Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Perpustakaan Univeristas Multimedia Nusantara

BAB II

TINJAUAN PUSTAKA

2.1. Jacob

Jacob merupakan sebuah aplikasi *voice chatbot* berbahasa Inggris berbasis web dengan bantuan *platform* Wit.ai yang berfungsi untuk menyediakan informasi mengenai Program Dual Degree Informatika di UMN yang meliputi biaya, kurikulum, mata kuliah, fasilitas, dan karir (Wijaya dan Wicaksana, 2019). Jacob menerima masukan berupa suara dan memberikan keluaran juga berupa suara, serta bahasa yang digunakan adalah bahasa Inggris. Aplikasi *voice chatbot* Jacob dikembangkan dengan menggunakan database MySQL dan platform Wit.ai. Kemudian, basis pengetahuan Jacob disimpan dalam database MySQL (Octavany, 2019) dan (Archilles, 2019).

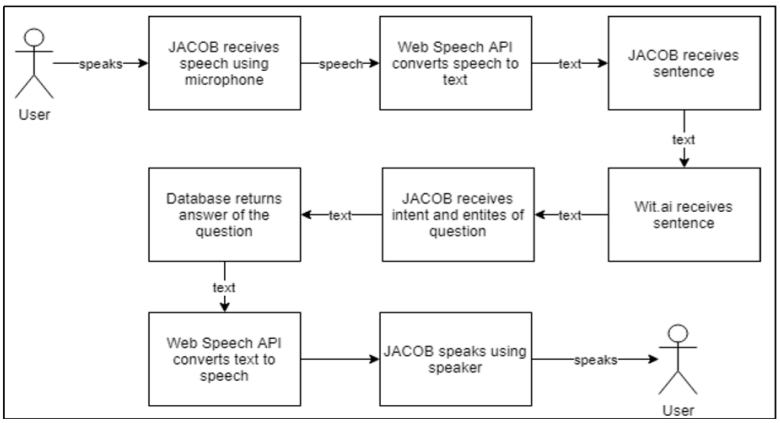
Chatbot Jacob memiliki tiga modul, yang berperan terhadap fitur dari *chatbot* Jacob, yaitu:

1. Modul Jacob

Modul Jacob berfungsi untuk membangun fitur *chatbot* Jacob sebagai *voice chatbot* pada umumnya. Modul Jacob dikembangkan dalam bahasa pemrograman PHP dengan *framework* Laravel. Pada modul Jacob ini, Jacob bekerja dengan menerima masukan berupa suara dari pengguna yang masuk melalui *microphone* diubah menjadi *text* (*speech to text*) dengan menggunakan *library* Web Speech API, berupa *Speech Recognition*. Agar *voice chatbot* Jacob dapat memproses bahasa manusia (*Natural Language Processing*), digunakan platform Wit.ai. Jadi, teks

yang diterima oleh Jacob tersebut dikirimkan ke Wit.ai dan kemudian Wit.ai menentukan konteks kalimat (Wijaya dan Wicaksana, 2019).

Wit.ai adalah sebuah *cloud platform Natural Language Understanding* (NLU) yang dimiliki dan dikembangkan oleh Facebook. Perbedaan Wit.ai dengan *tool Natural Language Understanding* lain adalah Wit.ai lebih berfokus dalam mencari arti atau makna dari suatu kalimat (Canonico dan Russis, 2018). Platform Wit.ai menggunakan integrasi *webhook*, yang pengiriman informasinya dilakukan melalui *web service*. Penentuan konteks percakapan dilakukan oleh platform Wit.ai, dan jawaban diberikan berupa *intent* dan *entities*. Wit.ai menggunakan dan mempelajari dari contoh kalimat serupa yang pernah digunakan sebelumnya dalam penentuan *intent* dan *entities* yang akan diberikan (Rahman dkk., 2017). Hasil yang didapat dari Wit.ai, berupa *intent* dan *entities*, selanjutnya dicocokkan oleh modul Jacob dengan basis data agar mendapatkan respon yang sesuai untuk menjawab pertanyaan pengguna (Wijaya dan Wicaksana, 2019). Selanjutnya, Jacob mengeluarkan respon suara dengan cara mengubah teks untuk respon menjadi suara dengan menggunakan *library* Web Speech API, berupa *Speech Synthesis* (Wijaya dan Wicaksana, 2019). Cara kerja Jacob ditunjukkan pada Gambar 2.1.



