

BAB 3 METODOLOGI PENELITIAN

3.1 Identifikasi Masalah

Penelitian ini didorong oleh tugas magang di perusahaan, yaitu merancang *avatar chatbot* berbasis AI dengan tampilan visual realistis dan animasi ekspresi wajah bersama *lip-sync* berbasis *visemes* dan volume suara secara *real-time*.

Permintaan dari pihak mentor magang berfokus pada pemanfaatan *Unreal Engine 5.2* sebagai platform utama pengembangan, dengan dukungan dari teknologi *MetaHuman*, *Text-to-Speech* (TTS), dan animasi *lip-sync* berbasis *visemes*. Namun, dalam pelaksanaan tugas tersebut muncul keterbatasan animasi *MetaHuman* dalam menampilkan ekspresi wajah, dan gerakan mulut secara *real-time*.

Identifikasi masalah ini menjadi dasar untuk menyusun strategi pembangunan *avatar chatbot* yang tidak hanya mempertimbangkan aspek teknis, tetapi juga aspek *usability* dari perspektif pengguna akhir. Sistem yang dihasilkan diharapkan dapat memenuhi ekspektasi dan berpotensi untuk diterapkan lebih luas dalam aplikasi industri seperti layanan pelanggan digital, simulasi sosial, maupun edukasi berbasis *avatar*.

3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mengidentifikasi seluruh komponen yang dibutuhkan oleh *avatar chatbot* agar dapat berjalan dengan baik dan sesuai dengan tujuan penelitian. Berikut beberapa kebutuhan dalam *avatar chatbot* ini:

1. *Epic Games Launcher* untuk mengunduh dan membuka aplikasi *Unreal Engine 5.2*.
2. Memiliki *template Avatar MetaHuman* dalam proyek.
3. Mendapatkan *API key ChatGPT* agar dapat melakukan panggilan ke *API ChatGPT*.
4. Mendapatkan *API key Azure TTS* agar dapat panggilan ke *API Azure TTS*.
5. Membangun *Pose Asset* yang terdiri atas ekspresi wajah natural, senang, dan marah.

6. Membangun *Pose Asset* yang terdiri atas semua bentuk mulut *visemes* yang *Azure TTS* dapat berikan.
7. Membangun animasi *blueprint* sistem yang dapat menggabungkan *Pose Asset* ekspresi dan *visemes*.
8. Membangun *blueprint* sistem yang dapat menerima input teks dari pengguna.
9. Membangun *blueprint* sistem yang dapat menghasilkan respons berupa teks menggunakan API *ChatGPT*.
10. Membangun *blueprint* sistem yang dapat mengingat percakapan antara pengguna pada sesi tersebut agar *ChatGPT* dapat memahami konteks perbincangan.
11. Membangun *blueprint* sistem yang dapat mengelola teks yang dihasilkan akan diubah menjadi suara melalui layanan *Microsoft Azure TTS* dan menghasilkan *timestamp viseme* suara TTS.
12. Membangun animasi *blueprint* sistem yang dapat menyinkronkan gerakan bibir dan ekspresi wajah secara *real-time* dan terlihat natural berdasarkan *viseme* dan volume dari TTS.

