

## ANALISIS DAN IMPLEMENTASI LIP READING (GERAKAN BIBIR) MENJADI TEXT PADA PENGGUNAAN BAHASA INGGRIS

Dayanni Vera Versanika<sup>1</sup>, Faiqunisa<sup>2</sup>, Julia Arisanda Hidayat<sup>3</sup>

<sup>1,2,3</sup>STMIK BANDUNG

Sekolah Tinggi Manajemen dan Informatika Bandung

JL. Cikutra No.113, Bandung 40124, INDONESIA

Contact address:

<sup>1</sup>[dayanniveraversanika@gmail.com](mailto:dayanniveraversanika@gmail.com)

---

### ABSTRAK

Dalam era globalisasi yang terus berkembang, komunikasi menjadi semakin vital. Kemajuan teknologi membuka peluang baru untuk berkomunikasi secara internasional, mendorong kebutuhan akan kemampuan bahasa asing. Bahasa Inggris, sebagai bahasa global, memiliki peran sentral dalam komunikasi global. Namun, belajar Bahasa Inggris memunculkan tantangan, terutama dalam memahami hubungan antara pengucapan dan penulisan kata. Hal ini menjadi fokus utama penelitian ini. Penelitian ini mengusulkan metode baru berbasis *deep learning* dan arsitektur *Convolutional Neural Network (CNN)* untuk menerjemahkan gerakan bibir menjadi teks dalam Bahasa Inggris. Penggunaan teknologi ini diharapkan dapat membantu individu yang mengalami kesulitan memahami teks Bahasa Inggris saat diucapkan, meningkatkan kemampuan komunikasi, dan memungkinkan partisipasi aktif dalam situasi di mana pemahaman Bahasa Inggris lisan sangat penting. Hasil eksperimen menunjukkan potensi sistem ini dalam mendeteksi gerakan bibir pada video dan menerjemahkannya menjadi teks, meskipun masih menghadapi kendala dalam variasi posisi mulut, wajah, dan perbedaan pengucapan. Meskipun demikian, implementasi teknologi ini dapat menjadi langkah awal yang signifikan untuk mengatasi kendala bahasa dan memfasilitasi komunikasi yang lebih baik di era globalisasi ini.

**Kata Kunci :** Komunikasi global, Bahasa Inggris, Menerjemahkan gerakan bibir, *Deep learning*, *Convolutional Neural Network (CNN)*, *Lip reading*, *Artificial Intelligence*, *Tensorflow*

---

### ABSTRACT

*In the rapidly evolving era of globalization, communication plays a crucial role. Understanding English, as a global language, is increasingly important for effective communication. Challenges arise when individuals need to comprehend the written form of English based on diverse pronunciations, especially for English language learners. The differences between pronunciation and written forms, along with various accents and dialects, add complexity. This research proposes an innovative method based on deep learning and Convolutional Neural Network (CNN) to translate lip movements into English text. The use of this technology is expected to assist individuals struggling to comprehend English text when spoken, enhancing communication skills, and enabling effective participation in critical situations that require oral English comprehension. Experimental results demonstrate the potential of this system in detecting lip movements in videos and translating them into text, although it still faces challenges related to variations in mouth and face positions, as well as pronunciation differences. The implementation of this technology represents a significant initial step in overcoming language barriers and facilitating improved communication in this era of globalization.*

**Keywords :** Global communication, English, lip movement translation, deep learning, Convolutional Neural Network (CNN), Lip Reading, Artificial Intelligence, Tensorflow.

---

### 1. Pendahuluan

Di era globalisasi yang semakin berkembang saat ini, peran komunikasi menjadi semakin vital. Era kemajuan ilmu pengetahuan dan teknologi yang semakin pesat, semakin membuka kesempatan untuk berkomunikasi secara internasional. Pelaksanaan pasar bebas menuntut Bangsa Indonesia memiliki kompetensi yang kompetitif dalam berbagai bidang. Indonesia tidak bisa lagi hanya mengandalkan sumber daya alam dan kemampuan fisik untuk mencapai

kesejahteraan bangsanya tetapi harus lebih mengandalkan sumber daya manusia yang profesional. Salah satu persyaratan mutlak untuk mencapainya adalah dengan memiliki kemampuan berkomunikasi yang baik.

Salah satu kendala yang dihadapi adalah saat seseorang ingin mengetahui cara menulis kata-kata dalam Bahasa Inggris sesuai dengan pengucapannya. Hal ini terutama relevan untuk orang yang sedang belajar Bahasa Inggris atau belum memiliki pemahaman yang baik mengenai

aturan penulisan dalam Bahasa Inggris. Ini berhubungan dengan perbedaan antara cara pengucapan dan penulisan dalam Bahasa Inggris, serta variasi dalam aksen dan dialek. Untuk mengatasi hal ini, diperlukan sumber daya seperti kamus pengucapan atau sistem yang menyediakan informasi tentang ejaan yang sesuai berdasarkan ucapan.

Untuk menghadapi tantangan ini, diperlukan pengembangan algoritma dan teknologi yang maju, memanfaatkan kecerdasan buatan dan pembelajaran mesin. Diperlukan data pelatihan yang luas agar sistem dapat dilatih untuk mengenali berbagai pola gerakan bibir dan menghubungkannya dengan kata-kata yang sesuai. Dengan kehadiran sistem *lip reading* yang akurat, individu yang mengalami kesulitan dalam memahami teks Bahasa Inggris saat diucapkan akan dapat mengandalkan teknologi ini untuk memperoleh informasi dengan efektif. Hal ini akan meningkatkan kemampuan komunikasi mereka dan memungkinkan mereka berpartisipasi dalam berbagai situasi di mana pemahaman Bahasa Inggris lisan sangat diperlukan.

Komunikasi dengan berbahasa Indonesia memungkinkan lawan bicara memahami apa yang dibicarakan dan juga mengetahui seperti apa bentuk tulisan yang diucapkan. Sedangkan untuk berbahasa Inggris ada yang dapat memahami ujaran tersebut tetapi tidak mengetahui seperti apa tulisannya jika sebuah ucapan tersebut menjadi text.

Sebagai bahasa global, bahasa Inggris memegang fungsi dan peran yang sangat besar. Salah satu implikasi yang terlihat adalah semakin banyak orang berusaha belajar/ menguasai bahasa Inggris dengan baik. Dalam bidang pendidikan misalnya. Untuk menghadapi persaingan global, bahasa Inggris dikenalkan kepada siswa lebih dini. Banyak siswa sekolah dasar (SD) bahkan taman kanak-kanak (TK) mulai mempelajari bahasa Inggris. Pemakaian bahasa Inggris juga mulai banyak digunakan di bidang non pendidikan misalnya ekonomi dan bisnis.

Berdasarkan hal tersebut maka dalam Skripsi ini penulis membuat sebuah judul skripsi “ANALISIS DAN IMPLEMENTASI LIP READING (GERAKAN BIBIR) MENJADI TEXT PADA PENGGUNAAN BAHASA INGGRIS”. Dengan gagasan untuk menganalisis Gerakan bibir menjadi sebuah text dengan berbahasa inggris, dalam proses ini juga dibatasi hanya melakukan memilih video yang tersedia dan akan di konversikan menjadi sebuah text.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka penulis merumuskan beberapa rumusan masalah yaitu :

1. Bagaimana cara proses deteksi gerak bibir yang diterjemahkan menjadi sebuah text?

2. Bagaimana mengetahui sebuah tulisan yang diucapkan oleh individu dengan Bahasa Inggris?
3. Video apa saja yang dapat digunakan pada *lip reading* ini?
4. Bagaimana proses deteksi gerak bibir jika menggunakan video yang bukan dari dataset?

## 1.3 Batasan Masalah

Agar masalah yang dibahas tidak menyimpang dari tujuan, maka diperlukan suatu batasan masalah, sebagai berikut:

1. Proses diterjemahkan saat melakukan pemilihan video berbahasa inggris.
2. Proses konversi tidak menggunakan kamera secara langsung.
3. Video yang digunakan hanya video yang tersedia di dataset saja.
4. Video lain yang dapat digunakan pada proses deteksi gerak bibir harus memiliki format yang sama dengan video yang ada di dataset.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang diuraikan diatas, maka tujuan dari penelitian ini adalah sebagai berikut :

1. Melakukan analisis fitur gerak bibir yang akan dikonversi menjadi text.
2. Mengidentifikasi model yang digunakan dalam hasil deteksi gerak bibir (*Lip Reading*).
3. Mengidentifikasi jenis video yang dapat digunakan pada *Lip Reading*.

