

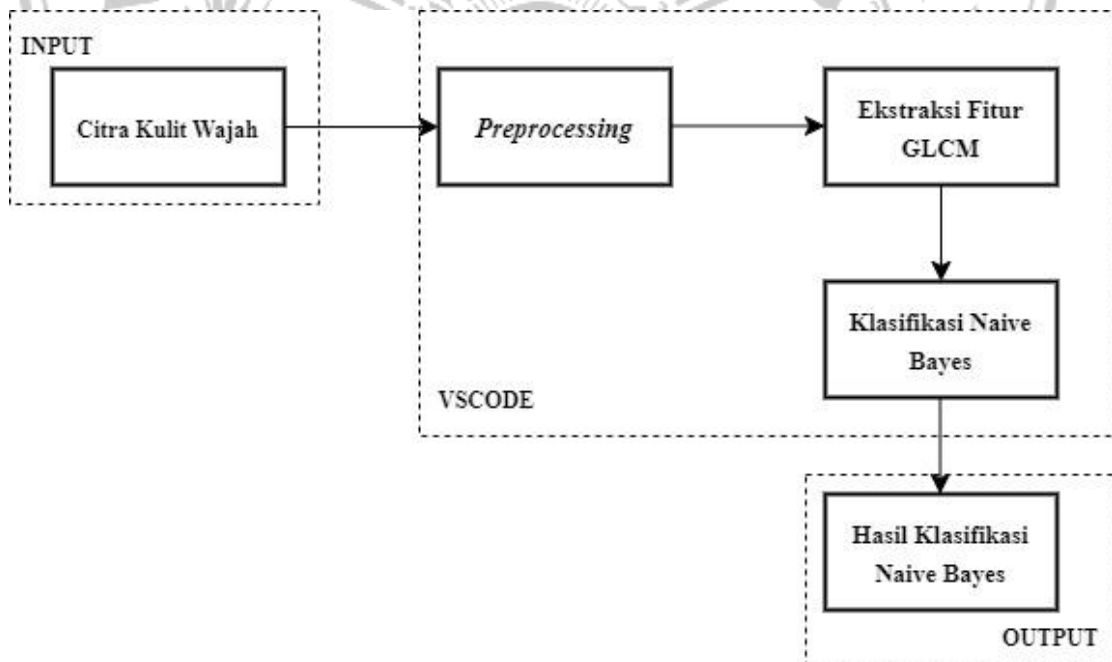
BAB III

METODE PENELITIAN

Bab ini akan menjelaskan mengenai bagaimana tahapan dan metode yang digunakan pada penelitian yang dilakukan untuk proses klasifikasi dan ekstraksi ciri pada jenis kulit wajah dengan metode *Naïve Bayes* dan metode *Gray Level Co-Occurrence Matrix* (GLCM).

3.1 Tahapan Penelitian

Pada bab ini akan menjawab permasalahan yang telah dirumuskan di atas, dengan pengembangan sebuah system menggunakan metode GLCM untuk melakukan ekstraksi ciri berupa tekstur citra yang kemudian akan diklasifikasi dengan menggunakan metode *Naïve Bayes* untuk mengetahui jenis kulit wajah. Maka diusulkan metode perancangan sistem secara umum yang digambarkan dengan menggunakan blok diagram pada gambar di bawah, sebagai berikut :



Gambar 3.1 Diagram blok Perancangan Sistem

3.2 Prosedur Pengujian


3.2.1. Pengumpulan Data


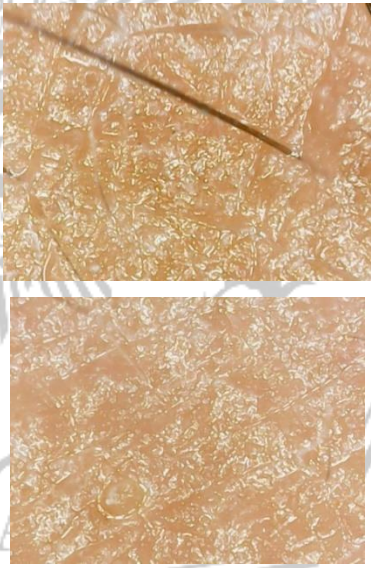
Pada tahap ini, pengambilan data pada kulit wajah yang dilakukan secara primer yaitu pengambilan data secara langsung dengan menggunakan alat *microscope* digital dan sekunder yaitu melalui website Telkom *University Dataverse*. Pada web tersebut sudah memberikan data kulit wajah dengan kategori yang penulis butuhkan untuk penulisan tugas akhir. Kemudian hasil citra yang sudah didapat akan diklasifikasikan dengan metode Naïve Bayes sesuai dengan kategori jenis kulit wajah. Input ini sendiri akan dibagi menjadi dua data, yaitu data training dan data testing. Dengan sebesar 70% data yang diambil adalah sebagai data training, dan 30% data yang diambil sebagai data testing.