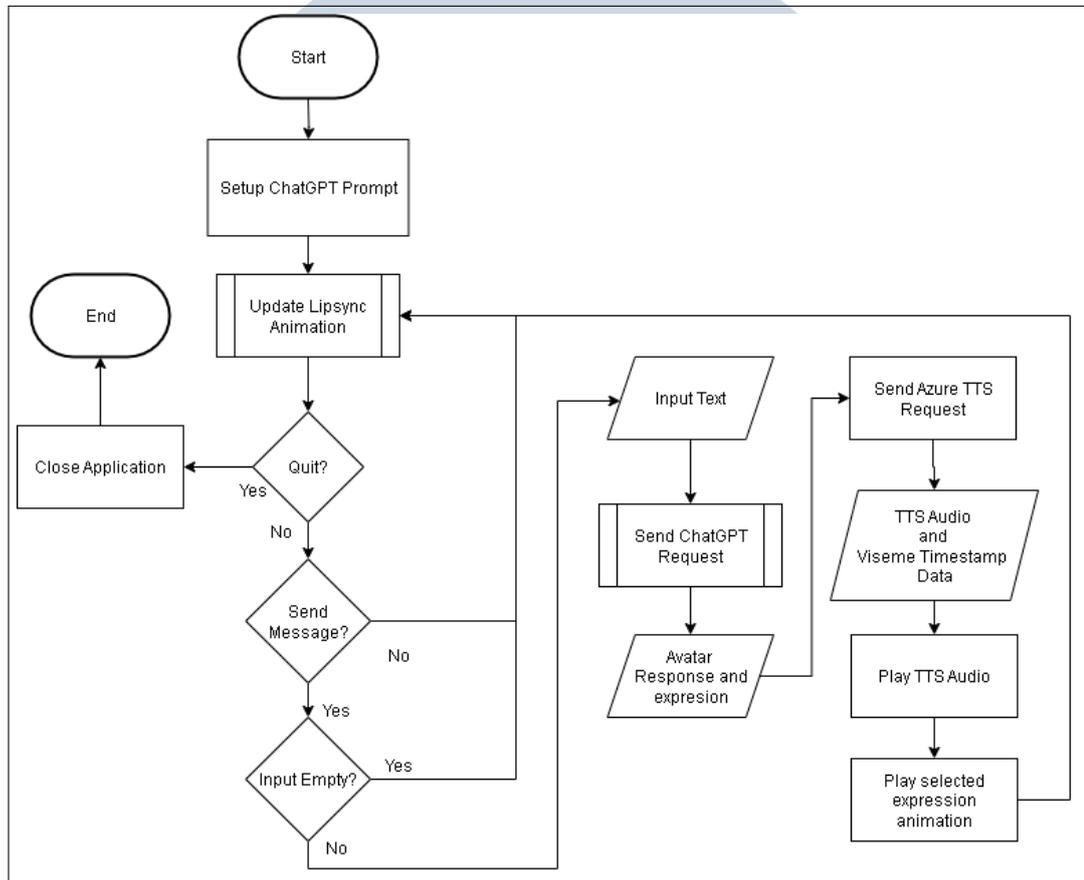
3.3 Desain Sistem

Pada tahap ini, dilakukan perancangan alur sistem untuk menggambarkan bagaimana interaksi antara pengguna, layanan API, dan sistem *Unreal Engine 5.2* yang menampilkan *avatar* digital berbasis *MetaHuman*.

3.3.1 Flowchart Proses Aplikasi

Gambar 3.1 merupakan proses alur kerja aplikasi. Sistem pertama akan mengatur *prompt* agar *ChatGPT* dapat menghasilkan respons *avatar* dan ekspresi dalam satu panggilan dalam bentuk *json*. Saat pengguna mengirim teks, sistem memeriksa kekosongan. Jika tidak kosong, sistem mengirim permintaan ke *ChatGPT* API dan menerima respons teks dan ekspresi dalam format *JSON*. Respons teks *avatar* dari *ChatGPT* dikirim ke *Microsoft Azure TTS* API untuk menghasilkan audio dan data *viseme timestamp* yang dapat digunakan untuk memainkan animasi *visemes* yang akurat dengan audio TTS bersama animasi

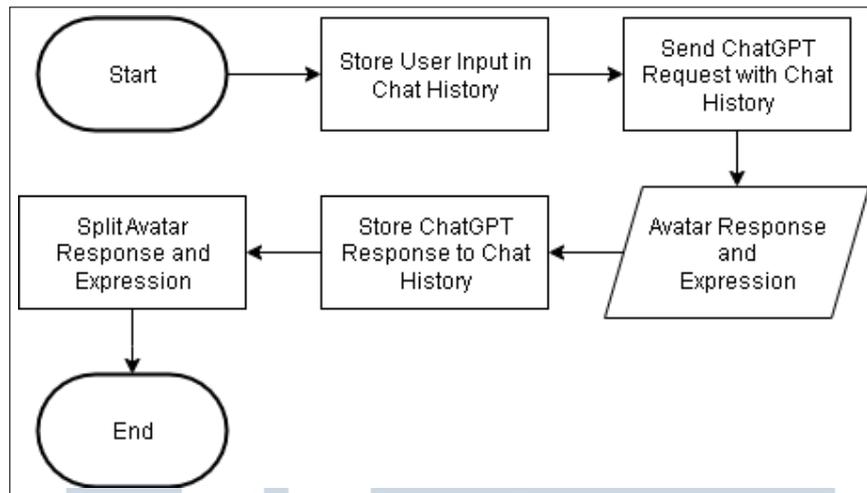
ekspresi yang telah ditentukan oleh *ChatGPT*. Setelah mendapatkan audio dari *Microsoft Azure*, audio tersebut akan dimainkan bersama dengan animasi ekspresi yang telah ditentukan oleh *ChatGPT*.



