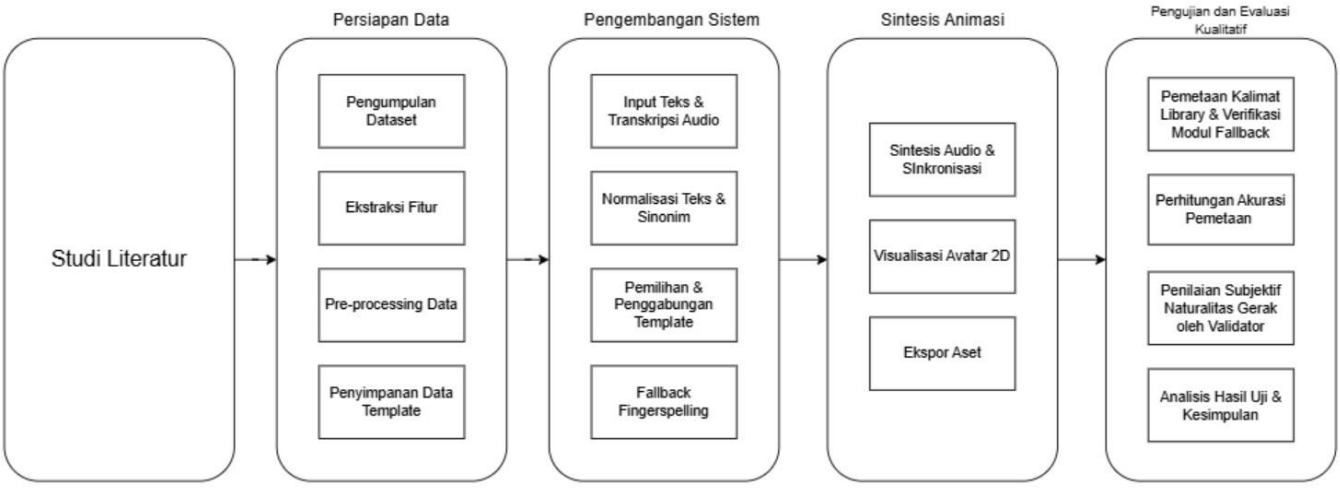
BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Penelitian yang dilakukan dengan beberapa tahapan, dimulai dari studi literatur mengenai penelitian audio/text-to-sign dan teknologi yang mendukung, lalu ke perancangan modul sistem yang mencakup pengolahan data teks dan gestur serta integrasi pipeline. Setelah itu, melalukan implementasi visualisasi gestur dalam bentuk avatar. Tahapan akhir adalah pengujian dan evaluasi, dilanjutkan dengan penyusunan laporan. Pada Gambar 3.1 diberikan alur metode penelitian yang akan dilakukan oleh peneliti.



Gambar 3. 1 Alur Penelitian

3.2 Studi Literatur

Penulis memulai dengan melakukan riset mengenai topik yang berhubungan dengan penerjemahan teks ke bahasa isyarat, ekstraksi pose tubuh menggunakan *mediapipe*, visualisasi gestur dengan skeleton, serta teknik pemrosesan teks untuk pemetaan gestur. Melalui penelitian terdahulu, ditemukan berbagai dataset gestur seperti dataset BISINDO yang tersedia di platform Kaggle, serta metode pre-processing pose dan teks yang efektif untuk meingkatkan akurasi pemetaan. Pemilihan dataset dan metode yang tepat menjadi langkah penting karena akan menjadi basis pengembangan sistem audio/text-to-gesture yang *robust*.

3.3 Dataset dan Pra-pemrosesan

Pada tahap ini mencakup persiapan template gestur yang akan digunakan sebagai basis data gerakan. Proses dilakukang untuk mengubah koleksi rekaman video gestur BISINDO menjadi format keypoint yang terstruktur dan di-normalisasi.

