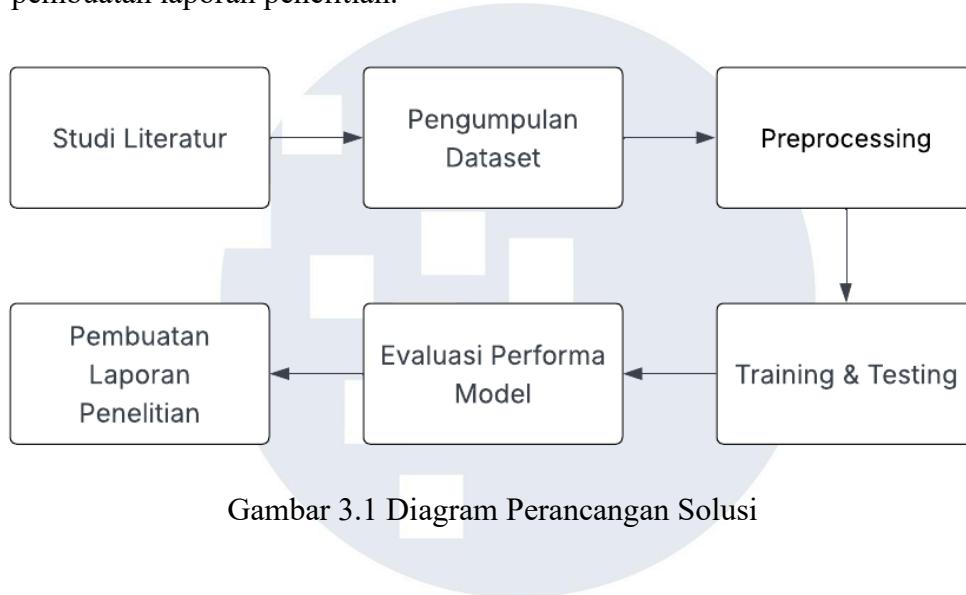


## BAB III

### METODE PENELITIAN

#### 3.1 Perancangan Solusi

Pada penelitian ini terdapat 6 tahap perancangan solusi yang dilakukan oleh penulis untuk menyelesaikan masalah penelitian yaitu: studi literatur, pengumpulan dataset, pra-pemrosesan, training testing, evaluasi, dan pembuatan laporan penelitian.



#### 3.2 Studi Literatur

Tahap pertama yang dilakukan dalam penelitian ini adalah studi literatur. Studi literatur bertujuan untuk memperoleh pemahaman mendalam mengenai konsep, teori, serta penelitian terdahulu yang relevan dengan topik deteksi gerakan tangan untuk penerjemahan Bahasa Isyarat Indonesia (BISINDO). Pada tahap ini, penulis menelaah berbagai sumber referensi berupa jurnal, artikel ilmiah, dan publikasi daring yang membahas tentang bahasa isyarat, khususnya BISINDO sebagai sistem komunikasi utama penyandang disabilitas rungu, serta teknologi yang digunakan dalam pengenalan bahasa isyarat berbasis visi komputer. Selain itu, penulis juga mempelajari literatur terkait metode *classification* dan algoritma YOLO (You Only Look Once), khususnya versi YOLOv8 yang memiliki peningkatan dalam hal akurasi dan efisiensi dibandingkan versi sebelumnya. Studi literatur ini menjadi dasar dalam

merumuskan kerangka penelitian, menentukan metode yang digunakan, serta mengidentifikasi kelebihan dan keterbatasan pendekatan yang telah dilakukan pada penelitian sebelumnya, sehingga penelitian ini dapat memberikan kontribusi baru dalam pengembangan sistem penerjemahan BISINDO berbasis kecerdasan buatan.