Gambar 3.1. Flowchart proses alur kerja avatar chatbot

3.3.2 Flowchart Request ChatGPT API

Gambar 3.2 merupakan proses saat ingin melakukan *request ChatGPT* API. Pertama, sistem akan menyimpan input dari pengguna ke *chat history* agar *ChatGPT* dapat memahami konteks percakapan dan menghasilkan respons yang akurat. Respons *ChatGPT* akan disimpan untuk pemanggilan berikutnya, respons *ChatGPT* berisi respons *avatar* dan ekspresi dalam bentuk *json* yang akan dipisahkan untuk pemanggilan *Azure TTS* API dan animasi ekspresi.

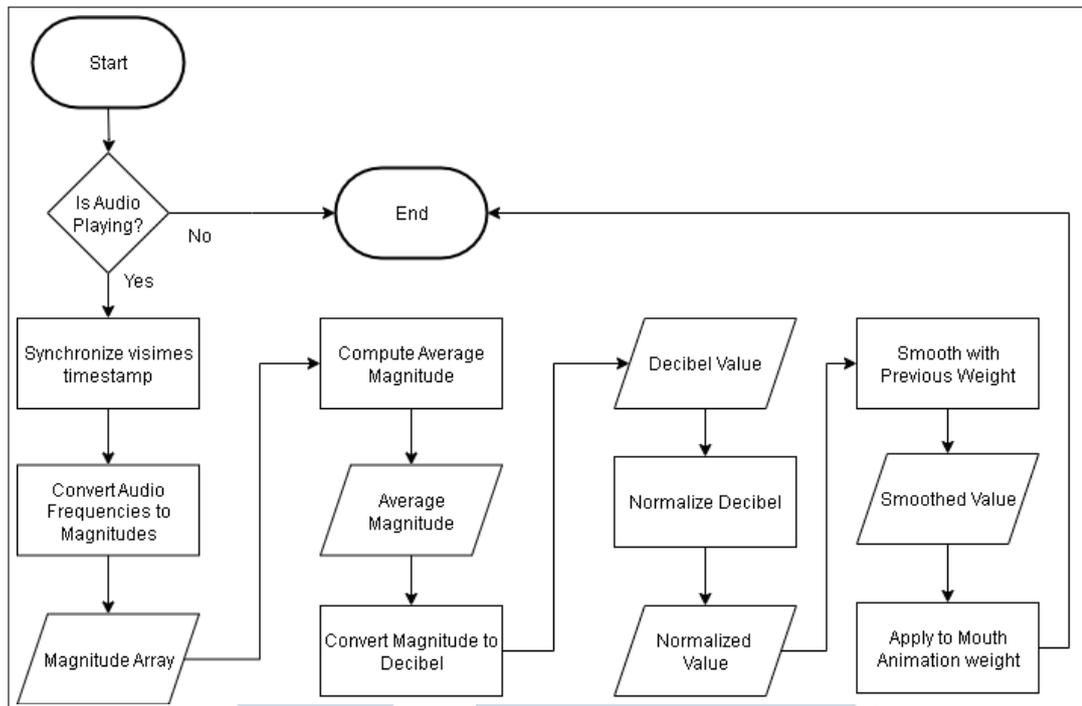


Gambar 3.2. *Flowchart Request ChatGPT API*

3.3.3 *Flowchart Lipsync Update*

Gambar 3.3 merupakan proses pengaturan *lipsync visemes* dan volume suara audio. Sistem memeriksa kesesuaian antara audio TTS yang diputar dan *timestamp viseme*, secara bersamaan sistem akan mengambil frekuensi dari audio yang sedang dimainkan dan dijadikan *magnitudes* yang nantinya akan dirata-ratakan. Rata-rata *magnitude* tersebut dikonversi menjadi *decibel* atau besar volume suara audio. *Decibel* akan dinormalisasi agar dapat digunakan ke *weight* animasi. Hasil normalisasi akan dimuluskan dengan hasil *weight* animasi agar gerakan mulut tidak langsung meloncat dengan gerakan yang cepat.