Gambar 2.1. Alur kerja *voice chatbot* Jacob
(Wijaya dan Wicaksana, 2019)

2. Modul Cleveree

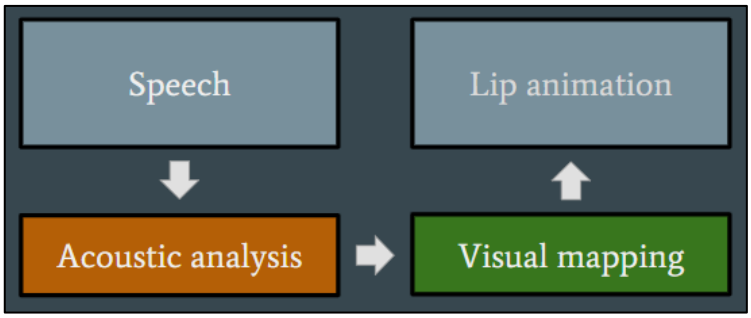
Modul Cleveree berfungsi untuk membangun fitur "*smart*", berupa *automatic summarization* dan parafrasa jawaban. Modul Cleveree dikembangkan dalam bahasa Python dengan framework Flask. Fitur *automatic summarization* dibangun untuk membuat ringkasan pertanyaan yang ditanyakan pengguna kepada *chatbot* Jacob, yang akan digunakan oleh admin Jacob (Octavany, 2019). Fitur parafrasa berfungsi untuk memperbanyak variasi jawaban yang diucapkan oleh *chatbot* Jacob. Modul ini juga dijadikan sebagai *web service* dari *chatbot* Jacob (Octavany, 2019).

3. Modul Vision

Modul Vision berfungsi untuk membangun fitur "penglihatan", yaitu berupa *face recognition*, sehingga *chatbot* Jacob dapat melihat dan mengenali penggunanya. Jacob mengenali pengguna dengan fitur *face recognition* yang ada. Modul ini dikembangkan dalam bahasa Python dengan framework Flask. Fitur *face recognition* pada modul ini dibangun dengan metode *machine learning*, yaitu *Convolutional Neural Network* (CNN). Modul ini juga dijadikan sebagai *web service* dari *chatbot* Jacob (Archilles, 2019).

2.2. Algoritma Rule-based Lip-syncing

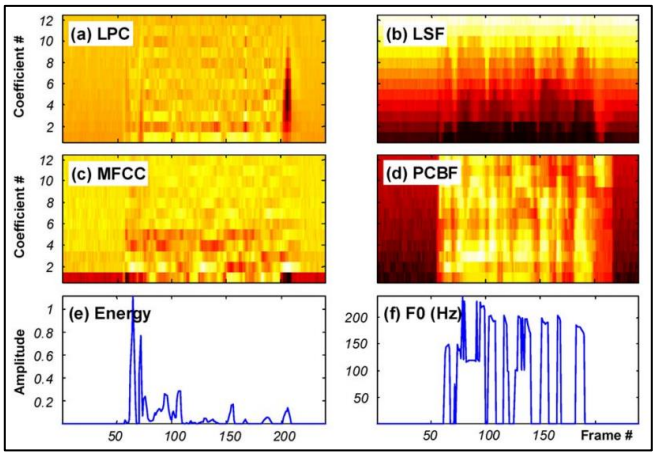
Alur dari algoritma *Rule-based Lip-syncing* yang dikemukakan oleh Gerard Llorach dkk. ditunjukkan pada Gambar 2.2, yaitu dimulai dari tahap *speech*, *acoustic analysis*, *visual mapping*, lalu *lip animation* (Llorach dkk., 2016).



Gambar 2.2. Alur dari algoritma *Rule-based Lip-syncing*
(Llorach dkk., 2016)

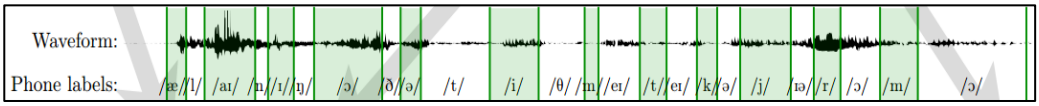
2.2.1. Acoustic analysis

Dalam beberapa dekade terakhir, terdapat dua pendekatan untuk menganalisis *speech*. Salah satunya, fitur akustik, yang disebut *audio-driven articulation*, seperti *fundamental frequency*, Linear Predictive Coding (LPC), dan Mel-Frequency Cepstral Coefficients (MFCCs), antara lain, diekstraksi dari *speech signal* (Kakumanu dkk., 2006). Saat menggunakan *audio-driven articulation*, fitur akustik tambahan dapat diekstraksi, seperti prosodi (menurut KBBI Online, prosodi adalah kajian tentang persajakan, yaitu mengkaji tekanan, matra, rima, irama, dan bait dalam sajak) dan intensitas, yang dapat mengubah lip-sync dan ekspresi wajah secara keseluruhan (Gutierrez-Osuna dkk., 2005). Namun, metode ini mungkin sangat tergantung pada pembicara karena bergantung pada *training data* (Taylor dkk., 2012), dan bisa sulit untuk diterapkan karena membutuhkan keahlian yang luas (Llorach dkk., 2016). Contoh penerapan dari fitur akustik yang diekstraksi dari *speech signal* terdapat pada "A comparison of acoustic coding models for speech-driven facial animation" oleh Praveen Kakumanu dkk. pada 2016, seperti terlihat pada Gambar 2.3.