### 3.3 Pengumpulan Dataset

Pengumpulan dataset merupakan langkah penting dalam proses penelitian ini. Dataset yang digunakan adalah dataset BISINDO yang dibuat oleh Achmad Noer, yang berisi 312 gambar gerakan tangan untuk alfabet BISINDO (A-Z). Dataset ini sama seperti yang digunakan dalam penelitian sebelumnya dan dapat diakses melalui platform Kaggle. Setiap gambar dalam dataset mewakili satu huruf dari alfabet BISINDO, sehingga terdapat 26 kelas berbeda yang sesuai dengan huruf-huruf dalam alfabet. Dataset ini memiliki variasi tiga jenis background, yaitu body dot, body white, dan wall white. Variasi background ini digunakan untuk meningkatkan keberagaman dalam deteksi gerakan tangan dan memastikan bahwa model dapat bekerja dengan baik dalam berbagai kondisi lingkungan. Proses pengumpulan dataset dilakukan dengan memastikan bahwa gambar-gambar yang digunakan memiliki kualitas yang baik dan representatif untuk deteksi gerakan tangan. Gambar-gambar dalam dataset ini sudah dilabeli dengan benar berdasarkan kelas yang sesuai, yang memudahkan proses pelatihan model.



Gambar 3.2 Dataset Alfabet

### **3.4 Preprocessing**

Pra-pemrosesan (preprocessing) pada penelitian ini bertujuan untuk menyiapkan dataset agar siap digunakan pada tahap pelatihan model klasifikasi. Berbeda dengan pendekatan object detection yang membutuhkan anotasi bounding box, pada pendekatan image classification setiap gambar hanya diberi label kelas (misalnya A–Z) sesuai gerakan tangan BISINDO yang ditampilkan pada gambar. Langkah preprocessing yang dilakukan adalah sebagai berikut:

#### **3.4.1 Pelabelan Kelas (Image Labeling)**

Proses pelabelan dilakukan menggunakan platform Roboflow dengan cara mengunggah seluruh gambar, kemudian menetapkan label kelas untuk setiap gambar berdasarkan 26 kelas alfabet BISINDO (A–Z). Pada tahap ini tidak dilakukan pembuatan bounding box karena tugas yang digunakan adalah klasifikasi.

#### **3.4.2 Pembagian Dataset (Dataset Split)**

Setelah pelabelan selesai, dataset dibagi menjadi tiga bagian, yaitu data training, validation, dan testing dengan rasio 60:20:20. Pembagian ini dipilih agar data training cukup untuk proses pembelajaran, sementara validation dan testing tetap tersedia untuk evaluasi selama training dan pengujian akhir.

Berdasarkan total 312 gambar, pembagian menghasilkan:

- Training: 187 gambar
- Validation: 62 gambar
- Testing: 63 gambar

Pembagian dataset dan pelabelan kelas dilakukan melalui Roboflow pada mode klasifikasi, kemudian diekspor ke format Ultralytics YOLOv8 Classification agar dapat digunakan langsung pada proses pelatihan di Google Colab.

### **3.4.3 Ekspor Dataset**

Dataset yang sudah diberi label dan dibagi sesuai rasio kemudian diekspor dari Roboflow ke format yang kompatibel dengan Ultralytics YOLOv8 Classification, sehingga dapat langsung digunakan pada proses training melalui Google Colaboratory.

Pada penelitian ini tidak dilakukan augmentasi dataset secara manual di Roboflow. Namun, pada tahap training digunakan opsi augment=True yang berarti augmentasi dilakukan secara on-the-fly selama pelatihan (augmentasi bawaan), bukan menambah jumlah file gambar pada dataset.

## **3.5 Training & Testing**

Proses pelatihan (training) dan pengujian (testing) dilakukan di Google Colaboratory menggunakan framework Ultralytics YOLOv8 pada mode klasifikasi (task=classify). Model yang digunakan adalah yolov8s-cls.pt (pre-trained) untuk menerapkan transfer learning, sehingga model dapat mempelajari pola gerakan tangan BISINDO dengan lebih efektif meskipun jumlah data terbatas. Proses pelatihan dan pengujian menggunakan dataset yang telah diproses sebelumnya.

