aplikasi tersebut tanpa memerlukan perangkat yang kemampuan grafis terbatas seperti Gambar 3.15.



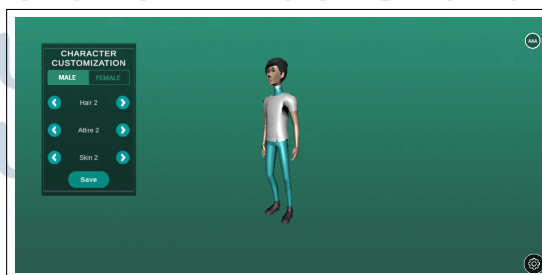
Gambar 3.15. Tampilan *Unreal Pixel Streaming* dalam browser

3.3.6 *Character customization*

Pada minggu ke tiga belas, mentor memberikan projek N dan memberikan tugas untuk menambahkan fitur *Character customization*. *Character customization* merupakan fitur yang memberikan kontrol kepada pemain untuk menciptakan karakter yang merefleksikan preferensi atau identitas pemain seperti Gambar 3.16 dan Gambar 3.17.



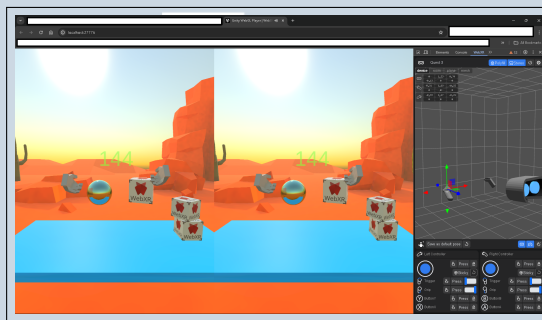
Gambar 3.16. Tampilan *Character customization Female*



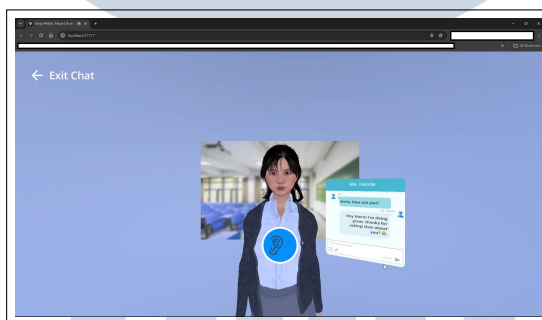
Gambar 3.17. Tampilan *Character customization Male*

3.3.7 WebXR

Pada minggu ke lima belas, mentor memberikan tugas untuk melakukan riset tentang *Unity WebXR* (Gambar 3.18). *WebXR* adalah API yang memungkinkan pengalaman VR atau AR yang dapat diakses melalui *web browser*. Setelah melakukan riset, mentor memberikan tugas untuk men-integrasikan proyek A dengan *WebXR* terlihat pada Gambar 3.19.



Gambar 3.18. Tampilan *WebXR Demo* dalam bentuk VR



Gambar 3.19. Tampilan *WebXR* proyek A dalam bentuk AR

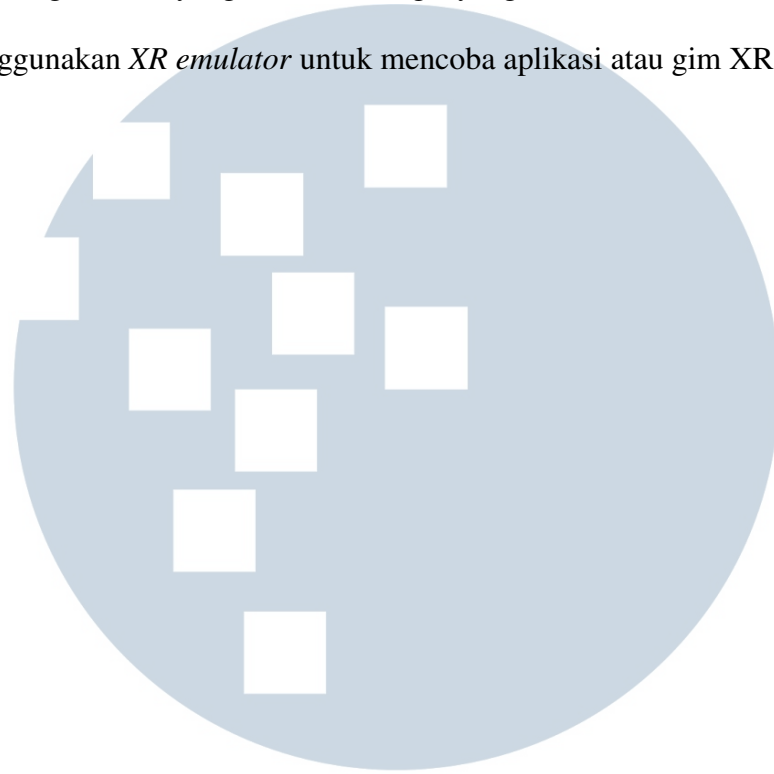
3.4 Kendala dan Solusi yang Ditemukan

Semasa pelaksanaan magang di FX Media, ada beberapa kendala saat bekerja magang di FX Media yaitu :

1. Beberapa *Unity asset* tidak dapat digunakan pada versi *Unity* terbaru.
2. Tidak memiliki perangkat keras untuk mencoba aplikasi atau gim XR.

Dari kendala-kendala diatas, ada sebuah solusi dari masing-masing kendala tersebut yaitu :

1. Menggunakan *asset* terbaru atau menggunakan *asset* alternatif atau membangun *asset* yang memiliki fungsi yang sama dari awal.
2. Menggunakan *XR emulator* untuk mencoba aplikasi atau gim XR.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA