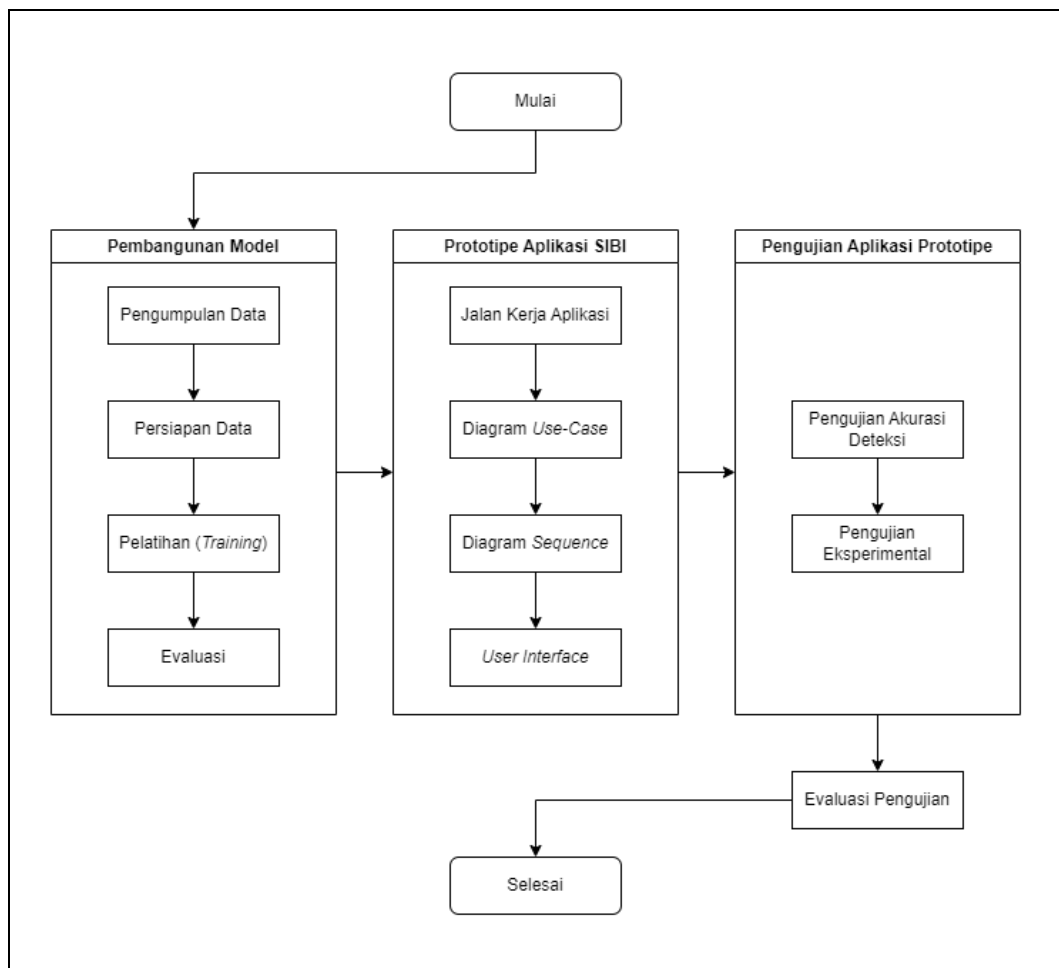


BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metode penelitian ini menjelaskan bagaimana tahapan penelitian yang akan dilakukan. Di bawah ini merupakan tahapan penelitian yang disajikan pada Gambar 3.1.



Gambar 3.1 Alur penelitian

Metode ini secara umumnya dibagi menjadi empat tahap penelitian, yaitu pembangunan atau pelatihan model, prototipe aplikasi SIBI, pengujian aplikasi prototipe dan evaluasi pengujian. Tahap pembangunan model terdiri dari proses pengumpulan data, persiapan data, pelatihan (*training*) dan evaluasi dari pelatihan tersebut. Tahap prototipe aplikasi SIBI terdiri dari proses menjelaskan jalan kerja, diagram *use-case* dan *sequence* serta *user interface* dari aplikasi yang dibuat. Tahap pengujian aplikasi dapat dibagi menjadi dua proses, yaitu pengujian akurasi deteksi dan pengujian eksperimental yang dimana deteksi gestur dilakukan di berbagai posisi. Hasil yang didapatkan dari pengujian aplikasi akan diakhiri dengan tahap terakhir yaitu evaluasi model.