U M N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



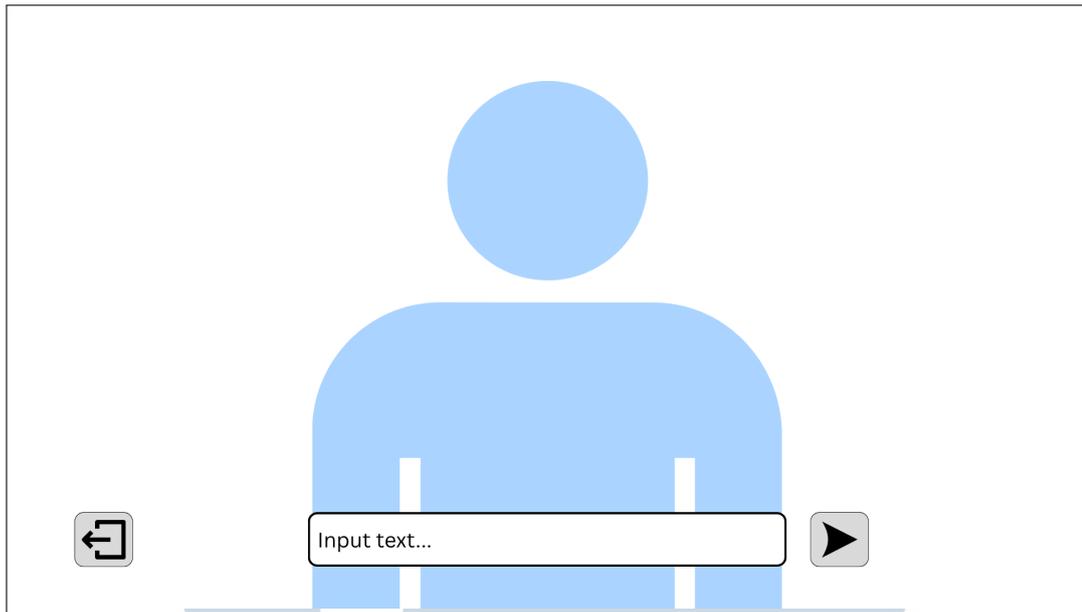
Gambar 3.3. Flowchart Lipsync Update

3.4 Perancangan Desain Antarmuka

Perancangan desain antarmuka bertujuan untuk memberikan gambaran visual mengenai bagaimana pengguna *User Interface* (UI) dari aplikasi akan berinteraksi dengan *avatar chatbot*.

3.4.1 Mockup Aplikasi

Antarmuka dirancang sederhana agar pengguna mudah berinteraksi melalui input teks, lalu melihat ekspresi dan *lip-sync* avatar. Gambar 3.4 merupakan tampilan antarmuka yang berfokus pada kesederhanaan, keterbacaan, dan kemudahan navigasi, dan penempatan elemen-elemen penting seperti wajah *avatar*, kolom input teks, tombol kirim, dan tombol keluar.



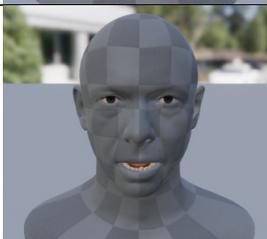
Gambar 3.4. *Mockup Avatar Chatbot*

Setelah merancang antarmuka pengguna, langkah selanjutnya adalah membangun sistem animasi ekspresi wajah dan pergerakan mulut *visemes* ke dalam *avatar*. Tabel 3.1 menunjukkan gambar dan keterangan ekspresi wajah dan *visemes* yang digunakan dalam perancangan *avatar*.