3.3.1. Dataset Template Gestur

Dataset yang digunakan adalah koleksi template gerakan BISINDO yang telah di-ekstraksi dari rekaman video menggunakan kerangka kerja *mediapipe*. Dataset primer terdiri dari 91 kosakata BISINDO di mana setiap kosakata memiliki beberapa varian *frame sequence* untuk memberikan keragaman gerakan. Setiap template disimpan sebagai urutan data numerik 3D (bukan video). Contoh output per frame adalah struktur dictionary yang berisi kooridnat X, Y dan Z untuk setiap keypoint. Jumlah 91 kosakata ini menjadi basis kosakata dasar yang kemudian diperluas fungsionalitasnya melalui modul NLP dan *fingerspelling*. Setiap template gestur disimpan sebagai urutan keypoint dalam format Python Pickle (.pkl) yang akan di-kelompokan berdasarkan kelas gestur di yang mendukung pemanggilan template secara cepat saat *runtime*.

Peneliti menggunakan data yang diperoleh dari *Kaggle* dengan sumber yang berbeda-beda. Jumlah dataset per kosakata juga memiliki jumlah yang beragam. Berikut rincian detail pada dataset kosakata BISINDO.

Tabel 3. 1 Kosakata BISINDO yang digunakan

Kosakata								
A	Berangkat	Dimana	Hijau	Kamu	Mandi	N	Sabar	Teman
Air	Berdiri	Duduk	Hitam	Kapan	Marah	Nama	Saya	Terima Kasih
Anak	Bingung	E	Hobi	Keluarga	Melihat	O	Sedih	Tidur
Apa	C	F	I	Kita	Membaca	Olahraga	Sekian	Tuli
Asal	Cari	G	Ibu	Kuning	Mengapa	P	Selamat	U
Ayah	D	Guru	Ingat	L	Menulis	Pagi	Senang	V

B	Dan	H	J	Lagi	Merah	Q	Siang	W
Bagaimana	Datang	Halo	K	M	Mereka	R	Siapa	X
Baik	Dengar	Hari	Kalian	Maaf	Minum	Ramah	Sore	Y
Belajar	Dia	Hari ini	Kami	Malam	Motor	S	T	Z

Pada penelitian ini, jumlah 91 kosakata ditetapkan sebagai batas library dikarenakan keterbatasan sumber data BISINDO yang tersedia secara publik di internet. BISINDO adalah bahasa alami yang belum memiliki repositori atau basis data gestur skala besar yang ter-standarisasi dan dapat diakses untuk pelatihan model AI. Data diperoleh dari Kaggle dengan sumber yang berbeda-beda dan jumlah dataset per kosakata juga memiliki jumlah frame yang beragam. Oleh karena itu, 91 kosakata ini mewakili jumlah maksimal data yang berhasil diakuisisi dan validasi untuk memastikan kualitas dan konsistensi data.

3.3.2. Perancangan Ekstraksi Keypoint

Ekstraksi keypoint merupakan proses fundamental untuk megubah data visual yang kontinu menjadi data numerik diskrit 3D. Proses ini mengubah data visual menjadi data numerik 3D dan menyimpannya dalam struktur dictionary di setiap frame dengan menggunakan kerangka kerja keypoint detection *mediapipe pose* dan *mediapipe hands* untuk akses yang efisien. Hal ini penting karena video mentah tidak dapat di-blending, Hanya data koordinat numerik 3D yang dapat dihitung untuk menciptakan gerakan transisi yang smooth, sehingga memastikan sintesis gerakan yang koheren. Proses ini dibutuhkan karena video mentah tidak dapat langsung diproses atau digabungkan secara komputasi untuk menciptakan gerakan baru. Keypoint dibutuhkan karena:

1. Koordinat numerik dapat dinormalisasi ke rentang [0, 1], membuat gestur tidak sensitive terhadap posisi subjek di kamera.
2. Hanya data koordinat numerik 3D yang dapat dihitung, smoothing dan blending antar gestur untuk menciptakan gerakan-transisi yang natural karena video mentah akan menghasilkan transisi yang kaku.