3.2 Pelatihan Model

3.2.1 Pengumpulan Data

Pengumpulan data adalah tahap mengumpulkan informasi dari satu atau pelbagai sumber yang dianggap penting untuk pelatihan data. Hal ini bertujuan untuk mencapai suatu tujuan fungsi dari sebuah proyek *machine learning*, yakni dalam kasus ini adalah memiliki kemampuan mengklasifikasi bermacam-macam gestur SIBI. Tahap pengumpulan data dalam penelitian ini dilakukan paling awal sebelum berlanjut ke tahapan lainnya.

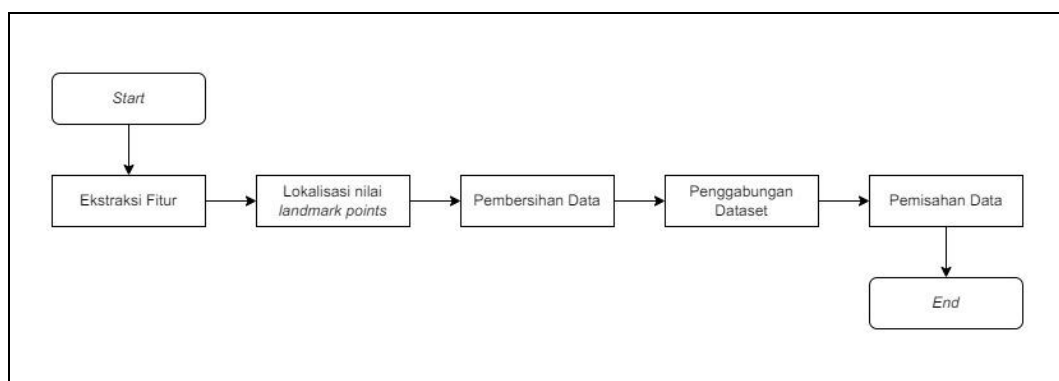
Pengumpulan ini akan dilakukan dengan pengambilan dataset citra SIBI yang tersedia dari situs *Kaggle*. Dataset yang tersedia dalam situs *Kaggle* dipilih untuk penelitian ini karena telah memiliki verifikasi dari Lembaga Penelitian dan Pengembangan (LEMLITBANG) dan merupakan satu-satunya yang tersedia dan terverifikasi dalam situs *Kaggle* (semasa penelitian ini sedang dilakukan)

sehingga dipastikan bahwa gestur SIBI yang terdapat dalam dataset ini adalah gestur yang benar.

3.2.2 Persiapan Data

Persiapan data adalah sebuah proses mengolah data mentah yang diterima dari tahap pengumpulan data menjadi format yang dapat dibaca dan dilatih oleh mesin. Tahap ini dianggap penting untuk memaksimalkan performa pelatihan dan menghindari beberapa kesalahan yang dapat mempengaruhi kinerja atau hasil dari pelatihan dengan buruk. Hasil dari pelatihan yang buruk akan menyebabkan performa deteksi gestur SIBI yang rendah.

Persiapan data dalam penelitian ini secara garis besar bertujuan untuk mengambil data *landmark points* dari *Mediapipe Hands* dan mengubahnya menjadi data numerik yang tersimpan dalam *spreadsheet* berformat *.csv* agar dapat dilatih. Proses ini dimulai dari ekstraksi fitur, lokalisasi *landmark points* atau *wrist point referencing*, pembersihan data, penggabungan dataset dan diakhiri dengan pemisahan data sebelum dilatih. Proses persiapan data dalam penelitian ini dapat dilihat pada Gambar 3.1.