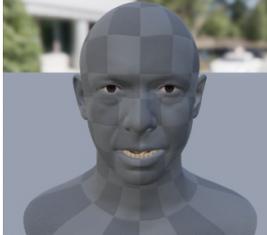
Tabel 3.1. Bentuk ekspresi wajah dan mulut

Gambar	Keterangan
	<p>Ekspresi wajah natural.</p>
	<p>Ekspresi wajah senang.</p>
<p>Lanjut pada halaman berikutnya</p>	

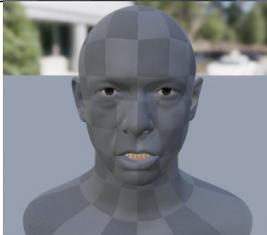
Tabel 3.1 Bentuk ekspresi wajah dan mulut (Lanjutan)

Gambar	Penjelasan
	<p>Ekspresi wajah marah.</p>
	<p>Viseme vokal terbuka sedang seperti "cat", "sofa", dan "luck".</p>
	<p>Viseme suara "ah" terbuka, seperti "father".</p>
	<p>Viseme suara "aw" seperti "law".</p>
	<p>Viseme suara "Eh" seperti "bet".</p>
	<p>Viseme suara "Er" seperti "bird".</p>
<p>Lanjut pada halaman berikutnya</p>	

Tabel 3.1 Bentuk ekspresi wajah dan mulut (Lanjutan)

Gambar	Penjelasan
	<p><i>Viseme</i> suara "Ee" seperti "see" dan "ih" seperti "bit".</p>
	<p><i>Viseme</i> suara "W" dan "oo" seperti "woo".</p>
	<p><i>Viseme</i> suara "O" seperti "boat".</p>
	<p><i>Viseme</i> suara "ow" seperti dalam "house".</p>
	<p><i>Viseme</i> suara "oy" seperti dalam "boy".</p>
	<p><i>Viseme</i> suara "eye" seperti dalam "hi".</p>
<p>Lanjut pada halaman berikutnya</p>	

Tabel 3.1 Bentuk ekspresi wajah dan mulut (Lanjutan)

Gambar	Penjelasan
	<p><i>Viseme</i> suara "H" seperti dalam "hey".</p>
	<p><i>Viseme</i> suara "R" seperti dalam "red".</p>
	<p><i>Viseme</i> suara "L" seperti dalam "love".</p>
	<p><i>Viseme</i> suara "S" dan "Z" mirip dengan bentuk bibir "see".</p>
	<p><i>Viseme</i> suara "Sh", "ch", dan "j" seperti "shoe" atau "chop".</p>
	<p><i>Viseme</i> suara "Th" seperti dalam "this".</p>
<p>Lanjut pada halaman berikutnya</p>	

Tabel 3.1 Bentuk ekspresi wajah dan mulut (Lanjutan)

Gambar	Penjelasan
	<p>Viseme suara "F" dan "V" memerlukan gigi atas menyentuh bibir bawah seperti dalam "fridge" dan "victory".</p>
	<p>Viseme suara "D", "T", dan "N" melibatkan penempatan lidah seperti dalam "dim" dan "team".</p>
	<p>Viseme suara "K", "G", dan "ng" melibatkan belakang lidah seperti dalam "cat", "get", dan "gang".</p>
	<p>Viseme suara "P", "B", dan "M" menggunakan kedua bibir bersama seperti dalam "pop", "bob", dan "mop".</p>

3.5 Pembuatan Aplikasi

Pada bagian ini dijelaskan tahapan realisasi dari desain antarmuka menjadi sebuah aplikasi interaktif menggunakan *Unreal Engine 5.2*. Proyek ini bertujuan untuk membangun *avatar chatbot* yang realistis dengan ekspresi wajah dan sinkronisasi gerak mulut berdasarkan *output* suara dari sistem *Text-to-Speech* (TTS).

3.5.1 Lingkungan Pengembangan

Pengembangan aplikasi *avatar chatbot* realistik ini dilakukan menggunakan *Unreal Engine 5.2*, yang merupakan salah satu *game engine* dengan kemampuan *rendering real-time*, animasi karakter realistik melalui *MetaHuman*, dan sistem *Blueprint visual scripting* yang memudahkan pengembangan tanpa harus bergantung sepenuhnya pada bahasa pemrograman tingkat rendah.