## 1.5 Metodologi Penelitian

Dalam merancang sebuah alat tersebut digunakan beberapa metode pengumpulan data diantara-Nya:

### a. Studi Literatur

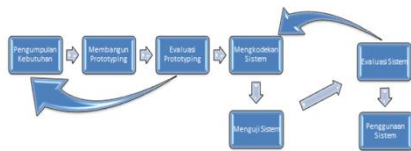
Studi literatur berisi kegiatan pengumpulan dan pengkajian dasar teori yang terpercaya untuk menunjang penulisan skripsi ini. Literatur dapat bersumber dari paper, jurnal, buku, artikel, maupun website yang menyediakan informasi terkait.

### b. Rancang Bangun System

Pada tahap ini penulis melakukan rancang bangun sistem yaitu merancang desain implementasi sistem yang akan diterapkan pada *lip reading* (deteksi gerak bibir) pada Bahasa Inggris.

### 1.5.1 Metode Pengembangan Perangkat Lunak

Metode pembangunan perangkat lunak yang digunakan dalam penelitian ini adalah metode *prototype*. Metode *prototype* dipilih Pendekatan ini didasarkan pada konsep model yang berfungsi (*working model*). Tujuan utama dari metode *prototype* yaitu mengembangkan model awal menjadi sistem akhir dengan cepat dan dengan biaya yang lebih rendah.



Gambar 1.1 Model Prototype

### 1. Pengumpulan Kebutuhan

Tahapan awal dalam metode *prototype* adalah mengidentifikasi semua perangkat dan masalah yang ada. Proses analisis dan identifikasi kebutuhan secara menyeluruh dari sistem merupakan langkah penting dalam metode *prototype*. Dengan demikian, langkah-langkah dan permasalahan yang akan dihadapi dapat ditentukan dan diatasi. Pengumpulan kebutuhan menjadi tahap yang sangat krusial dalam proses ini.

### 2. Membangun Prototype

Langkah berikutnya dalam metode *prototype* yaitu pembangunan *prototype* yang berorientasi pada kepuasan pelanggan, seperti menciptakan input dan output hasil sistem. Saat ini, hanya prototipe yang dibuat, sementara akan ada tindak lanjut yang perlu dikerjakan di kemudian hari.

### 3. Evaluasi Prototype

Sebelum melanjutkan ke langkah berikutnya, tahap ini harus diperiksa dengan cermat karena merupakan hal yang wajib dilakukan. Tahap 1 dan 2 sangat menentukan keberhasilan dan merupakan proses yang sangat krusial. Jika terdapat kekurangan atau kesalahan pada tahap 1 dan 2, maka akan menjadi sulit untuk melanjutkan ke langkah-langkah selanjutnya.

### 4. Mengkodekan Sistem

Sebelum memulai proses pengkodean, penting untuk memahami penggunaan bahasa pemrograman. Proses ini menjadi tantangan tersendiri karena melibatkan penerapan kebutuhan dalam bentuk kode program.

### 5. Menguji Sistem

Setelah tahap pengkodean, langkah selanjutnya yaitu melakukan pengujian (*testing*). Terdapat berbagai cara untuk melakukan pengujian, seperti menggunakan pendekatan *white box* atau *black box*. Pengujian dengan pendekatan *white box* berfokus pada pengujian kode program, sedangkan pendekatan *black box* lebih berorientasi pada pengujian fungsi-fungsi tampilan dan kecocokan aplikasi.

## 2. LANDASAN TEORI

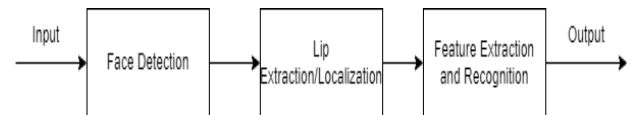
### 2.1 Lip Reading

Dikemukakan oleh (Data Science Society, 2022), *Lip Reading* adalah proses mengartikan teks dari gerakan mulut pembicara. Hal ini memainkan peran penting dalam komunikasi manusia dan pemahaman bicara. *Lip Reading* merupakan tugas yang sulit bagi manusia, terutama tanpa adanya konteks. Sebagian besar gerakan bibir saat membaca bibir dan kadang-kadang lidah dan gigi bersifat samar dan sulit dijelaskan tanpa adanya konteks.[1]

Sedangkan menurut (Sooraj, Hardhik, Nishanth, Sandesh, Sashidhar, 2020),

mengungkapkan bahwa *Lip reading* merupakan teknik mengenali kata-kata hanya melalui representasi visual gerakan bibir.[2]

Dari definisi yang telah dijelaskan sebelumnya, bahwa *lip reading* atau pembacaan gerak bibir adalah proses mengartikan teks dari gerakan mulut pembicara. Ini merupakan tugas yang sulit bagi manusia, terutama tanpa adanya konteks, karena sebagian besar gerakan



bibir saat membaca bibir bersifat samar dan sulit dijelaskan tanpa konteks.

Gambar 2.1 Blok Diagram Lip Reading

### 2.2 Face Detection

Menurut (Yustiwati, Ratna, Evelina, Rasyad, 2018), *Face Detection* adalah proses identifikasi manusia menggunakan wajah. Dalam pendeteksian wajah, teknologi ini hanya deteksi wajah dan mengabaikan hal lain seperti bangunan pohon dan sejenisnya.[3]

Dijelaskan kembali oleh (S. Yang et al., 2016), *Face detection* atau Deteksi wajah adalah menentukan keberadaan wajah dalam gambar dan, jika ada, mencari lokasi gambar dan bagian masing-masing wajah. Deteksi wajah merupakan hal yang mudah bagi manusia, namun sangat sulit untuk komputer.[4]

Dari penjelasan yang diberikan, dapat disimpulkan bahwa deteksi wajah atau *face detection* adalah proses mengidentifikasi dan mencari keberadaan wajah manusia dalam suatu gambar. Teknologi *face detection* memiliki kemampuan untuk menentukan lokasi dan bagian-bagian dari setiap wajah yang terdapat dalam gambar tersebut. Meskipun deteksi wajah merupakan tugas yang mudah dilakukan secara alami oleh manusia, namun hal ini menjadi sangat sulit bagi komputer. Dalam proses pendeteksian wajah, teknologi *face detection* fokus pada deteksi wajah saja dan mengabaikan elemen-elemen lain seperti bangunan, pohon, dan objek lainnya dalam gambar. Penelitian yang dilakukan oleh Yustiwati, Ratna,

aman menggunakan *website* STMIK Bandung sebagai media untuk pembelajaran jarak jauh. Hal ini dilaku Evelina, dan Rasyad pada tahun 2018 serta penelitian yang dikutip oleh S. Yang et al. pada tahun 2016 memberikan pemahaman yang lebih jelas tentang konsep dan tantangan yang terkait dengan *face detection*. Oleh karena itu, *face detection* menjadi teknologi penting dalam berbagai aplikasi yang membutuhkan identifikasi dan analisis wajah manusia.

### 2.3 Kecerdasaan Buatan

Dikemukakan oleh (Andreas, 2019), Kecerdasaan buatan sebagai “kemampuan sistem untuk menafsirkan data eksternal dengan benar, untuk belajar dari data tersebut, dan menggunakan pembelajaran tersebut guna mencapai tujuan dan tugas tertentu

melalui adaptasi yang fleksibel". [5]

Menurut (Muchlisin, 2020), Kecerdasan buatan atau *Artificial Intelligence (AI)* adalah teknologi di bidang ilmu komputer yang mensimulasikan kecerdasan manusia ke dalam mesin (komputer) untuk menyelesaikan berbagai persoalan dan pekerjaan seperti dan sebaik yang dilakukan manusia. [6]

Dengan penjelasan yang telah dikemukakan diatas, disimpulkan bahwa *Artificial Intelligence (AI)*, atau yang dikenal sebagai kecerdasan buatan, merujuk pada kemampuan sistem komputer untuk secara akurat menganalisis data eksternal, melakukan pembelajaran dari data tersebut, dan mengaplikasikan pengetahuan yang diperoleh untuk mencapai tujuan dan tugas tertentu dengan tingkat fleksibilitas. AI merupakan suatu teknologi dalam bidang ilmu komputer yang berusaha meniru kemampuan berpikir dan tindakan manusia dengan mengintegrasikannya ke dalam mesin (komputer) untuk menyelesaikan berbagai masalah dan pekerjaan sebagaimana yang bisa dilakukan oleh manusia.

Kecerdasan buatan memiliki beberapa kelebihan, antara lain:

- Kecerdasan buatan lebih permanen, sedangkan kecerdasan alami dapat mengalami perubahan, ini dikarenakan faktor sifat manusia yang mudah lupa.
- Kecerdasan buatan lebih mudah untuk diduplikasikan dan disebar.
- Kecerdasan buatan lebih konsisten.
- Kecerdasan buatan lebih murah daripada kecerdasan alami.