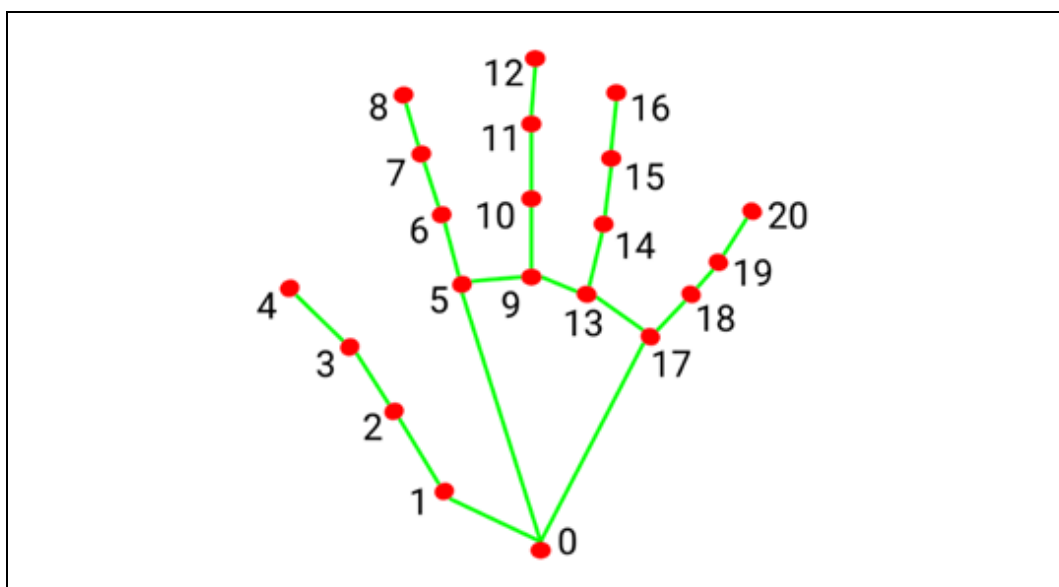


Gambar 3.2 Proses persiapan data

Penjelasan lebih rinci mengenai setiap proses dalam persiapan data seperti yang ditunjukkan di dalam Gambar 3.1 adalah sebagai berikut :

a. Ekstraksi fitur

Proses persiapan data dalam penelitian ini pertama kali akan dimulai dengan melakukan ekstraksi fitur. Ekstraksi fitur menggunakan *Mediapipe Hands* dimulai dengan proses deteksi gestur tangan yang terdapat dalam citra. Hasil dari deteksi ini *Mediapipe Hands* akan menempatkan 21 *keypoints* pada tangan sehingga membentuk sebuah *landmark points*.



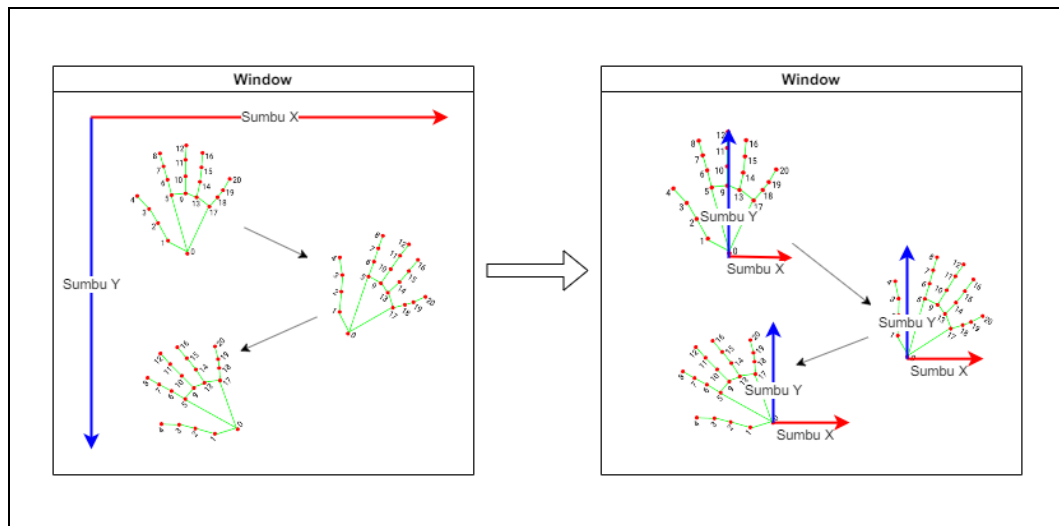
Gambar 3.3 *Landmark points* dari *Mediapipe Hands*

Keypoints dari *landmark* ini memiliki data numerik ternormalisasi tiga dimensi berbentuk koordinat x (data horizontal), koordinat y (data vertikal) dan koordinat z (data kedalaman) yang dapat diekstrak. Namun ekstraksi tidak dilakukan pada koordinat z untuk menetapkan posisi kedalaman tangan. Hal ini membolehkan nilai kedalaman posisi tangan disamaratakan jika terdapat

kumpulan citra yang memiliki ukuran dan kedalaman gestur tangan berbeda serta meringkaskan proses ekstraksi fitur. Hasil akhir dari tahap ini adalah kumpulan data yang diekstrak dan disimpan ke dalam bentuk format penyimpanan yang dapat dihasilkan oleh *framework Mediapipe Hands* yaitu *nested array*.