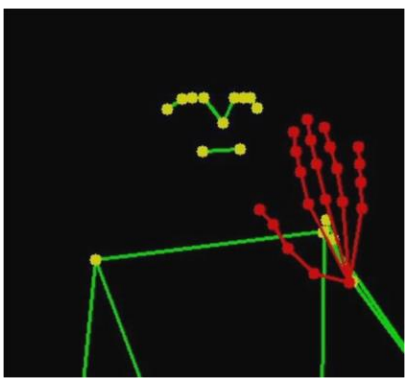
Output dari proses ekstraksi ini adalah template yang mengisi library pada tabel 3.1. Setiap baris dalam tabel tersebut diwakili oleh satu atau lebih file .pkl yang berisi urutan keypoint hasil ekstraksi. Struktur keypoint per frame dibagi menjadi 4 segmen berdasarkan mediapipe:

1. Pose: terdiri dari 33 keypoint yang mencakup seluruh kerangka tubuh (torso, kepala, anggota gerak).
2. Left_hand: terdiri dari 21 keypoint yang secara detil me-representasikan artikulasi jari tangan kiri.
3. Right_hand: terdiri dari 21 keypoint yang secara detil me-representasikan artikulasi jari tangan kanan.
4. Face: terdiri dari 468 keypoint fitur wajah. Meskipun diekstrak, data ini dapat di-sederhanakan atau diabaikan.

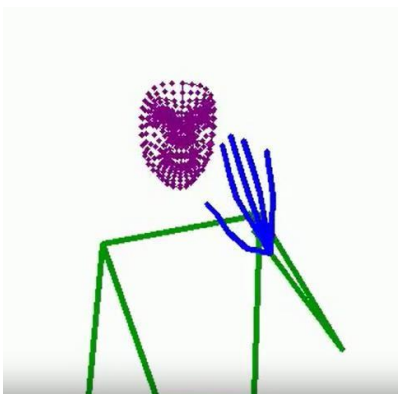
Seluruh koordinat di semua segmen telah di-normalisasikan untuk mencapai konsistensi spasial. Koordinat X dan Y nilainya di-normalisasi ke rentang [0, 1], mencerminkan posisi relatif keypoint pada bidang 2D kamera. [0, 0] adalah sudut kiri atas dan [1, 1] adalah sudut kanan bawah. Koordinat Z nilainya merupakan estimasi kedalaman 3D relatif terhadap keypoint di tengah panggul. Data Z dipertahankan dalam keypoint untuk konsistensi dataset namun tidak digunakan dalam proses rendering 2D akhir.

Untuk menjamin kualitas dan konsistensi data dalam *library*, penulis melakukan kurasi manual terhadap sumber video mentah. Dari puluhan video per kosakata, hanya 1 hingga 2 varian dengan kualitas visual yang baik, gerakan yang paling representative dan minim noise yang dipilih sebagai *template* akhir yang akan di-ekstraksi *keypoint*nya.

(a)



(b)



(c)

Gambar 3. 2 Visualisasi proses ekstraksi keypoint dari gestur BISINDO: (a) frame video mentah sebagain input sistem, (b) hasil deteksi keypoint pose tubuh dan tangan menggunakan MediaPipe dalam bentuk skeleton 2D dan (c) contoh representasi keypoint wajah yang disimpan sebagai struktur data

3.3.3. Perancangan Normalisasi Skala Global

Setelah normalisasi per grame, dilakukan normalisasi global pada seluruh dataset. Langkah ini bertujuan untuk menciptakan dimensi fisik yang seragam bagi semua *template* gestur guna mengatasi variasi jarak subjek dari kamera. Fungsi ini menghitung batas ekstrem dari seluruh koordinat X dan Y yang valid (antara 0 dan 1). Rentang vertical (Ymax – Ymin) dari data normalisasi ini menjadi kunci untuk mendefinisikan *scale_factor* untuk proyeksi keypoint ke dalam *viewport* 2D.