A *Azure Speech Plugin*

Plugin Azure Speech digunakan untuk mengakses layanan *Text-to-Speech* (TTS) dan juga data *visemes* dari *Microsoft Azure*. Konfigurasi *plugin* dilakukan melalui pembuatan akun *Azure*, pendaftaran layanan, dan penyesuaian *Key API*, *Region ID*, *Locale*, *Voice*, dan *Enable viseme* pada sistem *Blueprint Unreal Engine 5.2*.

B *VaRest Plugin*

VaRest adalah *plugin* pihak ketiga di *Unreal Engine 5.2* yang memungkinkan komunikasi HTTP REST API secara visual melalui *Blueprint*.

C *MetaHuman Framework*

MetaHuman adalah komponen integral dalam pengembangan visual karakter *avatar*. *MetaHuman* menyediakan struktur wajah yang sangat rinci dan mendukung ekspresi wajah secara *real-time*.

3.5.2 Antarmuka Pengguna

Antarmuka pengguna dirancang dengan prinsip kesederhanaan dan efisiensi agar pengguna dapat berinteraksi secara intuitif dengan *avatar chatbot*. Antarmuka dilakukan langsung di dalam *Unreal Engine 5.2* dengan memanfaatkan sistem UI bernama *Unreal Motion Graphics* (UMG). Antarmuka utama terdiri dari beberapa elemen penting yang mendukung interaksi:

A Tampilan Wajah Avatar

Bagian utama dari layar menampilkan wajah *avatar* yang dirancang menggunakan *MetaHuman*. *Avatar* ini akan merespons pengguna melalui ekspresi wajah dan pergerakan mulut, termasuk pergerakan *viseme* yang disesuaikan dengan volume audio saat berbicara. *Viewport* ini bersifat dinamis dan ditampilkan dalam kamera yang mengarah ke wajah karakter.

B Input Teks

Elemen *Input* teks memungkinkan pengguna untuk menuliskan pertanyaan atau perintah yang ingin diberikan kepada *avatar chatbot*. Setelah mengetik, pengguna dapat mengirimkannya melalui tombol kirim.

C Tombol Kirim

Tombol ini digunakan untuk mengirimkan teks yang telah ditulis pengguna ke sistem *backend chatbot*. Setelah dikirim, respons yang diterima akan diproses dan dikonversi menjadi audio melalui TTS, dan *avatar* akan mulai berbicara serta bergerak sesuai ekspresi.

D Tombol Keluar

Tombol keluar disediakan agar pengguna dapat keluar dari aplikasi dengan mudah. Ini mengirimkan perintah untuk menutup aplikasi secara keseluruhan.

3.5.3 Sistem Suara dan Ekspresi

Sistem suara dan ekspresi merupakan komponen inti dalam menciptakan interaksi yang realistis antara pengguna dan *avatar chatbot*. Sistem ini terdiri dari dua bagian utama, yaitu TTS untuk menghasilkan suara dari respons teks, dan sistem ekspresi wajah dan *lipsync* untuk menyelaraskan pergerakan wajah dan mulut *avatar* dengan audio yang dihasilkan.