Gambar 2.3. Contoh hasil dari algoritma yang menerapkan fitur akustik yang diekstraksi dari *speech signal*
(Kakumanu dkk., 2006)

Pendekatan lainnya adalah penggunaan algoritma *phoneme recognition* secara *custom* atau *external*. Menurut KBBI Online, fonem adalah satuan bunyi terkecil yang mampu menunjukkan kontras makna (misalnya /h/ adalah fonem karena membedakan makna kata harus dan arus. *Phoneme-driven articulation* memiliki keuntungan, yaitu tidak tergantung pada pembicara dibandingkan dengan *audio-driven articulation*, tetapi *phoneme-driven articulation* tidak memperhitungkan informasi akustik prosodik dan tidak dapat digunakan secara *real time*, karena fonem berikutnya tidak tersedia sampai dikenali (Llorach dkk., 2016). Contoh penerapan dari *phoneme-driven articulation* terdapat pada "Dynamic units of visual speech" oleh Taylor dkk., seperti terlihat pada Gambar 2.4.



Gambar 2.4. Contoh penerapan *phoneme-driven articulation*

(Taylor dkk., 2012)

Pada penelitian ini, analisis *speech* dilakukan dengan menggunakan *audio-driven articulations*. Algoritma *speech recognition* dengan fitur frekuensi, seperti

MFCC dan LPC, tidak dapat diekstraksi secara real-time. Oleh karena itu, pada penelitian ini digunakan data real-time yang tersedia, berupa *short-term spectrum*, dan energi dari beberapa pita frekuensi dihitung untuk menggerakkan *blend shapes*. Metode yang digunakan adalah *simple energy-based vocal tract model*. Energi formant diperkirakan untuk menghasilkan fitur visual. Formant adalah puncak spektral bunyi yang secara menyeluruh dipengaruhi oleh *vocal tract* (Khairina, 2014). *Smoothed short-term power spectrum density* (st-PSD) diekstrak secara *real time* dengan memproses setiap blok audio dengan cara sampel input di-window dengan Blackman window dan kemudian menghitung Fast Fourier Transform (Llorach dkk., 2016).

Teknik window digunakan untuk merancang FIR (Finite Impulse Response) Filter, yang merupakan filter dari sinyal digital, yang memiliki beberapa tipe, seperti Hanning Window, Hamming Window, Blackman Window, dan Rectangular Window. Blackman Window dipilih untuk penelitian ini karena Blackman Window memiliki keunggulan dalam kinerja dan menunjukkan fungsionalitas terbaik berdasarkan pengujian pada penelitian "Comparative Performance Analysis of Hamming, Hanning and Blackman Window" oleh Prajoy Podder dkk. pada tahun 2014. Fast Fourier Transform (FFT) adalah metode untuk transformasi sinyal suara dalam domain waktu menjadi dalam domain frekuensi, yang berarti proses rekam suara disimpan secara digital berupa gelombang *spectrum* suara yang berbasis frekuensi, yang dapat memudahkan dalam menganalisa *spectrum* frekuensi suara (Sipasulta dkk., 2014). Output st-PSD diperhalus dari waktu ke waktu dengan output sebelumnya dan kemudian dikonversi ke dB (Llorach dkk., 2016).

Smoothing yang dilakukan berdasarkan data yang telah di-*smoothing* sebelumnya dengan Rumus (2.1) (Llorach dkk., 2016).

$$\hat{X}[k] = \tau\, \hat{X}_{-1}[k] + (1-\tau)|X[k]|, for \begin{cases} k = 0, \dots, N-1 \\ 0 < \tau < 1 \end{cases} \quad \dots(2.1)$$

Pada Rumus (2.1), $X[k]$ adalah data domain *complex frequency* yang dihitung. $\hat{X}[k]$ adalah *spectrum* yang telah di-smooth dan τ adalah variabel smoothing yang didefinisikan sendiri, yang bernilai antara 0 hingga 1. Kemudian, hasil tersebut dikonversi ke dB, yang dilambangkan sebagai $Y[k]$, dengan menghasilkan output seperti terlihat pada Rumus (2.2) (Llorach dkk., 2016).

$$Y[k] = 20 \log_{10} \hat{X}[k], for\ k = 0, \dots, N-1 \quad \dots(2.2)$$

Hasil dari $Y[k]$, yang berkisar antara -25dB hingga -160dB, kemudian di-*scale* dan di-*increase* sesuai *sensitivity threshold* yang diterapkan, yang dilambangkan dengan δ (*sensitivity threshold* bernilai 0.5 secara *default*), menggunakan Rumus (2.3) untuk memetakan *dynamic range* ke interval [-0.5,0.5], menghasilkan *smoothed* st-PSD yang telah diproses sebagai $\dot{Y}[k]$ (Llorach dkk., 2016).