3.4 Perancangan Modul Sintesis Gerakan

Modul sintesis gerakan adalah komponen runtime utama sistem yang bertugas untuk memproses teks input dan menggabungkan menjadi satu urutan 3D keypoint yang koheren. Logika ini di-implementasikan melalui modul NLP sehderhana, mekanisme *template concatenation* dan algoritma smoothing.

3.4.1 Modul NLP sederhana dan *Synonym Matching*

Karena keterbatasan data BISINDO (91 kosakata), Modul NLP dirancang untuk meningkatkan jangkauan kosakata secara efektif melalui *rule-based matching*. Setiap teks input pertama melalui konversi seluruh karakter menjadi huruf kecil dan pembersihan dari tanda baca atau symbol non-alfabetik. Normalisasi memastikan bahwa pencarian template di direktori tidak sensitif terhadap kasus. Sistem juga membuat kamus sinonim untuk mengatasi masalah *Out of Vocabulary* (OOV). Kata yang tidak ditemukan langsung di template akan dicari yang tersedia di dataset. Strategi ini secara signigikan meningkatkan *Vocabulary Coverage* sistem tanpa memerlukan pelatihan model NLP yang kompleks.

3.4.2 Perancangan *Template Concatenation* dan *Fallback*

Setelah tiap kata berhasil dipetakan, sistem akan menyusun keypoint menjadi urutan gerakan yang berkelanjutan. Untuk kata yang berhasil di-cocokkan baik secara langsung maupun melalui sinonim, sistem mengambil *template* .pkl yang sesuai. Jika suatu kata memiliki beberapa

varian template (misalnya, saya_01.pkl, saya_02.pkl), sistem akan memilih satu template acak. Pemilihan acak ini bertujuan untuk memberikan variasi alami pada gerakan yang disintesis, mencegah animasi terlihat monoton dan berulang.

Untuk *fingerspelling fallback*, merupakan mekanisme robustness sistem terhadap kata yang tidak bisa dicakup (OOV) bahkan setelah *synonym matching*. Jika sebuah kata dinyatakan OOV absolut, sistem akan secara otomatis beralih ke *fingerspelling*. Kata tersebut akan dipecah menjadi karakter individual dan sistem mencari template gestur per karakter (A, B, C, dsb) untuk me-sintesis gerakan. Mekanisme fallback ini fitur krusial untuk memastikan kalimat input pengguna bisa di-terjemahkan secara visual, baik melalui gestur kata atau *fingerspelling*.

3.4.3 Algoritma *Smoothing* dan *Blending*

Penggabungan urutan keypoint secara langsung akan menghasilkan gerakan yang kaku. Oleh karena itu, diterapkan algoritma *smoothing* melalui Interpolasi Linier. Antar kata (gestur murni) disediakan 8 *transition frames* untuk blending antar 2 gestur kata yang berbeda (A→B). Durasi yang lebih lama ini memberikan transisi yang lebih halus dan natural untuk gerakan tubuh yang besar. Antar karakter (*fingerspelling*) disediakan 3 *transition frames*. Jumlah *frame* yang lebih sedikit ini diterapkan karena gerakan jari harus lebih cepat dan gesit untuk menjaga kecepatan komunikasi *fingerspelling*.

Untuk setiap keypoint K di frame transisi t, koordinat baru dihitung berdasarkan interpolasi antara frame akhir gestur A dan frame awal getur B. Proporsi α menentukan seberapa dekat keypoint berada ke gestur B. Dimana $\alpha = t/(N+1)$ dan N adalah total *transition frames*. Metode ini menghasilkan blending yang mulus, menghilangkan dikontinuitas yang disebabkan oleh perbedaan posisi keypoint antara akhir dan awal dua gestur yang berurutan.

3.5 Perancangan Modul Visualisasi dan Output

Modul ini adalah tahap akhir dalam pipeline untuk mem-visualisasikan urutan keypoint gabungan yang telah di-sintesis dan *smoothing* menjadi format output yang bisa digunakan oleh pengguna (video 2D).