Dikemukakan (Simarmata, 2010), Kecerdasan buatan mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktivitas manusia yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman Bahasa alami dan sebagainya. Teknologi kecerdasan buatan dapat dipelajari dalam berbagai bidang-bidang seperti robotika (*robotics*), penglihatan komputer (*computer vision*), pengolahan bahasa alami (*natural language processing*), pengenalan pola (*pattern recognition*), sistem syaraf buatan (*artificial neural system*), pengenalan suara (*speech recognition*) dan sistem pakar (*expert system*).[7]

## 2.4 Open CV

Menurut (Budiarjo, 2020), OpenCV adalah *library* yang menyediakan fungsi-fungsi untuk pemrograman *computer vision* dalam waktu nyata. OpenCV merupakan perangkat lunak *open-source* yang dapat digunakan untuk tujuan akademis dan komersial. *Library* ini menyediakan antarmuka untuk Bahasa Pemrograman C, C++, Python, dan Java yang dapat dijalankan di sistem operasi Windows, Mac, Linux, dan Android. [8]

## 2.5 Tensorflow

Berdasarkan pendapat (Nurfita, 2018), Tensorflow merupakan antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan untuk mengeksekusi perintah dengan menggunakan

informasi yang dimiliki tentang objek tersebut atau target yang dikenali serta dapat membedakan objek satu dengan objek lainnya. Tensorflow memiliki fitur untuk menjalankan pelatihan model menggunakan *Central Processing Unit (CPU)* dan pelatihan model *Graphic Processing Unit (GPU)* yang memiliki waktu pelatihan yang lebih cepat dibanding pelatihan model CPU. Namun dalam implementasi ini menggunakan CPU.[10]

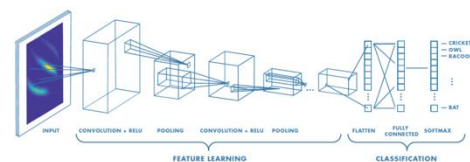
## 2.6 Convolutional Neural Network (CNN)

Mengutip (Ghosh, Sufian, Sultana, Charabarti, & De, 2020), CNN merupakan tipe spesial dari *multilayer neural network* atau arsitektur *deep learning* yang terinspirasi dari persepsi visual makhluk hidup. [11]

Berdasarkan pendapat (Benny, 2020), *Convolutional Neural Network (CNN)* adalah salah satu algoritma *deep learning* yang dirancang untuk mengolah data dua dimensi. CNN biasanya digunakan untuk mempelajari dan mendeteksi *feature* pada sebuah gambar. [12]

Berdasarkan penarikan kesimpulan sebelumnya, *CNN (Convolutional Neural Network)* merupakan sebuah tipe spesial dari *multilayer neural network* atau arsitektur *deep learning* yang terinspirasi oleh kemampuan persepsi visual yang dimiliki oleh makhluk hidup. CNN dirancang secara khusus untuk memproses data dengan dimensi dua, dan sering kali digunakan untuk mempelajari serta mendeteksi fitur-fitur yang terdapat dalam gambar.

Gambar 2.5 Tahapan Proses CNN



Seperti yang dapat kita lihat pada Gambar diatas, bahwa tahapan proses pada CNN terdiri dari dua bagian utama, yakni *feature learning* dan *classification*. Berikut ini adalah penjelasan dari kedua bagian tersebut.

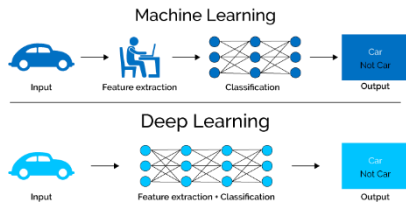
## 2.7 Deep Learning

Dalam pandangan (Miceli, 2018), *Deep Learning* merupakan sub-bidang dari *Machine Learning* yang bekerja dengan melakukan representasi dari data pada lapisan pembelajaran yang berlapis-lapis hingga representasi menjadi lebih bermakna.[13]

Disampaikan oleh (Wayan, 2018), *Deep Learning* merupakan sebuah algoritma *neural network* yang menggunakan metadata sebagai input dan mengolah input tersebut menggunakan sekumpulan fungsi transformasi *non-linier* yang ditata berlapis-lapis dan mendalam. [14]

*Deep Learning* dapat dikatakan sebagai turunan dari *machine learning*, dimana metode ini terdiri banyak tingkatan proses informasi *non-linear* dan abstraksi untuk dapat melakukan *supervise*, *unsupervised learning*, dan

*representation*, klasifikasi, dan pengenalan pola.



Gambar 2.6 Perbedaan ML dan DL

### 3. ANALISIS SISTEM

#### 3.1 Deskripsi Sistem

*TalkClarity* adalah sistem berbasis website yang dikembangkan untuk membantu masyarakat mengenali cara penulisan Bahasa Inggris ketika diucapkan oleh seseorang. Saat ini, website *TalkClarity* membatasi pemilihan video yang digunakan, yaitu hanya video yang tersedia di dataset saja yang dapat digunakan. Meskipun begitu sistem ini akan semakin membantu seseorang dalam menulis Bahasa Inggris.

#### 3.2 Analisis Prosedur

Analisis prosedur pada website *TalkClarity* (*Lip Reading*) ini digambarkan bagaimana proses sistem *lip reading* ini berjalan. Berikut adalah beberapa proses yang teridentifikasi dari tahapan analisis :

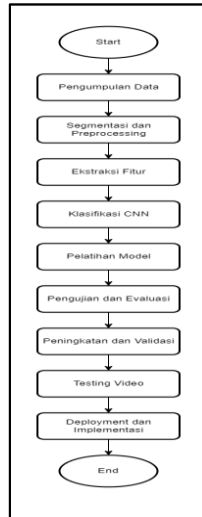
1. Proses pengumpulan data, Pada proses pengumpulan data untuk pembuatan model *Lip Reading* ini yaitu mencari sumber data yang sesuai. Salah satu cara untuk mendapatkan data adalah melalui repositori Github, di mana banyak proyek *open-source*, seperti yang saya gunakan yaitu proyek "LipNet" yang menyediakan dataset *lip reading*.
2. Proses preprocessing data, Penting untuk menjalani serangkaian langkah *preprocessing* pada data yang akan digunakan untuk membuat model *lip reading*. Proses ini bertujuan membersihkan, memisahkan, dan mengonversi dataset menjadi format yang tepat sebelum digunakan sebagai input dalam melatih model. Dengan demikian, dataset *lip reading* dapat diproses lebih lanjut dengan efisien.
3. Proses ekstraksi fitur oleh grid corpus, *Grid corpus* menggambarkan visual dari pergerakan bibir manusia saat berbicara, berupa rangkaian *frame* gambar yang menampilkan berbagai konfigurasi bibir yang berubah bentuk selama percakapan. Pada *lip reading*, proses ekstraksi fitur dari *grid corpus* melibatkan teknik untuk mengidentifikasi fitur-fitur penting dari setiap *frame* gambar dalam *grid corpus* tersebut. Fitur-fitur ini kemudian digunakan sebagai input dalam proses pelatihan model *lip reading*.

#### 4. Proses klasifikasi CNN

Langkah-langkah dalam proses klasifikasi dengan *Convolutional Neural Network (CNN)* melibatkan pelatihan dan penggunaan model untuk memprediksi kelas atau label dari data input. CNN merupakan jenis arsitektur jaringan saraf yang sangat efisien dalam tugas-tugas visi komputer, termasuk klasifikasi gambar.

5. Proses pelatihan model, Pelatihan model CNN dalam *lip reading* melibatkan serangkaian langkah berulang dengan tujuan mengoptimalkan parameter model sehingga mampu mengenali dan menginterpretasikan gerakan bibir manusia secara akurat. Proses pelatihan ini memerlukan iterasi yang hati-hati dan pemilihan parameter yang tepat untuk menghasilkan model yang efektif dan dapat diandalkan.
6. Proses pengujian dan evaluasi model, Pentingnya proses pengujian dan evaluasi model tidak dapat diabaikan, karena peranannya yang krusial dalam mengevaluasi kemampuan model untuk melakukan tugas klasifikasi dengan akurasi. Hasil evaluasi ini memberikan wawasan tentang kemampuan model dalam situasi penggunaan nyata, serta membantu meningkatkan performa model melalui penyesuaian yang dibutuhkan.
7. Proses peningkatan dan validasi, Tahap peningkatan dan validasi ini sebuah proses berkelanjutan yang bertujuan menciptakan model yang lebih unggul dan dapat diandalkan. Selama tahap ini, model disesuaikan berdasarkan pemahaman yang lebih mendalam tentang data dan tantangan yang dihadapinya, sehingga hasil prediksi yang dihasilkan menjadi lebih akurat dan efektif dalam aplikasi kehidupan nyata.
8. Proses testing video, Proses pengujian video pada model *lip reading* memungkinkan pengukuran performa model dalam situasi realistik, di mana video mencakup gerakan bibir dalam urutan waktu yang menantang untuk pengenalan dan interpretasi ucapan. Evaluasi yang teliti dan analisis hasil yang tepat akan berkontribusi dalam menghasilkan model *lip reading* yang lebih dapat diandalkan dan efisien.
9. Proses deployment dan implementasi, Proses *deployment* dilakukan dengan tahap pertama yaitu membuat folder baru untuk *deployment*, yang dimana isinya adalah dataset, model, beberapa file python untuk menunjang proses *deployment*. *Deployment lip reading* pada website ini menggunakan streamlit, dikarenakan video asli di dataset memiliki format mpg maka dilakukan konversi terlebih dahulu menjadi format mp4, setelah itu dilakukan proses pendeklarasian kembali mode, *load data*, dan juga *load alignment*

untuk proses deteksi gera bibir. Setelah itu membuat file *deployment* streamlit yang isinya adalah video yang dipilih, proses deteksi, decode deteksi sebagai token, dan juga hasil teks yang sesuai dengan video tersebut.