$$\dot{Y}[k] = \delta + (Y[k] + 20)/140,\ for\ k = 0, \dots, N-1 \quad \dots(2.3)$$

Hasil dari rumus (2.3) dibagi menjadi pita frekuensi, dan energi dihitung untuk tiap pita frekuensi. Energi dihitung menggunakan *log-scaled data*. Pita-pita frekuensi tersebut didefinisikan secara empiris, dan batas frekuensi atau *frequency bins* (F_{bins}) diskalasi menggunakan *vocal tract length factor* (γ), yang didefinisikan pada Rumus (2.4) (Llorach dkk., 2016).

$$F_{bins} = [0, 500\gamma, 700\gamma, 3000\gamma, 6000\gamma]\ Hz \quad \dots(2.4)$$

Frequency bins tersebut lalu ditransformasi menjadi *frequency data indices* (F_{ind}), sebagai indeks dari setiap *sample* dalam suatu *audio block*, seperti pada

Rumus (2.5), di mana fs sebagai *sampling frequency*, M adalah jumlah frekuensi pembatas dan N adalah jumlah *sample* per *audio block* (Llorach dkk., 2016).

$$F_{ind}[m] = \frac{2N}{fs} F_{bins}[m], for\ m = 0, \dots, M-1 \qquad \dots(2.5)$$

Kemudian, energi dari setiap *bin*, yang dilambangkan sebagai $\mathrm{E}[m]$, dihitung dengan hanya memperhitungkan nilai positif dari st-PSD yang diproses menggunakan Rumus (2.6) (Llorach dkk., 2016).

$$\mathrm{E}[m] = \frac{1}{F_{ind}[m+1] - F_{ind}[m]} \Sigma_{j=F_{ind}[m]}^{F_{ind}[m+1]} \dot{Y}[j], for \begin{cases} \dot{Y}[j] > 0 \\ m = 0, \dots, M-2 \end{cases} \dots(2.6)$$

Bagian penting dari algoritma ini adalah definisi dari *energy bins* dan frekuensi pembatas, di mana yang menjadi fokus adalah *energy* dan frekuensi dari formant setiap huruf hidup (Llorach dkk., 2016). Referensi untuk formant huruf hidup rata-rata terdapat pada Tabel 2.1.

Tabel 2.1. Formant huruf hidup rata-rata (Scherer dkk., 2015)

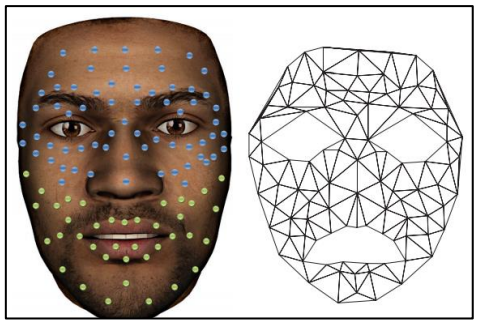
Formant	Gender	a	i	u	e	o
F1	Male	768	342	469	476	497
F1	Female	936	437	519	536	555
F2	Male	1333	2322	1122	2089	910
F2	Female	1551	2761	1225	2530	1035

2.2.2. Visual Mapping

Setelah *speech* dianalisis, fitur audio atau fonem yang diekstraksi dipetakan ke parameter bibir dari karakter virtual, menggunakan viseme (viseme adalah visual fonem, yaitu bentuk bibir ketika mengucapkan suatu fonem (Fadillah dkk., 2015)), *action units*, *codewords*, *control parameters*, dan lain-lain. Parameter visual ini menentukan deformasi wajah dan dapat memiliki tingkat yang berbeda kompleksitas. Sebagai contoh, beberapa peneliti menggunakan MPEG-4 standar, yang mendefinisikan sekitar 66 perpindahan dan rotasi dari titik fitur suatu wajah