3.5.1 Visualisasi Skeleton 2D dan Sinkronisasi Output

Visualisasi Skeleton 2D berfungsi sebagai validasi cepat dan output yang dapat diakses pengguna. Tujuannya untuk memproyeksikan koordinat 3D Keypoint yang telat di-normalisasi [0, 1] ke dalam ruang 2D pixel (1080 x 720) untuk rendering video. Penskalaan dilakukan dengan menggunakan rentang *global min/max*. Koordinat keypoint di mapping secara linear ke resolusi viewport akhir untuk memastikan skeleton terlihat proporsional dan berpusat di tengah layar, terlepas dari ukuran aslinya di dataset

Untuk memastikan tangan terhubung secara anatomis ke pergelangan tangan model pose, *keypoint* 0 (pangkal pergelangan) dari model tangan akan disetel ulang posisinya secara dinamis di setiap *frame*. Posisi baru *keypoint* 0 tangan di-letakkan persis diatas *keypoint* 15 (pergelangan tangan kiri) dan *keypoint* 16 (pergelangan tangan kanan) model pose. Prosedur *alignment* ini mencegah dislokasi visual yang bisa terjadi saat menggabungkan data pose dan hand dari *mediapipe*.

Caption teks akan ditampilkan di video skeleton 2D yang dimana kata yang sedang disintesis ditandai dengan kurung siku, memberikan feedback sinkronisasi visual, audio dan gestur secara *real-time.*

3.6 Rencana Pengujian Sistem

Rencana pengujian sistem ini dirancang untuk mem-validasi pemenuhan dua persyaratan utama penelitian yang terkait dengan latar belakang dan rumusan masalah kinerja waktu.

3.6.1 Pengujian Fungsionalitas

Pengujian ini bertujuan untuk mengukur kemampuan sistem dalam menerjemahkan teks masukan ke dalam urutan gestur BISINDO secara semantik dan visual.

1. Metode Pengujian:
    - Menggunakan 10 kalimat bahasa Indonesia yang dirancang dengan kombinasi gestur dari *library* penulis.
    - Menggunakan kata yang tidak ada dalam *library* penulis untuk memverifikasi bahwa modul fallback bekerja dan menghasilkan

fingerspelling per karakter yang akurat.

2. Parameter Metrik:
    - Persentase kata yang berhasil dipetakan ke gestur yang benar (*termasuk synonym matching*)
    - Penilaian subjektif oleh validator mengenai seberapa natural dan benar gestur yang dihasilkan secara visual.

3.6.2 Pengujian Waktu Produksi

Pengujian ini bertujuan untuk membuktikan efisiensi sistem dibandingkan dengan metode produksi manual.

1. Metode Pengujian:
    - Sistem akan diuji menggunakan empat skenario kalimat dengan panjang yang bervariasi.
    - Setiap skenario dijalankan sebanyak 10 kali repeteisi untuk mendapatkan nilai rata-rata yang stabil dan menghitung standar deviasi.
2. Parameter Metrik:
    - Durasi komputasi algoritma untuk penggabungan template dan proses smoothing
    - Durasi keseluruhan dari input hingga file video selesai di encoding.
    - Perbandingan antara durasi video yang dihasilkan dengan waktu produksi total.
    - Perbandingan waktu total sistem terhadap standar industri produksi manual.

3.6.3 Pengujian Keterpahaman Linguistik

Pengujian ini dilakukan untuk memastikan bahwa gestur yang dihasilkan tidak hanya mulus secara visual tetapi juga membawa makna yang benar secara linguistik.

1. Metode Pengujian:
    - Ahli diminta melihat video hasil sistem dan menerjemahkan Kembali ke dalam teks bahasa Indonesia tanpa mengtahui teks input aslinya.
2. Parameter Metrik:
    - Persentasi kemiripan antara teks masukan asli dengan hasil translasi balik ahli.