Gambar 3.1 Proses Lip Reading

### 3.2.1 Pengumpulan Dataset

Dalam langkah-langkah mengumpulkan dataset *lip reading*, penulis mencoba mencari dataset pada github dengan mencari repositori yang relevan dengan *lip reading*. Repositori yang menjadi acuan saya adalah <https://github.com/nicknochnack/LipNet.git> dan dataset <https://drive.google.com/uc?id=1YlvpDLix3S-U8fd-ggRwPcWXAXm8JwjL>. Setelah menemukan repositori yang sesuai, penulis memastikan apakah dataset sudah sesuai dengan kebutuhan. Setelah dipastikan dataset tersebut diunduh. Setelah diunduh penulis memahami struktur dataset terlebih dahulu sebelum masuk ke tahap selanjutnya.

### 3.2.2 Praproses Data

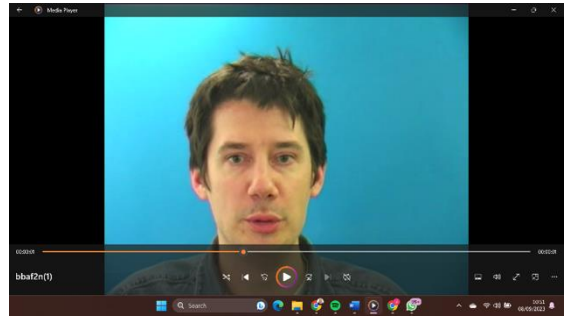
Praproses yaitu sebuah langkah-langkah yang dilakukan untuk mengubah dan mempersiapkan data mentah menjadi bentuk yang sesuai untuk digunakan dalam pembuatan model atau analisis lebih lanjut. Berikut adalah tahapan praproses data:

1. Data Alignment : Pada isi data alignment video menyajikan informasi yang digunakan untuk menyelaraskan transkripsi teks dengan waktu pada video. Seperti pada contoh video *bbaf2n.mp4* dan *bbaf2n.txt* sebagai alignment, terdapat isi dari alignment tersebut :

```

0 23750 sil
23750 29500 bin
29500 34000 blue
34000 35500 at
35500 41000 f
41000 47250 two
47250 53000 now
53000 74500 sil
  
```

Dan cuplikan video pada saat mengucapkan "blue"



Gambar 3.2 Cuplikan video

Pada data alignment diatas, dijelaskan :

- a. "0 23750 sil": Pada rentang waktu 0 hingga 23.750 milidetik dalam video, tidak ada ucapan yang terjadi, dan ini diwakili oleh label "sil" yang merupakan singkatan dari "silent" atau "diam."
- b. "23750 29500 bin": Dari 23.750 milidetik hingga 29.500 milidetik dalam video, kata "bin" diucapkan atau muncul di layar.
- c. "29500 34000 blue": Dari 29.500 milidetik hingga 34.000 milidetik dalam video, kata "blue" diucapkan atau muncul di layar.
- d. "34000 35500 at": Dari 34.000 milidetik hingga 35.500 milidetik dalam video, kata "at" diucapkan atau muncul di layar.
- e. "35500 41000 f": Dari 35.500 milidetik hingga 41.000 milidetik dalam video, kata "f" diucapkan atau muncul di layar.
- f. "41000 47250 two": Dari 41.000 milidetik hingga 47.250 milidetik dalam video, kata "two" diucapkan atau muncul di layar.
- g. "47250 53000 now": Dari 47.250 milidetik hingga 53.000 milidetik dalam video, kata "now" diucapkan atau muncul di layar.
- h. "53000 74500 sil": Dari 53.000 milidetik hingga 74.500 milidetik dalam video, tidak ada ucapan yang terjadi, dan ini diwakili oleh label "sil" lagi.

2. Memuat video: Fungsi `load_video` digunakan untuk memuat video, mengubahnya menjadi citra skala abu-abu, dan memotong bagian tertentu dari setiap frame. Hal ini dilakukan untuk mengubah data video mentah menjadi format yang lebih sesuai untuk model jaringan saraf.

```

def load_video(path:str) -> List[float]:

    cap = cv2.VideoCapture(path)
    frames = []
    for _ in range(int(cap.get(cv2.CAP_PROP_FRAME_COUNT))):
        ret, frame = cap.read()
        frame = tf.image.rgb_to_grayscale(frame)
        frames.append(frame[190:236,80:220,:])
    cap.release()

    mean = tf.math.reduce_mean(frames)
    std = tf.math.reduce_std(tf.cast(frames, tf.float32))
    return tf.cast((frames - mean), tf.float32) / std
  
```

Gambar 3.3 Proses Load Video

3. Memuat alignment: Fungsi `load_alignments` memuat data alignment teks dan mengonversinya menjadi bentuk numerik menggunakan `char_to_num`. Diperlukan untuk mengubah data teks mentah menjadi format yang dapat digunakan dalam model. `load_alignments` memuat *alignment* teks dari path yang diberikan, file teks dibaca baris per baris, dan kata-kata yang bukan 'sil' (karakter pengganti) diambil sebagai token. Token-token ini kemudian diubah menjadi angka menggunakan lapisan `char_to_num`.

```
def load_alignments(path:str) -> list[str]:
    with open(path, 'r') as f:
        lines = f.readlines()
        tokens = []
        for line in lines:
            line = line.split()
            if line[2] is 'sil':
                tokens = ["tokens," ,line[2]]
            else:
                tokens = ["tokens," ,line[2]]
        return char_to_num(tf.reshape(tf.strings.unidecode_split(tokens, input_encoding='UTF-8'), (-1,))[1:])
```

Gambar 3.4 Proses Load Alignment

4. Memuat data: Fungsi `load_data` menggabungkan data video dan alignment teks yang sudah diproses sebelumnya menjadi satu unit data yang lengkap dan dapat digunakan untuk pelatihan model.

```
def load_data(path: str):
    path = bytes.decode(path.numpy())
    #file_name = path.split('/')[-1].split('.')[0]
    # File name splitting for windows
    file_name = path.split('/')[-1].split('.')[0]
    video_path = os.path.join('data', 's1', f'{file_name}.mpg')
    alignment_path = os.path.join('data', 'alignments', 's1', f'{file_name}.align')
    frames = load_video(video_path)
    alignments = load_alignments(alignment_path)

    return frames, alignments
```

Gambar 3.5 Load Data

### 3.2.3 Arsitektur Model CNN

Arsitektur model CNN merujuk pada rancangan dan struktur sistem komputasi yang dimanfaatkan dalam *machine learning* dan *deep learning* untuk mengolah data serta melakukan prediksi atau pengambilan keputusan. Arsitektur model yang digunakan yaitu menggunakan 7 layer yaitu, *Convolutional Layer*, *Activation layer*, *Max Pooling Layer*, *Time Distributed*, *Bidirectional*, *Dropout*, *Dense*.

1. *Convolutional Layer*, Layer konvolusi 3D yang digunakan untuk mengekstraksi fitur dari data spasial dalam tiga dimensi. Parameter yang dipelajari selama pelatihan adalah bobot dari filter konvolusi.
2. *Activation Layer*, Layer aktivasi digunakan untuk memperkenalkan non-linearitas ke dalam model. Pada tahap ini, fungsi aktivasi (seperti ReLU) diterapkan pada keluaran dari layer sebelumnya.
3. *Max Polling Layer*, Layer max pooling 3D mengurangi dimensi spasial dari keluaran sebelumnya dengan cara memilih nilai maksimum dari setiap jendela.
4. *Time Distributed*, Layer ini mendistribusikan input ke seluruh dimensi dan menerapkan operasi yang sama ke setiap potongan data.
5. *Bidirectional*, Layer ini mengimplementasikan rekursi ke belakang di dalam jaringan. Ini berarti

bahwa informasi dapat mengalir maju dan mundur dalam model.