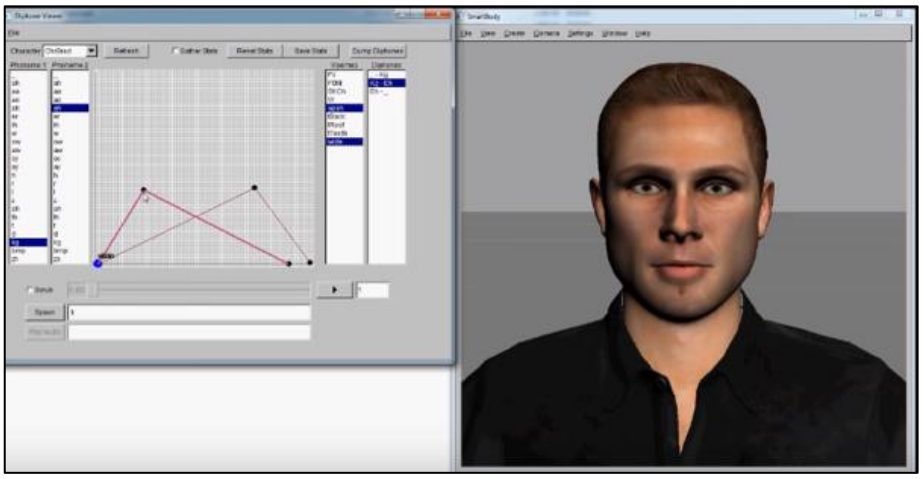
(Tekalp dan Ostermann, 2000), sementara yang lainnya menggunakan satu set kecil *predefined facial deformations* atau *blend shapes* (Wang dkk., 2007).

Terdapat dua macam metode untuk visual mapping, yaitu *data-driven mapping* dan *manual mapping.* Saat menggunakan beberapa parameter visual, memetakan fitur-fitur audio atau fonem secara manual ke parameter-parameter ini akan sangat rumit. Oleh karena itu, *data-driven audio-visual mappings* telah digunakan dalam sebagian besar penelitian terbaru, baik oleh *audio-driven articulation* maupun *phoneme-driven articulation* (Liu dkk., 2011) (Taylor dkk., 2012). Teknik *data-driven* menggunakan korpus yang telah direkam untuk mengekstrak parameter visual yang terkait dengan fitur akustik atau fonem. Hasil yang didapat bisa mencapai akurasi tinggi, tetapi karena teknik ini bergantung pada rekaman korpus, sehingga dapat membuat animasi yang bergantung pada pembicara, karena lintasan bibir berisi informasi tentang identitas seseorang (Jourlin dkk., 1997). Dengan demikian, mengirimkan parameter visual ke karakter dengan fitur visual yang berbeda bisa menjadi rumit, bahkan lebih rumit lagi apabila menggunakan karakter non-manusia. Contoh penerapan *data-driven mapping* terdapat pada penelitian "A Practical Model for Live Speech-Driven Lip-Sync" oleh Li Wei and Zhigang Deng seperti ditunjukkan pada Gambar 2.5.



Gambar 2.5. Facial motion dataset yang menggunakan *data-driven mapping* yang terdapat pada penelitian "A Practical Model for Live Speech-Driven Lip-Sync" (Wei dan Deng, 2015)

Sebagian besar pendekatan yang menggunakan *manual mappings* mengasosiasikan fonem dengan *blend shapes* tertentu, sehingga mereka dapat menangani satu set kecil parameter akustik dan visual yang komprehensif. Pendekatan ini tidak perlu menganalisis korpus yang telah direkam, yang dapat mengurangi jumlah pekerjaan dan bahan yang diperlukan. Sebagai contoh, penelitian "A Practical and Configurable Lip Sync Method for Games" oleh Yuyu Xu dkk. pada 2013 membutuhkan 441 animasi yang dirancang secara manual di antara fonem, dan penelitian "Assembling an expressive facial animation system" oleh A. Wang dkk. pada 2007 menggunakan 16 *blend shapes* untuk memetakan fonem. Contoh penggunaan pendekatan manual mappings terdapat pada penelitian "A Practical and Configurable Lip Sync Method for Games" oleh Yuyu Xu dkk. pada tahun 2013, dapat dilihat pada Gambar 2.6.



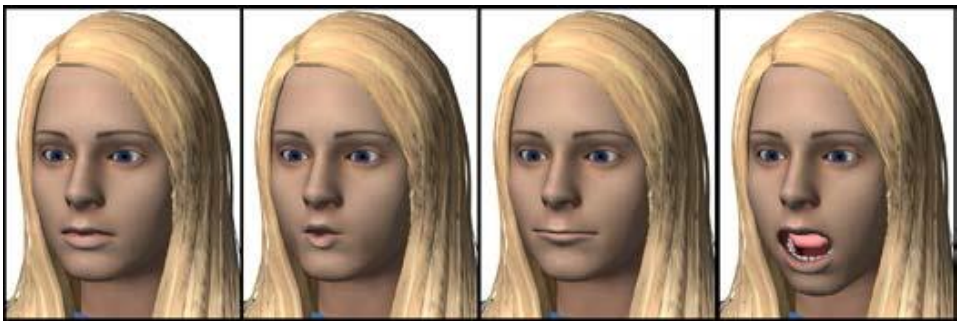
Gambar 2.6. Contoh penggunaan pendekatan manual mappings terdapat pada penelitian "A Practical and Configurable Lip Sync Method for Games"

(Xu dkk., 2013)

Visual mapping yang digunakan pada penelitian ini adalah *manual mapping*. Pemetaan ini bertujuan untuk menghubungkan energi spesifik pita frekuensi yang berbeda dengan parameter visual. Sistem harus "*speaker-independent*", dan

mengaitkan fitur frekuensi spesifik dengan fitur visual terperenci dapat menghasilkan animasi yang salah ketika sistem digunakan oleh pembicara yang berbeda. Fitur visual yang ada menggunakan tiga bentuk campuran.