b. *Wrist point referencing* atau lokalisasi *landmark points*

Wrist point referencing adalah sebuah proses mereferensikan keseluruhan *keypoint* jari yang terdapat dalam *landmark* ke satu sumber *keypoint*, yaitu *point* pergelangan tangan (*keypoint* 0) seperti yang ditunjukkan dalam Gambar 3.3. Tujuan proses ini adalah agar nilai koordinat x dan y pada *landmark points* tidak berubah jika gestur tangan dalam citra berada di posisi dan jarak berbeda. Hal ini karena nilai *landmark points* yang dihasilkan oleh *Mediapipe Hands* secara bawaan berubah berdasarkan posisi relatif dengan ukuran resolusi citra atau kamera seperti yang ditunjukkan pada *window* kiri dalam Gambar 3.3. Perubahan nilai ini akan menyebabkan hasil pelatihan menjadi buruk dan tidak dapat mengklasifikasi gestur abjad SIBI dengan akurat. Maka lokalisasi dilakukan agar nilai *landmark keypoints* tetap relatif stabil sehingga deteksi dapat dilakukan dengan hasil yang konsisten walaupun di berbagai posisi dan jarak seperti yang ditunjukkan pada *window* kanan dalam Gambar 3.3.



Gambar 3.4 Kinerja *wrist point referencing*

Proses ini diawali dengan melakukan denormalisasi nilai *landmark points* yang terdapat dalam *nested array* menjadi nilai piksel sesungguhnya. Data yang telah didenormalisasikan kemudiannya akan dilakukan lokalisasi *keypoints* dengan mereferensikan keseluruhan *keypoint* jari yang terdapat dalam *landmark* kepada satu buah sumber *keypoint* yang telah dijelaskan. Data kemudiannya akan dinormalisasikan kembali dengan nilai *keypoint* x atau y terbesar sebagai ketentuan nilai maksimumnya (1.0 atau 100%).

Proses ini akan diakhiri dengan menyimpan keseluruhan data tersebut ke dalam sebuah dataset baru berformat .csv untuk sebuah abjad. Penyimpanan dilakukan dengan merubah *nested array* menjadi *array* satu dimensi agar dapat dimasukkan ke dalam dataset berformat .csv. Hasil akhir dari proses ini adalah sebuah dataset numerik SIBI yang disimpan dalam *spreadsheet* berformat .csv secara terpisah untuk setiap abjad.

c. Pembersihan Data

Pembersihan data adalah proses mempersiapkan data untuk digunakan dalam tahap pelatihan data dalam *machine learning* yang merujuk pada proses mengidentifikasi dan memperbaiki atau menghapus kesalahan yang ada dalam data. Hasil kehilangan data akan menyebabkan jumlah data yang diekstrak lebih sedikit dari jumlah yang seharusnya. Ini akan mempengaruhi akurasi dan efektivitas pelatihan, sehingga penting untuk membersihkan dan mempersiapkan data sebelum menggunakannya untuk pelatihan atau pengujian.

Tahap pembersihan data dilibatkan dalam penelitian ini karena inkonsistensi kualitas citra yang ada dalam dataset berpotensi menyebabkan *framework Mediapipe Hands* gagal untuk mendeteksi gestur tangan sehingga terdapat kemungkinan terjadinya kehilangan data. Salah satu metode pembersihan data yang digunakan dalam penelitian ini adalah penanganan data hilang. Penanganan data yang hilang dilakukan dengan memasukkan nilai baru dalam setiap dataset melalui penggabungan beberapa baris data yang memiliki gestur yang hampir serupa. Repetisi pemasukan data baru dilakukan berdasarkan berapa jumlah data yang hilang dalam sebuah dataset. Hasil dari tahap ini adalah kumpulan dataset dengan jumlah data yang lengkap.

d. Penggabungan Dataset

Penggabungan dataset adalah proses menggabungkan beberapa dataset menjadi satu dataset terpadu yang lebih besar. Hal ini dilakukan untuk memvariasikan data sehingga algoritma *machine learning* yang akan dipilih dapat belajar dari rentang yang lebih luas dan lebih akurat dalam menangkap pola data.

Tahap penggabungan dataset dalam penelitian ini akan dilakukan dengan menggabungkan keseluruhan dataset yang telah dibersihkan menjadi satu dataset baru terdiri dari keseluruhan gestur abjad lengkap.

e. Pemisahan Data

Tahap pemisahan data (*data splitting*) adalah cara untuk membagi data yang digunakan untuk melatih model *machine learning* menjadi dua bagian atau lebih. Bagian pertama digunakan untuk melatih model (*training set*) dan bagian kedua digunakan untuk menguji model tersebut (*testing set*). Pemisahan data ini penting karena dapat membantu mengetahui seberapa baik performa model pada data yang belum pernah dilihat sebelumnya. Model mungkin akan terlatih terlalu spesifik pada data pelatihan dan berkinerja buruk pada data yang belum pernah dilihat sebelumnya jika tidak dipisah.