6. *Dropout*, Layer *dropout* diterapkan untuk membantu mencegah *overfitting* dengan secara acak mematikan sebagian node selama pelatihan.
7. *Dense*, Layer *dense* merupakan layer akhir dari model dan menghasilkan output dari model

| Model: "sequential"                |                          |         |
|------------------------------------|--------------------------|---------|
| Layer (type)                       | Output Shape             | Param # |
| conv3d (Conv3D)                    | (None, 75, 46, 140, 128) | 3584    |
| activation (Activation)            | (None, 75, 46, 140, 128) | 0       |
| max_pooling3d (MaxPooling3D)       | (None, 75, 23, 70, 128)  | 0       |
| conv3d_1 (Conv3D)                  | (None, 75, 23, 70, 256)  | 884992  |
| activation_1 (Activation)          | (None, 75, 23, 70, 256)  | 0       |
| max_pooling3d_1 (MaxPooling3D)     | (None, 75, 11, 35, 256)  | 0       |
| conv3d_2 (Conv3D)                  | (None, 75, 11, 35, 75)   | 518475  |
| activation_2 (Activation)          | (None, 75, 11, 35, 75)   | 0       |
| max_pooling3d_2 (MaxPooling3D)     | (None, 75, 5, 17, 75)    | 0       |
| time_distributed (TimeDistributed) | (None, 75, 6375)         | 0       |
| bidirectional (Bidirectional)      | (None, 75, 256)          | 6668096 |
| dropout (Dropout)                  | (None, 75, 256)          | 0       |
| bidirectional_1 (Bidirectional)    | (None, 75, 256)          | 394240  |
| dropout_1 (Dropout)                | (None, 75, 256)          | 0       |
| dense (Dense)                      | (None, 75, 41)           | 10537   |
| Total params: 8,471,924            |                          |         |
| Trainable params: 8,471,924        |                          |         |
| Non-trainable params: 0            |                          |         |

Gambar 3.6 Arsitektur Model (CNN)

Arsitektur CNN ini terdiri dari tiga lapisan konvolusi 3D yang digunakan untuk mengekstraksi fitur-fitur dari data spasial tiga dimensi. Setiap lapisan konvolusi diikuti oleh lapisan aktivasi yang memastikan bahwa outputnya selalu positif. Kemudian, lapisan *max pooling* mengurangi resolusi spasial dari fitur-fitur yang diekstrak. Selanjutnya, data melewati lapisan *time distributed*, yang mempertahankan urutan waktunya.

Dua lapisan rekursif *bidirectional* dengan *dropout* digunakan untuk memproses data secara maju dan mundur, memungkinkan model untuk memahami konteks sekitar dengan lebih baik. Akhirnya, terdapat lapisan *fully connected* yang menghubungkan setiap unit dari lapisan sebelumnya ke setiap unit pada lapisan ini. Total parameter pada arsitektur ini adalah sekitar 8.47 juta, dan semua parameter ini dapat diubah selama proses pelatihan untuk meminimalkan kesalahan prediksi.

### 3.3 Analisis Kebutuhan Sistem

Analisis yang berisi pengamatan dari masalah yang ada / pada sistem yang sedang berjalan. Dari hasil analisis pada pembuatan skripsi *Lip Reading* ini dapat dianalisis beberapa hal yang dibutuhkan, yaitu:

Tabel 3.1 Masalah dan Solusi

| Masalah | Solusi |
|---------|--------|
|---------|--------|

|                                                                                                                  |                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| Masih kurangnya pengetahuan dalam cara penulisan Bahasa Inggris yang diucapkan oleh seseorang.                   | <i>Lip Reading</i> dapat membantu dalam penulisan Bahasa Inggris yang dimana sebuah teks akan muncul ketika seseorang memilih video yang diinginkan. |
| Masih terdapat kesulitan seperti apa <i>grammar</i> , <i>vocabulary</i> dan <i>spelling</i> (ejaan) per-kalimat. | Mempermudah mengetahui penulisan <i>grammar</i> dan <i>vocabulary</i> dari pengucapan seseorang.                                                     |

### 3.4 Analisis Dokumen

Analisis dokumen pada *Lip Reading TalkClarity* dibuat untuk mengetahui dokumen apa saja yang melibatkan proses dari *lip reading*. Analisis dokumen ini melibatkan proses pengolahan visual dan pemahaman konteks percakapan dengan menggunakan gerakan dan bentuk bibir saat seseorang berbicara. Pada tabel 3.2 dijelaskan dokumen apa saja yang dibutuhkan.

Tabel 3.2 Tabel Analisis Dokumen Yang Diperlukan

| No | Dokumen                           | Uraian                                                                                                                                                                                                             |
|----|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Pengumpulan Data Video            | Deskripsi : Kumpulan dokumen atau rekaman video yang berisi adegan percakapan<br>Fungsi : Digunakan untuk menganalisis <i>lip reading</i>                                                                          |
| 2  | Pemrosesan Visual                 | Deskripsi : Suatu proses yang menggunakan algoritma pemrosesan gambar dan visi komputer.<br>Fungsi : Digunakan untuk ekstraksi wajah, segmentasi bibir, dan pemantauan Gerakan bibir.                              |
| 3  | Klasifikasi dan Pemahaman Konteks | Deskripsi : Pengklasifikasian dan memahami konteks percakapan dengan menghubungkan Gerakan bibir dengan kata kata atau frasa yang sesuai.<br>Fungsi : Digunakan untuk melakukan klasifikasi CNN pada video         |
| 4  | Evaluasi dan Validasi             | Deskripsi : Validasi hasil analisis dengan membandingkan hasil <i>lip reading</i> dengan transkripsi atau teks yang sebenarnya dari percakapan.<br>Fungsi : Digunakan untuk mendapatkan keakuratan pengenalan kata |

### 3.5 Analisis Data

Analisis data ini digunakan untuk mengetahui data apa saja yang akan menjadi data masukan dan data keluaran dalam *Lip Reading* ini. Terdapat data seperti berikut:

#### 3.5.1 Analisis Data Masukan

Suatu tahapan pada *Lip Reading* yang dapat digunakan untuk memasukkan . Data masukan yang dibutuhkan yaitu Data video yang akan diproses, diklasifikasi, dan juga di evaluasi sehingga akan menghasilkan sebuah teks.

#### 3.5.2 Analisis Data Keluaran

Suatu tahapan pada *Lip Reading* yang berfungsi untuk menampilkan hasil dari pemrosesan video berdasarkan Gerakan bibir. Sehingga pengguna dapat mengetahui bagaimana teks yang sesuai dengan yang ditampilkan pada video tersebut.

### 3.6 Analisis Fungsional

Analisis yang berisi sebuah proses-proses apa saja yang dilakukan oleh sistem ini. Perancangan yang dilakukan yaitu, pembangunan Sistem Website *Lip Reading* yang Bernama *TalkClarity*, dimana perancangan website ini merupakan suatu perancangan pendeteksian gerak bibir yang diproses menjadi sebuah teks, yang akan memudahkan pengguna dalam menulis teks Bahasa Inggris.

Di dalam pembangunan website ini diharuskan adanya fungsi – fungsi yang terdapat dalam website *Lip Reading* berdasarkan kebutuhan yang diperlukan, diantaranya :

1. Proses pilih video yang digunakan sebagai proses awal dalam deteksi gerak bibir.
2. Proses deteksi gerak bibir berdasarkan video yang dipilih.
3. Proses menampilkan hasil sebuah teks yang berasal dari video yang dipilih.

### 3.7 Analisis Penggunaan Sistem

Analisis pengguna digunakan untuk mengetahui siapa saja yang dapat menggunakan sistem website ini. Penggunaan sistem website ini tidak memerlukan akun untuk dapat memasukkan sebuah masukan. Sistem berbasis website ini dapat dibuka pada alat elektronik seperti pc maupun laptop.

Tabel 3.3 Tabel Analisis Penggunaan Sistem

| No | Pengguna            | Akses                                      |
|----|---------------------|--------------------------------------------|
| 1  | Link website        | Dapat membuka localhost <i>TalkClarity</i> |
| 2  | Pemilihan video     | Dapat memilih video yang sudah disediakan  |
| 3  | Deteksi gerak bibir | Dapat melihat hasil deteksi gerak bibir    |

## 4. PERANCANGAN SISTEM

Perancangan sebuah website *lip reading* ini menggunakan pendekatan berorientasi objek dengan mendefinisikan seluruh tipe objek-objek

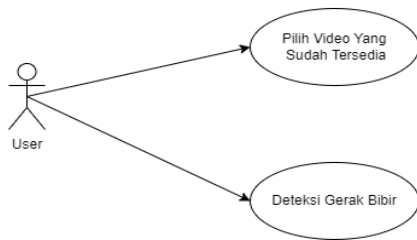
yang penting. Hasil yang diharapkan yaitu berupa sebuah website yang dapat mendeteksi gerakan bibir menjadi teks yang Bernama *TalkClarity*.