Perintah training yang digunakan adalah sebagai berikut:

```
!yolo task=classify mode=train model=yolov8s-cls.pt data=$DATA_DIR \
    imgsz=320 epochs=220 batch=64 lr0=0.0015 lrf=0.01 optimizer=AdamW \
    weight_decay=0.012 dropout=0.2 patience=20 \
    cos_lr=True augment=True cache=True workers=8 seed=42 device=0
```

Pelatihan dilakukan dengan parameter utama sebagai berikut:

- imgsz=320: ukuran input gambar pada pelatihan.
- epochs=220: jumlah iterasi pelatihan penuh terhadap seluruh data training.

- batch=64: jumlah sampel per langkah pembaruan bobot.
- lr0=0.0015 dan lrf=0.01: learning rate awal dan faktor learning rate akhir pada skema penurunan LR.
- optimizer=AdamW dan weight\_decay=0.012: optimizer AdamW + regularisasi untuk membantu mengurangi overfitting.
- dropout=0.2: dropout pada bagian classifier untuk meningkatkan generalisasi.
- patience=20: early stopping jika tidak ada peningkatan performa validasi selama 20 epoch.
- cos\_lr=True: penjadwalan LR cosine agar pembelajaran lebih stabil.
- augment=True: augmentasi bawaan dilakukan selama training (on-the-fly).
- cache=True: cache dataset untuk mempercepat loading.
- workers=8: jumlah worker untuk data loading.
- seed=42: menjaga konsistensi hasil eksperimen.
- device=0: menjalankan training pada GPU.

### 3.6 Evaluasi Performa Model

Setelah model selesai dilatih, tahap selanjutnya adalah evaluasi performa model menggunakan metrik yang relevan. Evaluasi dilakukan untuk mengetahui sejauh mana model dapat mendeteksi gerakan tangan BISINDO dengan akurasi yang tinggi. Beberapa metrik utama yang digunakan untuk mengevaluasi kinerja model antara lain:

1. Precision: Metrik yang mengukur akurasi model dalam mendeteksi kelas yang benar (positif prediksi yang benar). Precision dihitung dengan rumus:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Di mana:

- TP (True Positives): Jumlah prediksi benar sebagai positif.
- FP (False Positives): Jumlah prediksi salah sebagai positif.

Precision tinggi = prediksi model untuk suatu kelas umumnya benar (FP rendah). Artinya model jarang “salah menebak” kelas tersebut. Sedangkan jika Precision rendah = banyak prediksi keliru (FP tinggi). Artinya model sering memberi label suatu kelas padahal seharusnya kelas lain.

2. Recall: Mengukur seberapa baik model dalam menemukan semua contoh dari kelas yang benar. Recall dihitung dengan rumus:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Di mana:

- TP (True Positives): Jumlah prediksi benar sebagai positif.
- FN (False Negatives): Jumlah prediksi salah sebagai negatif.

Recall tinggi = model berhasil mengenali sebagian besar sampel kelas tersebut (FN rendah). Artinya model jarang “melewatkannya” kelas yang seharusnya dikenali. Sedangkan jika Recall rendah = banyak sampel kelas tersebut tidak terdeteksi (FN tinggi). Artinya model sering gagal mengenali kelas yang benar.

3. F1-Score: Kombinasi dari precision dan recall yang memberikan gambaran seimbang dari kinerja model. F1-Score dihitung dengan rumus:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-Score tinggi = precision dan recall sama-sama baik (model tidak hanya tepat, tetapi juga mampu menemukan sebagian besar sampel yang benar). Sedangkan jika F1-Score rendah = salah satu (atau keduanya) dari precision/recall rendah, sehingga performa model belum stabil untuk klasifikasi yang konsisten.

Selain ketiga metrik tersebut, akurasi (accuracy) dapat digunakan sebagai gambaran umum persentase prediksi yang benar pada data uji. Namun, untuk analisis yang lebih informatif per kelas, precision–recall–F1 serta confusion matrix lebih membantu untuk melihat kelas mana yang paling sering tertukar.

