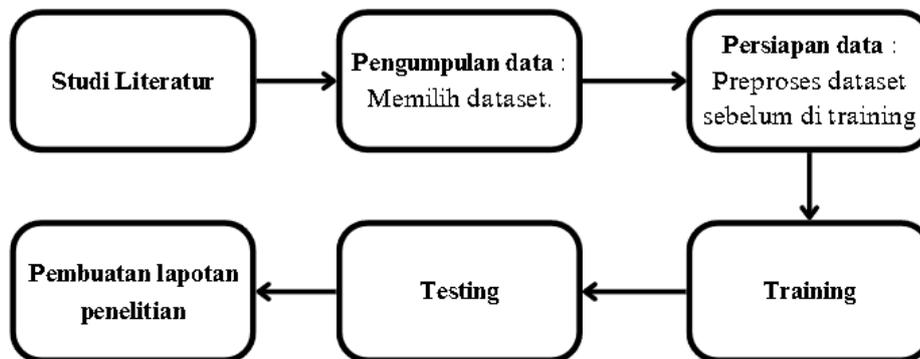


## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Pada penelitian ini, penulis melakukan 6 tahapan penelitian untuk mendapatkan hasil penelitian yaitu dengan studi literatur, pengumpulan data, persiapan data, training, testing, dan pembuatan laporan penelitian. Agar lebih jelas untuk alur metode penelitian dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram tahapan penelitian

#### 3.2 Studi literatur

Studi literatur dilakukan untuk mencari informasi berupa penelitian terdahulu dan juga beberapa dokumentasi teknologi yang berkaitan dengan penelitian ini. Dimana penelitian ini terkait dengan *face recognition*, *unsupervised face recognition*, *dataset MSIM-RetinaFace*, *model DeiT (Data-efficient Image Transformer)*, *N:N Evaluation (R-R)* dan *N:M Evaluation (R-S)*, *Rank-1 Accuracy* dan *True Acceptance Rate (TAR)*, *Silhouette Score* dan *Davies-Bouldin Index*. Tujuan tahap ini agar memberikan informasi yang dibutuhkan kepada penulis.

#### 3.3 Pengumpulan data

Pengumpulan data dilakukan dengan *download dataset MSIM-RetinaFace*. Untuk *dataset MSIM\_RetinaFace* merupakan dataset yang tidak memiliki label. *Dataset* ini digunakan sebagai *dataset* awal yang nanti akan di *preprocess* dan *training* dan juga *testing*. *Dataset MS1M-Retinaface* yang digunakan berjumlah 16586, yang didapatkan dari *Sample Size* dengan *Confidenece Level 99%*, *Margin of Error 1%*, *Population Proportion 50%*, dan *Population Size 5000000*. Dibawah ini adalah *sample* dari *dataset* yang saya gunakan :



Gambar 3.2 *Sample dataset MS1M-Retinaface*

### 3.4 **Persiapan data**

Persiapan data dilakukan untuk *preprocess dataset MS1M-RetinaFace*, *dataset MS1M-RetinaFace* akan di *preprocess* sesuai dengan *model DeiT*. *Preprocess* yang dilakukan berupa *resize* gambar menjadi 224 x 224 pixel, mengonversi gambar menjadi *tensor PyTorch*.

### 3.5 **Training**

Pada tahap pelatihan sistem *unsupervised face recognition* menggunakan *DeiT* (*Data-Efficient Image Transformer*), proses dimulai dengan mempersiapkan *dataset* melalui kelas *FaceDataset*, yang memuat dan memvalidasi *citra wajah* dari *direktori* secara *asinkron* menggunakan beberapa *thread* untuk meningkatkan *performa*, dengan 80% gambar diambil secara acak dari total yang tersedia. Gambar tersebut

diterapkan beberapa transformasi seperti `RandomResizedCrop`, `RandomHorizontalFlip`, dan normalisasi sebelum digunakan dalam pelatihan. Model DeiT yang digunakan diinisialisasi tanpa menggunakan pretrained weights, dan lapisan keluaran terakhirnya dimodifikasi untuk menghasilkan 128 fitur yang sesuai dengan tujuan pembelajaran representasi wajah, dan loss yang digunakan merupakan kombinasi dari Cross-Entropy Loss dan Kullback-Leibler Divergence (KL-Divergence) yang dirancang untuk memastikan model belajar representasi fitur yang serupa untuk gambar yang mirip secara semantik. Proses pelatihan dijalankan selama 100 epoch dengan optimisasi Adam dan pembaruan presisi campuran menggunakan GradScaler untuk efisiensi memori, serta penyesuaian learning rate menggunakan scheduler CosineAnnealingLR. Setelah setiap epoch, model dan state optimizer disimpan dalam checkpoint untuk memudahkan kelanjutan pelatihan jika diperlukan, dan di akhir pelatihan, loss divisualisasikan melalui grafik untuk melihat tren penurunan loss selama pelatihan. Model yang telah dilatih kemudian disimpan sebagai 'unsupervised\_face\_model\_deit\_TRAINING.pth', dan keseluruhan proses berjalan di atas PyTorch dengan dukungan CUDA, memanfaatkan GPU jika tersedia, untuk mempercepat proses komputasi.

### 3.6 Testing

Bagian testing dari kode ini berfokus pada evaluasi performa model unsupervised face recognition menggunakan DeiT (Data-Efficient Image Transformer) melalui beberapa tahap. Pertama, dataset wajah di-load menggunakan kelas `FaceDataset`, yang melakukan validasi gambar dan memilih 20% dari dataset untuk evaluasi, memastikan setiap kelas memiliki jumlah gambar yang cukup dengan memanfaatkan multithreading untuk mempercepat prosesnya. Model DeiT yang telah dilatih sebelumnya dimuat dan digunakan untuk mengekstrak fitur dari gambar-gambar tersebut, yang kemudian dinormalisasi untuk konsistensi. Evaluasi dilakukan dalam dua skenario utama: N:N(R-R), yang membandingkan semua gambar dalam dataset yang sama untuk menghitung True Accept Rate (TAR) dan False Accept Rate (FAR) menggunakan cosine similarity, dan N:M(R-S), yang membandingkan fitur probe dengan gallery untuk menghitung metrik serupa.

Akurasi Rank-1 juga dihitung untuk mengukur keakuratan prediksi gambar yang paling mirip. Selain itu, evaluasi clustering dilakukan menggunakan KMeans dengan dua metrik utama: Silhouette Score dan Davies-Bouldin Index, untuk mengukur kualitas cluster yang terbentuk berdasarkan fitur gambar. Hasil evaluasi kemudian divisualisasikan dalam bentuk kurva ROC untuk kedua skenario (N:N) dan (N:M), yang menggambarkan hubungan antara TAR dan FAR, serta disimpan sebagai file gambar. Evaluasi ini bertujuan untuk memberikan analisis menyeluruh tentang kinerja model face recognition yang diimplementasikan secara unsupervised.