#### 4.1 Perancangan Sistem Prosedural

Perancangan sistem prosedural sistem website *lip reading* melibatkan langkah-langkah penting guna mengembangkan sistem yang efektif dalam mengenali dan menginterpretasi gerakan bibir menjadi teks yang dapat dipahami. Tahapan perancangan tersebut mencakup pemilihan teknologi, akuisisi data, *preprocessing*, ekstraksi fitur, pembuatan model, dan evaluasi.

##### 4.1.1 Use Case Diagram Yang Diusulkan

Menggambarkan hubungan interaksi antara aktor dan sistem. Berikut adalah *use case diagram Lip Reading (TalkClarity)*. Pada gambar 4.1 di bawah dijelaskan bahwa pada website *TalkClarity* ini pengguna akan memilih video yang diinginkan untuk proses *Lip to Text*, untuk pemilihan video telah disediakan *select button* untuk pengguna memilih video mana yang ingin dideteksi. Setelah pengguna memilih video maka akan dilakukan proses deteksi gerak bibir oleh sistem yang akan menghasilkan output satu kalimat teks yang sesuai dengan video tersebut.



Gambar 4.1 Use Case Diagram yang dirancang

##### 4.1.2 Skenario Use Case Diagram Yang Diusulkan

Skenario *Use Case* mendeskripsikan urutan langkah-langkah dalam sebuah website, baik yang dilakukan user terhadap sistem maupun yang dilakukan oleh sistem terhadap pengguna. Berikut adalah skenario masing-masing *use case* :

###### 1. Use Case Skenario Pilih Video

Nama Use Case : Pilih Video  
 Pengguna : User  
 Deskripsi : User memilih video yang disediakan

Tabel 4.1 Use Case Skenario Pilih Video

| Skenario Utama                                          |                                                        |
|---------------------------------------------------------|--------------------------------------------------------|
| Pengguna Utama                                          | Sistem                                                 |
| User memilih video yang diinginkan untuk proses deteksi |                                                        |
|                                                         | Menyimpan video yang dipilih user untuk proses deteksi |

###### 2. Use Case Skenario Deteksi Gerak Bibir

Nama Use Case : Deteksi Gerak Bibir  
 Pengguna : User  
 Deskripsi : User mendapat hasil deteksi gerak bibir

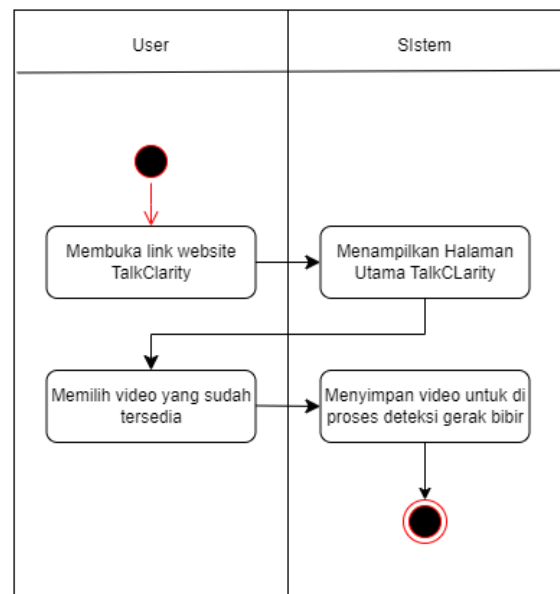
Tabel 4.2 Use Case Skenario Deteksi Gerak Bibir

| Skenario Utama                                                              |                                                                             |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Pengguna Utama                                                              | Sistem                                                                      |
| User melihat sebuah kalimat teks yang muncul dari hasil deteksi gerak bibir |                                                                             |
|                                                                             | Menampilkan teks dari proses konversi gerak bibir sesuai video yang dipilih |

#### 4.2 Activity Diagram Yang Diusulkan

Berikut ini adalah *activity diagram* untuk setiap *use case*:

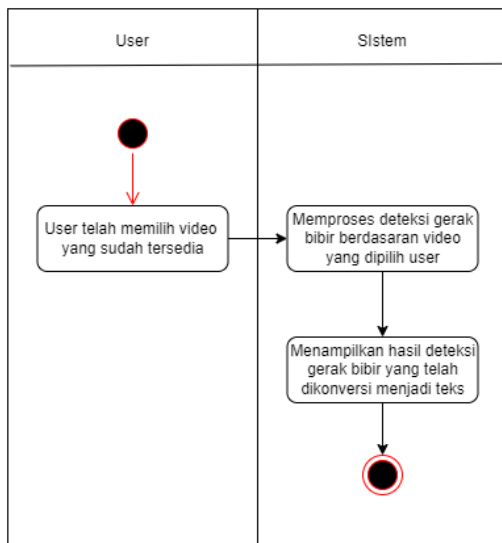
- Activity Diagram Pilih Video, dapat dilihat pada gambar 4.2.



Gambar 4.2 Activity Diagram Pilih Video

Pada gambar 4.2 dijelaskan pada activity diagram bagaimana proses user menggunakan website *lip reading*, berawal dari user membuka link website *TalkClarity*, dan sistem akan menampilkan halaman utama, setelah itu user dapat memilih video yang sudah tersedia untuk mengetahui hasil dari deteksi gerak bibir tersebut, lalu sistem akan menyimpan video yang telah dipilih dan melanjutkan ke proses deteksi gerak bibir.

2. Activity Diagram Deteksi Gerak Bibir dapat dilihat pada gambar 4.3.

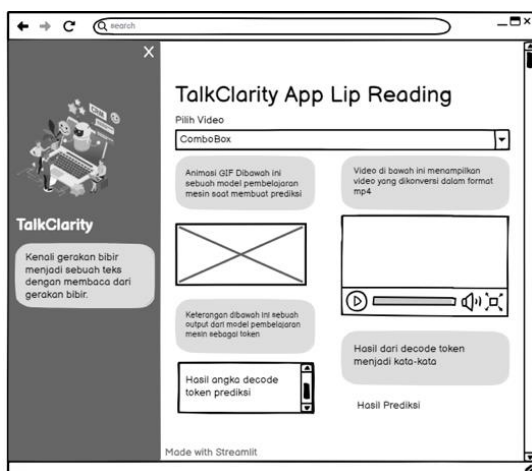


Gambar 4.3 Activity Diagram Deteksi Gerak Bibir

Pada gambar 4.3 dijelaskan pada activity diagram bagaimana proses deteksi gerak bibir, berawal dari user telah memilih video yang tersedia, lalu sistem yang sudah menyimpan video tersebut melanjutkan proses deteksi gerak bibir berdasarkan video yang dipilih user, setelah proses deteksi selesai, sistem akan menampilkan hasil deteksi gerak bibir yang telah dikonversi menjadi teks.

#### 4.3 Perancangan Antarmuka (User Interface)

Perancangan antar muka dibuat menggunakan balsamiq untuk mempermudah dalam merancang sistem website *Lip Reading* ini. Perancangan antar muka tersebut menampilkan tampilan utama yang meliputi *select button* pilih video dan hasil dari deteksi gerak bibir tersebut. Berikut adalah gambar dari perancangan antarmuka website *TalkClarity*.



Gambar 4.4 Mockup Sistem Website TalkClarity

## 5. IMPLEMENTASI DAN PENGUJIAN SISTEM

### 5.1 Implementasi Sistem

Tahap implementasi sistem merupakan proses menerjemahkan rancangan (desain) yang telah dibuat menjadi sistem yang dapat digunakan oleh pengguna. Proses implementasi sistem website *lip reading* ini meliputi pengumpulan dataset, pengolahan dataset, preprocess data, training model, testing, evaluasi, dan deployment. Sebelum memasuki tahapan implementasi sistem, dipersiapkan dulu perangkat keras dan perangkat lunak yang akan dipergunakan untuk implementasi dan pengujian.