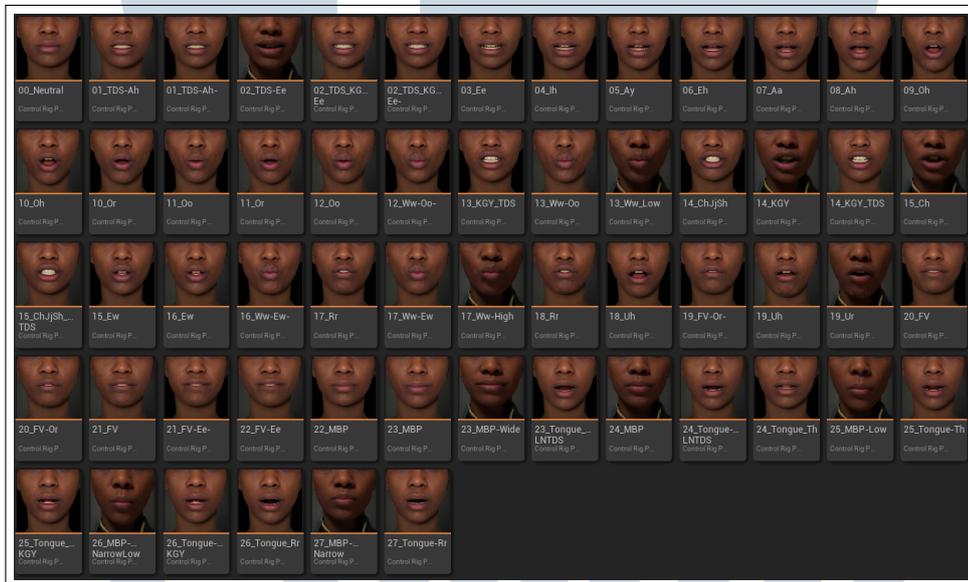
A Text-to-Speech

Azure Speech Plugin digunakan untuk menghasilkan suara berbasis *cloud* langsung ke dalam *Unreal Engine 5.2*. Setelah pengguna mengirim teks melalui UI,

teks tersebut dikirim ke server *Azure* menggunakan *plugin*, *Azure* mengembalikan hasil TTS dalam bentuk *variable sound wave* yang nantinya dapat diputar di dalam *Unreal Engine 5.2* melalui komponen audio.

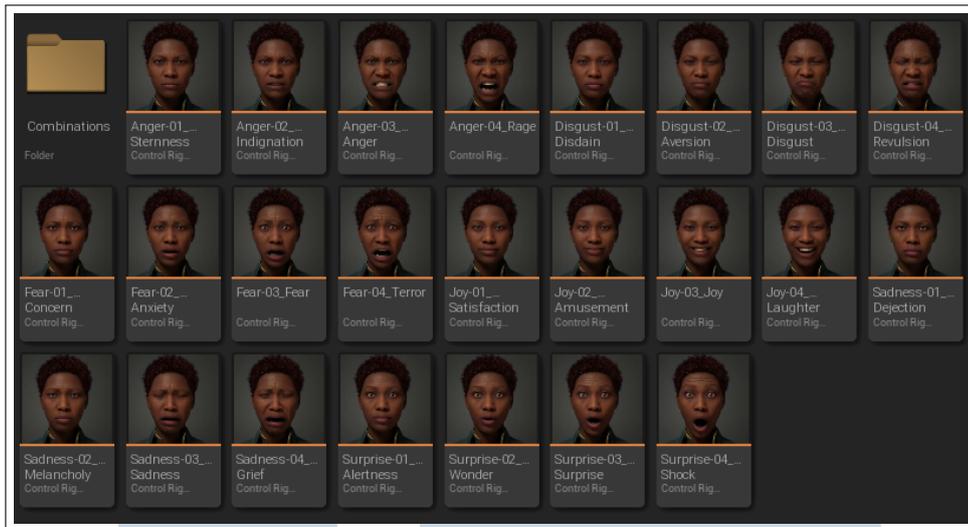
B Kontrol Ekspresi dan *Viseme*

Sistem *viseme* dan ekspresi wajah berbasis *Pose Asset* yang dihasilkan dengan cara menggunakan *visemes Control Rig Pose* seperti pada Gambar 3.5 dan ekspresi *Control Rig Pose* Gambar 3.6 yang telah disediakan dari *MetaHuman*. *Control Rig Pose* merupakan pose yang disimpan dan dapat diterapkan pada waktu yang berbeda dalam animasi.



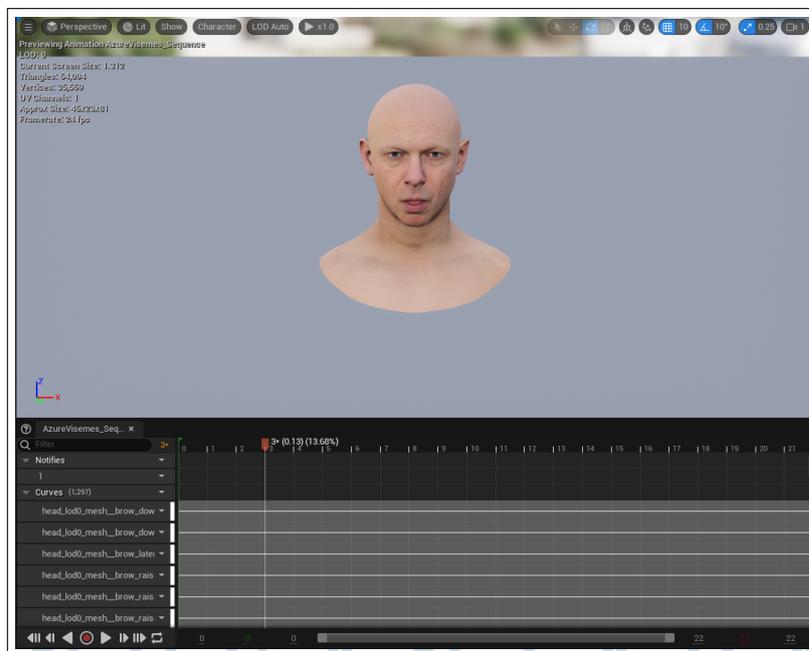
Gambar 3.5. *MetaHuman visemes Control Rig Pose*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