Algoritma Rule-based Lip-syncing mendefinisikan serangkaian aturan atau *rule* berbasis frekuensi untuk menggerakkan tiga blend shapes yang sesuai dengan konfigurasi bibir yang berbeda, yaitu *kiss*, *mouth open*, dan *lips pressed*, seperti yang terlihat pada Gambar 2.7. Tiga *blend shapes* ini mengontrol masing – masing bukaan horizontal mulut, bukaan vertikal dan volume bibir, dan berbagai konfigurasi bibir yang dapat dicapai. *Mouth open* adalah visual yang paling dasar, semua huruf hidup, seperti /a/, /i/, /u/, /e/, dan /o/ menggunakan *blend shape* ini. *Blend shape model kiss* digunakan untuk membedakan antar huruf hidup, /o/ dan /u/ adalah representasi dari *blend shape* ini. Untuk konsonan frikatif, *blend shape* yang digunakan adalah model *lip pressed*. Seperangkat aturan ini berfungsi untuk menggerakkan *blend shape*, di mana nilai dari masing – masing *blend shape* dihitung menggunakan energi dari daerah frekuensi, menggunakan Rumus (2.7), Persamaan (2.8), dan Rumus (2.9). (Llorach dkk., 2016).



Gambar 2.7. Dari kiri ke kanan: wajah default, kiss blend shape, lips pressed blend shape, mouth open blend shape
(Llorach dkk., 2016)

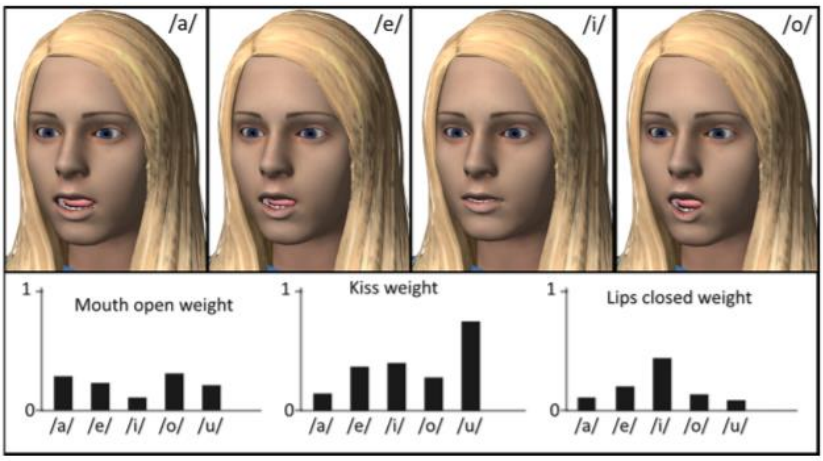
Tiga parameter visual dihitung menggunakan Rumus (2.7), Persamaan (2.8), dan Rumus (2.9) di mana $\mathrm{E[i]}$ adalah energy untuk bin i, dan $\mathrm{BS_{kiss}}$, $\mathrm{BS_{lips}}$, $\mathrm{BS_{mouth}}$ adalah *weight* untuk masing – masing *blend shape* model *kiss*, *lips closed*, dan *mouth open* (Llorach dkk., 2016).

$$\mathrm{BS_{kiss}} = \begin{cases} 1 - 2\mathrm{E}[2], for\ \mathrm{E}[1] \geq 0.2 \\ (1 - 2\mathrm{E}[2])\ .5\mathrm{E}[1], for\ \mathrm{E}[1] < 0.2 \end{cases} \quad \ldots(2.7)$$