#### 5.1.1 Kebutuhan Perangkat Lunak

Perangkat lunak digunakan mendukung kinerja sistem. Adapun perangkat lunak yang digunakan adalah sebagai berikut:

1. Sistem operasi minimal Windows 10
2. Visual Studio Code
3. Google Colab
4. Google Drive
5. OpenCv
6. Tensorflow versi 2.10.1
7. Tensorflow-gpu
8. Imageio versi 2.5.0
9. Jiwer
10. Matplotlib
11. Numpy
12. Streamlit
13. Balsamiq
14. Figma
15. Python versi 3.8.10
16. Github
17. Google Chrome, ataupun browser sejenis

#### 5.1.2 Kebutuhan Perangkat Keras

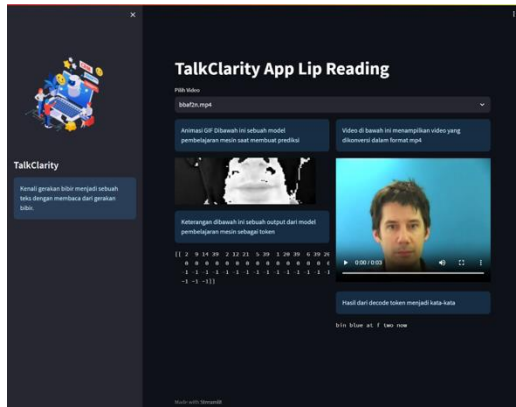
Perangkat keras diperlukan untuk media implementasi sistem website yang dirancang, dan harus sesuai dengan kebutuhan sistem. Adapun kebutuhan perangkat keras tersebut adalah:

1. Kebutuhan minimum:
  - a. Processor intel core i3/AMD Ryzen 3
  - b. RAM 4GB
  - c. Hardisk dengan kapasitas 500GB
  - d. Keyboard dan mouse
  - e. Mic dan Camera
  - f. Jaringan internet
2. Kebutuhan yang disarankan
  - a. Processor Intel core i5/AMD Ryzen 5
  - b. RAM 8GB
  - c. Hardisk dengan kapasitas 1TB

- d. Keyboard dan mouse
- e. Mic dan Camera
- f. Jaringan internet

## 5.2 Implementasi Penggunaan

Pada bagian ini dijelaskan bagaimana cara menggunakan website *Lip Reading*. Di bawah ini terdapat *interface* yang dapat digunakan oleh user.



```
Epoch 1/10
450/450 [=====] - 655s 1s/step - loss: 0.2326 - val_loss: 0.2198
Epoch 2/10
450/450 [=====] - 635s 1s/step - loss: 0.2905 - val_loss: 0.4693
Epoch 3/10
450/450 [=====] - 637s 1s/step - loss: 0.3089 - val_loss: 0.3429
Epoch 4/10
450/450 [=====] - 632s 1s/step - loss: 0.3987 - val_loss: 0.1261
Epoch 5/10
450/450 [=====] - 632s 1s/step - loss: 0.3447 - val_loss: 0.0159
Epoch 6/10
450/450 [=====] - 630s 1s/step - loss: 0.2293 - val_loss: 0.0114
Epoch 7/10
450/450 [=====] - 630s 1s/step - loss: 0.5058 - val_loss: 0.0251
Epoch 8/10
450/450 [=====] - 632s 1s/step - loss: 0.3163 - val_loss: 0.0361
Epoch 9/10
450/450 [=====] - 634s 1s/step - loss: 0.2820 - val_loss: 0.0140
Epoch 10/10
450/450 [=====] - 631s 1s/step - loss: 0.2347 - val_loss: 0.0146
```

Gambar 5.9 Train Model (6)

```
print("Akurasi multikelas: {:.2f}%".format(accuracy * 100))
print("Word Error Rate (WER):", wer)

Akurasi multikelas: 100.00%
Word Error Rate (WER): 0.0
```

Gambar 5.10 Hasil Akurasi (3)

```
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)

Precision: 1.0
Recall: 1.0
F1-score: 1.0
```

Gambar 5.11 Hasil Precision, Recall, F1-score

Dan pada akhir *train model* yang telah mendapatkan *best fit*, ditambahkan *WER (Word Error Rate)* untuk menambahkan seberapa besar kata yang tidak sesuai dengan deteksi gerak bibir tersebut, dan juga nilai hasil dari Precision, Recall, dan F1-score sebagai sejauh mana model klasifikasi kita baik dalam memprediksi kelas yang benar dan menghindari kesalahan.

#### 5.4 Pengujian Video (Test Video)

Tahap ini melakukan pengujian video berdasarkan best model yang didapat pada saat *train model*. Berikut adalah langkah-langkah proses *testing video*:

1. Memuat model, Memuat berat (*weights*) model yang telah disimpan sebelumnya menggunakan `model.load_weights`.
2. Mendapatkan data uji, Mengambil data uji dari iterator yang telah dihasilkan oleh `test.as_numpy_iterator()`.

```
test_data = test.as_numpy_iterator()
```

Gambar 5.12 Mengambil Data Uji

3. Memprediksi dengan model
  - a. Mengambil batch data uji dengan menggunakan `test_data.next()`.
  - b. Menggunakan model untuk memprediksi hasil dengan memanggil `model.predict` pada input batch.

```
[ ] sample = test_data.next()

[ ] yhat = model.predict(sample[0])

1/1 [=====] - 0s 275ms/step

print("-" * 100, "REAL TEXT")
[tf.strings.reduce_join(num_to_char(word) for word in sentence) for sentence in sample[1]]
[tf.Tensor: shape=(), dtype=string, numpy=b'place red in c three soon',
  tf.Tensor: shape=(), dtype=string, numpy=b'bin green at h three again')]

[ ] decoded = tf.keras.backend.ctc_decode(yhat, input_length=[7,7,5], greedy=True)[0][0].numpy()

[ ] print("-" * 100, "PREDICTIONS")
[tf.strings.reduce_join(num_to_char(word) for word in sentence) for sentence in decoded]
[tf.Tensor: shape=(), dtype=string, numpy=b'place red in c three soon',
  tf.Tensor: shape=(), dtype=string, numpy=b'bin green at h three again')]

PREDICTIONS
```

Gambar 5.13 Membuat Prediksi

4. Menampilkan *Real Text*
  - a. Menggunakan `loop` untuk menggabungkan karakter dari *alignment* yang terdapat dalam `sample[1]`.
  - b. Menggunakan `num_to_char` untuk mengubah angka *alignment* menjadi karakter.
  - c. Hasil *alignment* asli dari video ditampilkan.
5. Melakukan Prediksi dengan model
  - a. Menggunakan model yang telah dilatih sebelumnya untuk memprediksi teks dari video.
  - b. Memasukkan *frames* video sebagai input ke model menggunakan `tf.expand_dims` untuk menambah dimensi batch.
  - c. Menggunakan model *predict* untuk mendapatkan hasil prediksi dalam bentuk probabilitas karakter.
6. *Decoding* dan Menampilkan Prediksi
  - a. Menggunakan `ctc_decode` dari `tf.keras.backend` untuk mendekode hasil prediksi menjadi teks yang lebih terbaca.
  - b. Hasil *decoding* disimpan dalam variabel *decoded* sebagai array dari angka-angka yang mewakili karakter.
  - c. *Loop* digunakan untuk menggabungkan karakter dari *decoded* menggunakan `num_to_char`, dan hasil prediksi dari video ditampilkan.

```
Test on a Video

[ ] sample = load_data(tf.convert_to_tensor('content/drive/MyDrive/data/c1/p1/k1a-mpg'))

[ ] print("-" * 100, "REAL TEXT")
[tf.strings.reduce_join(num_to_char(word) for word in sentence) for sentence in sample[1]]
[tf.Tensor: shape=(), dtype=string, numpy=b'place red in c three soon',
  tf.Tensor: shape=(), dtype=string, numpy=b'bin green at h three again')]

[ ] yhat = model.predict(tf.expand_dims(sample[0], axis=0))

1/1 [=====] - 1s 383ms/step

[ ] decoded = tf.keras.backend.ctc_decode(yhat, input_length=[7,7,5], greedy=True)[0][0].numpy()

[ ] print("-" * 100, "PREDICTIONS")
[tf.strings.reduce_join(num_to_char(word) for word in sentence) for sentence in decoded]
[tf.Tensor: shape=(), dtype=string, numpy=b'place green in k one again']

PREDICTIONS
```

Gambar 5.14 Test On Video

#### 5.5 Jenis Pengujian Sistem

Pada tahapan ini, dilakukan pengujian sistem dengan menggunakan metode pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Tahapan pengujian ini diperlukan agar program yang dihasilkan telah sesuai dengan tujuan yang ingin dicapai dan untuk menghindari kesalahan-kesalahan dalam program.

### 5.6 Rencana Pengujian Sistem

Pengujian sistem berikut menggunakan data uji berupa pengolahan data, pengolahan proses. Berikut tabel rencana pengujian sistem.

Tabel 5.1 Tabel Rencana Pengujian Sistem

| Kelas Uji     |       | Butir Uji              |
|---------------|-------|------------------------|
| Deteksi Bibir | Gerak | a. Menguji gerak bibir |
|               | Bibir | b. Menguji teks        |

### 5.7 Kasus dan Hasil Pengujian Sistem

Dalam melakukan pengujian, terdapat objek yang diharuskan untuk diuji saat itu untuk mendapatkan hasil dari pengujian tersebut.