Tahap pemisahan dalam penelitian ini dilakukan dengan membagi dataset yang telah digabung dengan rasio 80:20 yang dimana 80% dari dataset akan digunakan untuk *training set*, dan 20% akan digunakan untuk menguji performa *machine learning* (*testing set*). Alasan penggunaan rasio 80:20 untuk pemisahan data didasarkan pada praktik standar *machine learning* yang umum dan pernah dilakukan dalam kumpulan penelitian terkait sebelumnya. Rasio ini umum digunakan karena memberikan jumlah contoh yang cukup untuk melatih model sambil juga menyediakan bagian yang cukup besar dari dataset untuk menguji performa model.

3.2.3 Pelatihan (*Training*)

Training dalam *machine learning* adalah sebuah proses mengajar program komputer untuk membuat prediksi atau keputusan akurat berdasarkan data yang diberi. Program ini dilatih menggunakan algoritma untuk sekumpulan data dengan jumlah yang diketahui dan belajar mengenali pola dalam data yang memungkinkannya untuk membuat prediksi tentang data baru. Tujuan dari pelatihan ini adalah untuk mengembangkan model yang dapat dengan akurat menggeneralisasi dari data *training* ke data baru, sehingga dapat digunakan untuk membuat prediksi atau keputusan di dunia nyata.

Pelatihan dalam penelitian ini dilakukan dengan menggunakan algoritma *Support-Vector Machine* (SVM) yang disediakan oleh *framework Scikit-Learn*. Fungsi kernel yang akan digunakan dalam penelitian ini adalah *linear*. Nilai parameter *default* akan digunakan dalam pelatihan ini karena ia dioptimalkan untuk kasus penggunaan umum.

3.2.4 Evaluasi Model

Evaluasi model adalah proses pengukuran seberapa baik sebuah model *machine learning* dapat membuat prediksi pada data baru (*testing set*) yang belum pernah dilihat sebelumnya. Tujuan dari evaluasi adalah untuk memperoleh pemahaman tentang kekuatan dan kelemahan model, serta untuk mengidentifikasi area di mana mungkin perlu ditingkatkan atau disempurnakan. Evaluasi ini dapat diukur dengan beberapa macam metrik untuk mendapatkan info hasil performa model.

Evaluasi model dalam penelitian ini akan diukur menggunakan beberapa macam metrik umum yaitu akurasi, presisi, *recall*, *F1-score* yang disediakan dan dihitung secara otomatis oleh *Scikit-learn*. Kumpulan metrik ini akan digunakan karena ia memberikan perspektif yang berbeda tentang kinerja model dan relevan untuk berbagai masalah termasuk klasifikasi data numerik SIBI dan umum digunakan dalam kumpulan penelitian terkait sebelumnya. Hasil akhir dari tahap ini adalah sebuah model terlatih yang diekspor dan dapat digunakan untuk prototipe aplikasi deteksi SIBI.

3.3 Prototipe Aplikasi Deteksi SIBI

Bagian ini menjelaskan tentang pembuatan aplikasi deteksi SIBI yang masih berupa prototipe. Pembuatan aplikasi prototipe akan dilakukan setelah proses ekspor model terlatih selesai dilakukan. Adapun hal yang akan dimasukkan dalam pembahasan bagian ini adalah jalan kerja aplikasi dan kumpulan diagram UML seperti *use-case* dan *sequence*. Selain itu, bagian ini juga akan menjelaskan tentang *user interface* dari aplikasi prototipe.

3.4 Pengujian Aplikasi Prototipe

Pengujian aplikasi prototipe adalah proses mengetahui apakah aplikasi yang telah dibuat dapat berfungsi dengan baik sesuai tujuan. Pengujian ini akan dilakukan dengan beberapa cara yaitu pengujian akurasi deteksi dan pengujian eksperimental. Pengujian akurasi deteksi berjalan dengan mendapatkan rata-rata hasil probabilitas akurasi deteksi setiap gestur tangan abjad yang diberikan ke aplikasi melalui kamera dan menganalisis hasil dari akurasi tersebut. Pengujian eksperimental berjalan dengan menguji efektivitas sistem dalam mendeteksi

gestur tangan dalam berbagai posisi ekstrim dan menganalisis juga hasil dari deteksi tersebut.

3.5 Evaluasi

Evaluasi dilakukan ini berupa menganalisis hasil yang didapatkan dari pengujian aplikasi prototipe. Adapun evaluasi ini menjelaskan tentang perbandingan hasil evaluasi model dan hasil *realtime* terlebih dahulu. Kemudian hasil *realtime* ini akan dibandingkan kembali dengan sistem deteksi abjad bahasa isyarat yang pernah dibuat pada penelitian-penelitian sebelumnya.