$$\mathrm{BS_{lips}} = 3\mathrm{E}[3] \quad \ldots(2.8)$$

$$\mathrm{BS_{mouth}} = 0.8(\mathrm{E}[1] - \mathrm{E}[3]) \quad \ldots(2.9)$$

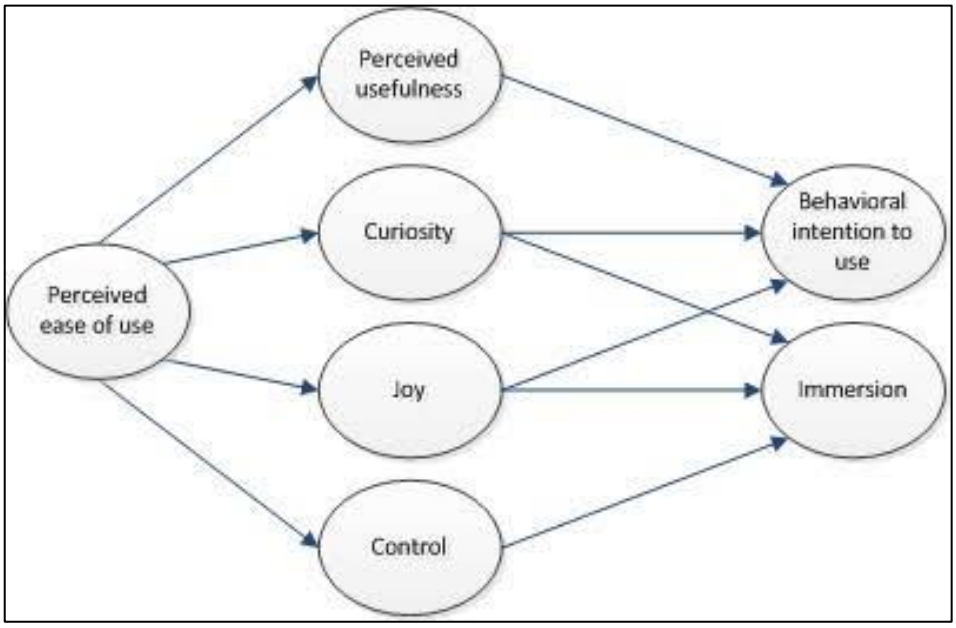
Weight setiap *blend shapes* mempengaruhi bentuk dari *blend shapes* tersebut. Contoh *weight* tiap *blend shape* untuk huruf hidup yang berbeda – beda dapat dilihat pada Gambar 2.8. Pada Gambar 2.8, diperlihatkan contoh apabila pada *audio* terdapat penggunaan huruf vokal yang sesuai dengan *formant* huruf hidup rata-rata seperti pada Tabel 2.1, maka akan didapatkan bentuk mulut yang sesuai dengan gabungan ketiga *blend shapes* yang memiliki *weight* tertentu, yang mempengaruhi bentuk mulut karakter virtual.



Gambar 2.8. Contoh dari *weight* tiap *blend shape* untuk huruf hidup yang berbeda (Llorach dkk., 2016)

2.3. Hedonic Motivation System Adoption Model (HMSAM)

Hedonic Motivation System Adoption Model (HMSAM) adalah suatu model pengujian mengenai tingkat penerimaan pengguna terhadap sistem HMS. *Hedonic Motivation Systems* (HMS) maksudnya adalah sistem yang lebih mementingkan unsur kesenangan dibanding produktivitas (Lowry dkk., 2013). Gambar 2.11 menunjukkan model HMSAM.



Gambar 2.9. Model HMSAM
(Lowry dkk., 2013)

Pada Gambar 2.11, terlihat konstruktor dari model *Hedonic Motivation System Adoption Model* (HMSAM) terdiri dari:

a. *Perceived ease of use* (PEOU)

   Perceived ease of use (PEOU) berfungsi untuk mengukur mengenai tingkat kemudahan penggunaan suatu sistem.

b. *Perceived usefulness* (PU)

   Perceived usefulness (PU) berfungsi untuk mengukur tingkat kinerja penggunaan dari suatu sistem.

c. *Curiosity* (C)

Curiosity (C) berfungsi untuk mengukur rasa ingin tahu yang dirasakan pengguna.

d. *Control* (Co)

Control (Co) berfungsi untuk mengukur persepsi pengguna bahwa dirinya diajak berinteraksi oleh sistem.

e. *Joy* (J)

Joy (J) berfungsi untuk mengukur tingkat rasa senang yang didapat pengguna dari interaksi yang dilakukan dengan sistem.

f. *Behavioral intention to use* (BIU)

Behavioral intention to use (BIU) berfungsi untuk mengukur tingkat keinginan pengguna untuk menggunakan suatu aplikasi.

g. *Focused immersion* (I)

Focused immersion (I) berfungsi untuk mengukur keterlibatan yang dilakukan antara pengguna dengan aplikasi, yang berguna untuk mengetahui seberapa dalam tingkat fokus pengguna ketika menggunakan sistem.

HMSAM memiliki dua aspek utama yaitu *behavioral intention to use* dan *focussed immersion*. *Behavioral intention to use* ditentukan dari aspek *perceived ease of use* (PEOU), *perceived usefulness* (PU), *curiosity* (C), dan *joy* (J), sedangkan *immersion* ditentukan dari aspek *perceived ease of use* (PEOU), *curiosity* (C), *joy* (J), dan *control* (Co). *Immersion* dianggap tidak memiliki pengaruh secara langsung terhadap peningkatan *behavioral intention to use* (Lowry dkk., 2013). Pengukuran dengan model HMSAM ditunjukkan pada Tabel 2.2.