- Deteksi Gerak Bibir, Pengujian Lip Reading (Deteksi Gerak Bibir) dilakukan sebagai proses konversi gerak bibir menjadi teks. Berikut pengujian yang ditampilkan dalam bentuk tabel:

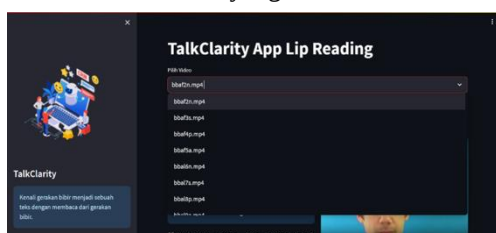
Tabel 5.2 Pengujian Deteksi Gerak Bibir

| Kasus dan Hasil Uji (Data Normal) |                                                                                 |                                                                                                    |            |
|-----------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|------------|
| Data                              | Yang Diharapkan                                                                 | Hasil Uji                                                                                          | Kesimpulan |
| Menguji Gerak Bibir               | Sistem dapat memproses deteksi yang akan menghasilkan sebuah decode token       | Sistem berhasil mendeteksi sehingga hasil teks yang sesuai dengan gerak bibir muncul               | OK         |
| Kasus dan Hasil Uji (Data Salah)  |                                                                                 |                                                                                                    |            |
| Sistem                            | Yang Diharapkan                                                                 | Hasil Uji                                                                                          | Kesimpulan |
| Menguji Gerak Bibir               | Sistem tidak dapat memproses deteksi yang akan menghasilkan sebuah decode token | Sistem tidak berhasil mendeteksi sehingga hasil teks yang muncul tidak sesuai dengan gerakan bibir | OK         |

### 5.8 Hasil Kesimpulan Pengujian Sistem

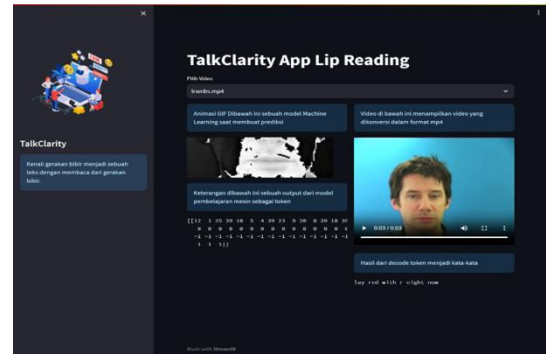
Setelah melakukan pengujian sistem, terdapat hasil kesimpulan dari pengujian sistem *Lip Reading* ini. Berikut adalah gambaran hasil dari sistem :

- Memilih Video, Sistem ini memberikan pilihan video dengan menggunakan *select button* yang dimana isi dari video yang ada di dataset.



Gambar 5.15 Interface Memilih Video

### 2. Hasil Deteksi Gerak Bibir.

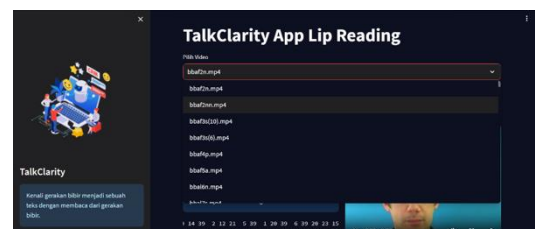


Gambar 5.16 Interface Hasil Deteksi Gerak Bibir

Lalu ketika sudah memilih video, sistem akan menampilkan animasi GIF dari sebuah model *Machine Learning* saat membuat prediksi, dan menampilkan sebuah output sebagai token. Menampilkan video yang sesuai dengan yang dipilih, lalu dibawah preview video akan menampilkan hasil yang dikonversi menjadi teks.

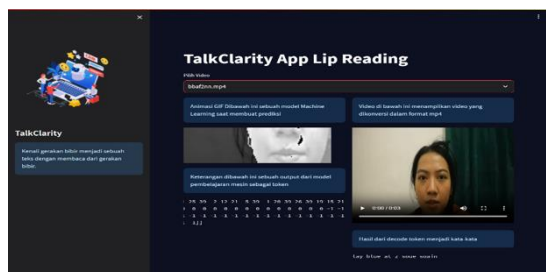
Hasil pengujian diatas merupakan hasil dari pengujian video yang ada di dataset, terdapat juga hasil pengujian yang dilakukan dengan menggunakan video lain dan dengan catatan memiliki properties video atau format video yang sesuai dengan video dataset.

- Memilih Video, Sistem ini memberikan pilihan video dengan menggunakan *select button* yang dimana isi dari video yang ada di dataset.



Gambar 5.17 Interface Memilih Video (2)

### 2. Hasil Deteksi Gerak Bibir



Gambar 5.18 Interface Hasil Deteksi Gerak Bibir

Dari hasil tersebut disimpulkan, model dapat mendeteksi Gerakan bibir di video yang lain sesuai dengan catatan yang diberikan sebelumnya, tetapi hanya 2 kata saja yang terdeteksi yaitu "*blue*" dan "*at*", dikarenakan masih terdapat perbedaan posisi mulut atau wajah dan juga perbedaan dalam pengucapannya.

## 6. PENUTUP

### 6.1 Kesimpulan

Berdasarkan dari yang telah dibuat pada website TalkClarity (lip reading) ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Dibuatnya sebuah website *lip reading* yang dapat menghasilkan teks dari pendeteksian Gerakan bibir. Dan video yang digunakan adalah video yang terdapat di dataset.
2. Terkonfigurasinya sebuah komponen dataset Kumpulan video dan *alignment* dengan pendeployan website menggunakan streamlit.
3. Dengan pengujian *training* model sebanyak 16 kali, sehingga terciptanya sebuah sistem *lip reading* yang memiliki nilai akurasi 100% dan nilai WER (*Word Error Rate*) 0.
4. Berhasil mendapatkan hasil deteksi gerakan bibir kurang lebih sebesar 30% atau sama dengan terdeteksinya 2 kata dari contoh yang digunakan dengan menggunakan video lain yang disesuaikan dengan format video dataset.

### 6.2 Saran

Pada sistem *lip reading* ini tentu masih banyak yang harus penulis perbaiki pada kekurangannya. Saran untuk pengembangan sistem *lip reading* di masa yang akan datang adalah sebagai berikut :

1. Menambah pemrosesan model agar dapat menggunakan video lain, sehingga tidak hanya menggunakan dataset.
2. Menyajikan tutorial dan panduan langkah demi langkah penggunaan website *lip reading*.
3. Penambahan fitur upload video yang dapat digunakan oleh video lain selain video dari dataset.
4. Penambahan fitur *history*, seperti *history* deteksi bibir sebelumnya.

### Daftar Pustaka

- [1] D. S. Society, "Medium," 28 Februari 2022. [Online]. Available: [https://medium.com/@insight\\_imi/lip-reader-using-cnn-lstm-b3fe3a434914](https://medium.com/@insight_imi/lip-reader-using-cnn-lstm-b3fe3a434914).
- [2] H. M. N. S. M. S. C. S. R. Soraj V, "Lip-Reading Techniques: A Review," *Jurnal Internasional Scientific & Technology Research*, vol. 9, no. 02, p. 6, 2020.
- [3] r. E. R. S Yustiatwat, "Sistem Pengaman Pada Pintu Menggunakan Face Detection Dengan Metode Haar Cascade Classifier," *Junral Elektro*, pp. 118-122, 2018.
- [4] Y. Y. e. al, "Style Aggregated Network for Facial Landmark Detection," 2018.
- [5] M. H. Andreas Kaplan, "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15-25, 2019.
- [6] D. D. Budiarjo, "Implementasi Sistem Cerdas Pada Otomatisasi Pendeteksian Jenis Kendaraan Di Jalan Raya," vol. 2507, 2020.
- [7] G. A. Royani Darma Nurfita, "Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari," *Jurnal Teknik Elektro*, vol. 18, 2018.
- [8] R. A. Husnibes Muchtar, "Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah dengan Metode Template Matching Menggunakan Open Source Computer Vision Library (Opencv)," *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, vol. 1, 2019.
- [9] Klaus Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, 3rd Edition, Giesecke & Devrient GmbH, Munich, Germany: West Sussex, 2010.
- [10] Rahayu, M. I., & Nasihin, A. (2020). PERANCANGAN DETEKSI PENGENALAN WAJAH MENGGUNAKAN METODE LOCAL BINARY PATTERN (LBP) . *Jurnal Teknologi Informasi Dan Komunikasi*, 9(1), 48–54. <https://doi.org/10.58761/jurtikstmikbandung.v9i1.145>