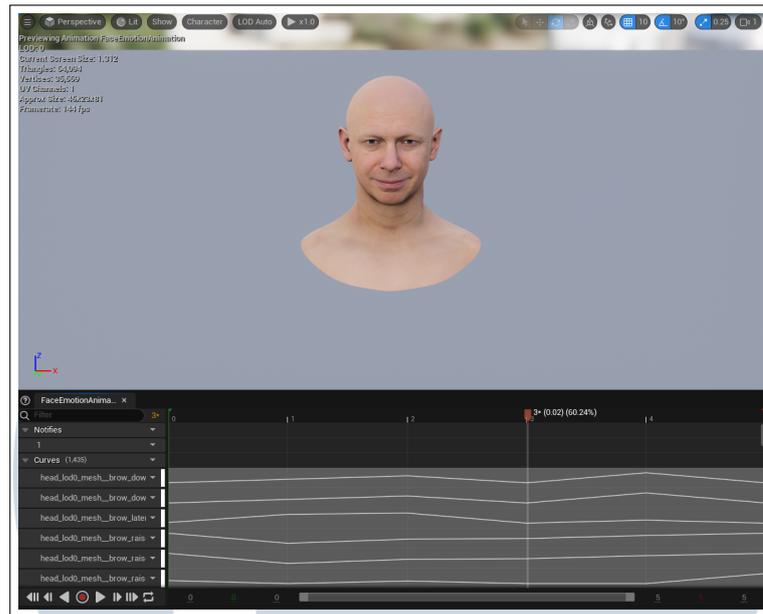


Gambar 3.6. *MetaHuman expressions Control Rig Pose*

Control Rig Pose tersebut akan digunakan pada *Animation Sequence* yang terdiri dari *viseme* dan ekspresi wajah seperti pada Gambar 3.7 yang menunjukkan pose animasi *viseme* suara 'eh' dan Gambar 3.8 yang menunjukkan pose animasi ekspresi senyum.



Gambar 3.7. *Visemes Animation Sequence*



Gambar 3.8. *Expressions Animation Sequence*

Animation sequence dapat membuat *viseme* dan ekspresi wajah *Pose Asset* yang dapat digunakan dalam *Animation Blueprint* untuk mengatur dan menggabungkan animasi *viseme* dan ekspresi wajah.

3.5.4 Pengujian Sistem

Setelah tahap pembangunan sistem selesai, dilakukan proses uji coba untuk memastikan bahwa seluruh fungsi dari aplikasi *chatbot avatar* berjalan sesuai dengan yang dirancang. Pengujian dilakukan secara menyeluruh terhadap antarmuka pengguna, pemanggilan API *ChatGPT* dan *Azure Speech*, sistem ekspresi wajah, sinkronisasi antara audio, dan menilai tingkat kegunaan sistem menggunakan instrumen *System Usability Scale (SUS)*.

3.5.5 Penulisan Laporan

Penulisan laporan dilakukan sebagai bagian akhir dari proses penelitian, dengan tujuan untuk mendokumentasikan secara menyeluruh hasil dari perancangan, pengembangan, pengujian, dan evaluasi sistem yang telah dilakukan. Laporan ini disusun berdasarkan hasil pembangunan sistem *lip-sync* berbasis volume untuk platform *Unreal Engine 5.2*, dan berdasarkan data evaluasi pengguna menggunakan *SUS*.