Tabel 2.2. Pengukuran dengan model HSAM (Lowry dkk., 2013)

Construct	Items
Perceived Ease Of Use	**PEOU1.** My interaction with the game was clear and understandable. **PEOU2.** Interacting with the game did not require a lot of my mental effort. **PEOU3.** I found the game to be trouble free. **PEOU4.** I found it easy to get the game to do what I want it to do. **PEOU5.** Learning to operate the game was easy for me. **PEOU6.** It was simple to do what I wanted with the game. **PEOU7.** It was be easy for me to become skillful at using the game. **PEOU8.** I found the game easy to use
Perceived Usefulness	**PU1.** The game decreased my stress. **PU2.** The game helped me better pass time. **PU3.** The game provided a useful escape. **PU4.** The game helped me think more clearly. **PU5.** The game helped me feel rejuvenated.
Curiosity	**CUR1.** This experience excited my curiosity. **CUR2.** This experience made me curious. **CUR3.** This experience aroused my imagination.
Control	**CTL1.** I had a lot of control. **CTL2.** I could choose freely what I wanted to see or do. ***CTL3.** I had little control over what I could do. **CTL4.** I was in control. ***CTL5.** I had no control over my interaction. **CTL6.** I was allowed to control my interaction
Joy	**JOY1.** I found playing the game to be enjoyable. **JOY2.** I had fun using the game. ***JOY3.** Using the game was boring. ***JOY4.** The game really annoyed me. **JOY5.** The game experience was pleasurable. ***JOY6.** The game left me unsatisfied.
Behavioral Intention to Use	**BIU1.** I would plan on using it in the future. **BIU2.** I would intend to continue using it in the future. **BIU3.** I expect my use of it to continue in the future.
Focused Immersion	**FI1.** I was able to block out most other distractions. **FI2.** I was absorbed in what I was doing. **FI3**. I was immersed in the game ***FI4.** I was distracted by other attentions very easily. **FI5.** My attention was not diverted very easily.

Dalam perhitungan skala pengukuran, tanda * diskalakan terbalik, sebagai contoh, "Sangat Tidak Setuju" akan diberikan nilai yang sama dengan "Sangat Setuju”.

2.4. Skala Likert

Skala Likert adalah skala yang berfungsi untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu fenomena sosial (Sugiyono, 2012). Pengukuran dapat dilakukan dengan memberikan pertanyaan kepada responden. Kemudian, responden memberikan jawaban dari enam pilihan jawaban yang ada, yang memiliki nilai jawaban yang berbeda. Skala Likert dalam penelitian ini menggunakan skala pengukuran berjumlah genap, yaitu enam, dengan tujuan agar tidak terdapat jawaban netral yang dapat meningkatkan ambiguitas pada hasil kuesioner. Enam poin skala pengukuran Likert yang digunakan pada penelitian ini, yaitu:

a. Sangat Setuju (SS) memiliki nilai 6.
b. Setuju (S) memiliki nilai 5.
c. Cukup Setuju (CS) memiliki nilai 4.
d. Cukup Tidak Setuju (CTS) memiliki nilai 3.
e. Tidak Setuju (TS) memiliki nilai 2.
f. Sangat Tidak Setuju (STS) memiliki nilai 1.

Pada Rumus (2.10), ditentukan persentase nilai yang didapatkan dari pengambilan data kuesioner yang telah dilakukan (Sugiyono, 2012). Rumus (2.10) digunakan untuk kode pertanyaan pada HMSAM yang tidak memiliki tanda *, sedangkan Rumus (2.11) digunakan untuk kode pertanyaan yang memiliki tanda *. Pada Rumus (2.10) dan Rumus (2.11), PS berarti persentase skor SS berarti sangat setuju, S berarti setuju, CS berarti cukup setuju, CTS berarti cukup tidak setuju, TS berarti tidak setuju, dan STS berarti sangat tidak setuju (Octavany, 2019).

$$\text{PS} = \frac{(SS * 6) + (S * 5) + (CS * 4) + (CTS * 3) + (TS * 2) + (STS * 1)}{(6 * Jumlah\ Responden)} \quad \ldots(2.10)$$

$$PS = \frac{(STS * 6) + (TS * 5) + (CTS * 4) + (CS * 3) + (S * 2) + (SS * 1)}{(6 * Jumlah\ Responden)} \quad \ldots(2.11)$$

2.5. Rumus Lemeshow

Apabila jumlah populasi tidak diketahui secara pasti, maka perhitungan sampel dapat dilakukan dengan menggunakan pendekatan rumus Lemeshow. Rumus Lemeshow didefinisikan pada Rumus (2.12), dengan n adalah jumlah sampel, z adalah skor z pada kepercayaan tertentu, p adalah maksimal estimasi, dan d adalah tingkat kepercayaan (Riyanto, 2020).

$$n = \frac{z_{1-\alpha/2}^{2} p(1-p)}{d^2} \quad \ldots(2.12)$$