Tabel 3.1 Data Image Total

| Kulit Wajah Kering | Kulit Wajah Berminyak | Kulit Wajah Normal | TOTAL |
|---------------------------|------------------------------|---------------------------|-----------------------|
| 52 data image | 64 data image | 76 data image | 192 data image |

Tabel 3.2 Contoh Citra Kulit Wajah

| Jenis Kulit Wajah | Citra Kulit Wajah |
|-------------------|--------------------------------------------------------------------------------------|
| Kulit Normal |  |

| | |
|-----------------|--------------------------------------------------------------------------------------|
| | |
| Kulit Kering |  |
| Kulit Berminyak |  |

3.2.2. Labelling Data

Pada penelitian ini, pemberian label pada data primer yang dilakukan yaitu memberikan keterangan gambar terkait tipe kulit wajah yang dimiliki berdasarkan ciri ciri kulit wajah. Pemberian label dilakukan secara manual dengan menentukan tipe kulit tersebut ke dalam 3 golongan, yaitu kulit wajah

normal, kulit wajah kering, dan kulit wajah berminyak untuk dipindah kedalam folder sesuai tipe kulit wajah. Langkah – langkah yang dilakukan untuk proses *labelling* yaitu :

1. Mengumpulkan data.
2. Memberikan label pada data image berdasarkan ciri – ciri kulit wajah yang dimiliki.
3. Data yang sudah ter – label dapat digunakan proses ekstraksi ciri dengan metode GLCM dan proses klasifikasi dengan metode *Naïve Bayes* untuk mendapatkan nilai akurasi.

Pada data sekunder, image yang didapatkan sudah terdapat *labelling* sesuai tipe kulit wajah. Terdapat perbedaan yang terjadi jika data yang didapatkan memiliki label dan tidak memiliki label. Pada data yang ter – label dapat langsung digunakan untuk Ekstraksi Ciri dengan menggunakan metode GLCM dan diklasifikasi dengan menggunakan metode *Naïve Bayes* sehingga menghasilkan output berupa nilai akurasi.

3.2.3. Pre – Processing Data

Pre – processing data pada penelitian ini yaitu mengubah suatu citra asli menjadi citra keabuan, yang mana pada proses ini menggunakan software VSCode untuk mengolah programnya yang dapat dilihat seperti gambar berikut:

```
direktori = f"data_set/kering/{i}"  
im = Image.open(direktori).convert('L')
```

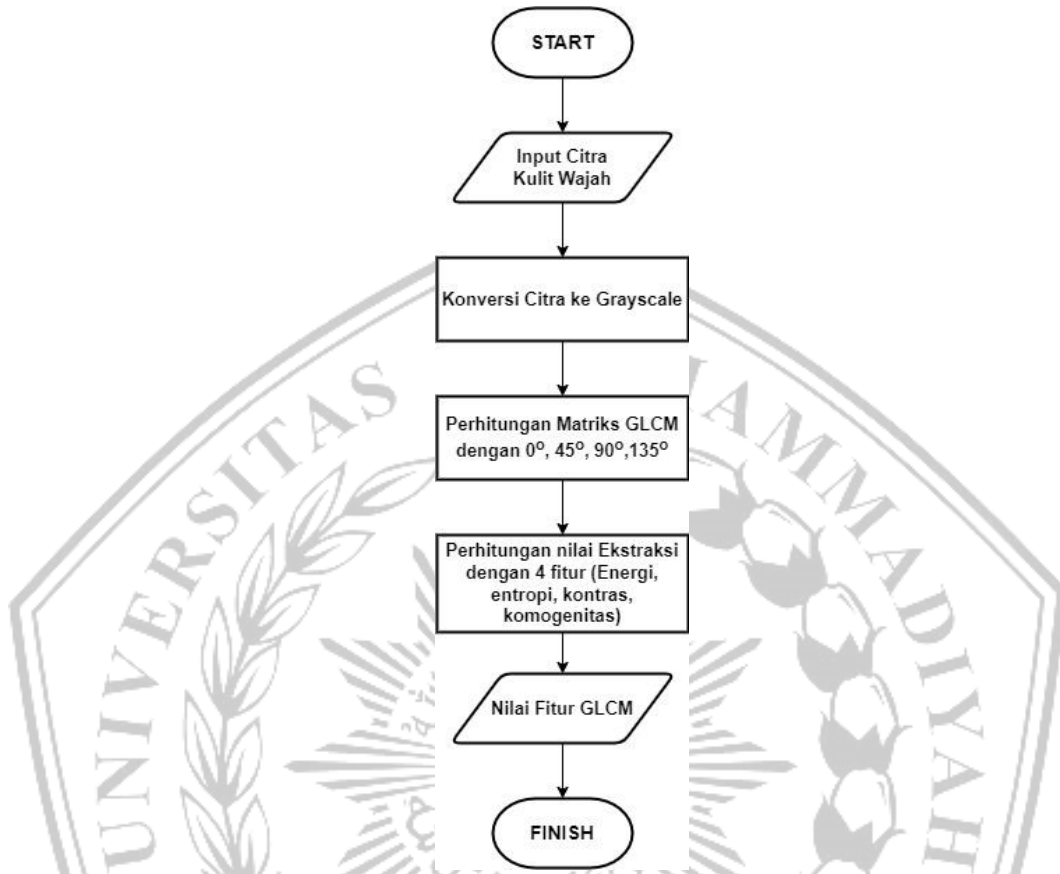
Gambar 3.2 Program *Preprocessing*

Pada gambar diatas dapat terlihat “convert (‘L’)” digunakan untuk mengonversi sebuah citra asli menjadi citra berskala abu – abu (*grayscale*). Dengan cara ini, citra akan direpresentasikan sebagai sebuah matrix intensitas piksel dengan satu saluran warna yang kemudian akan lebih mudah untuk melakukan proses selanjutnya.

3.2.4. Alur Perancangan Algoritma GLCM

Perancangan algoritma digunakan untuk menjelaskan rancangan yang kemudian akan dijadikan sebagai dasar penelitian. Pada penelitian ini system akan bekerja untuk melakukan proses ekstraksi ciri terlebih dahulu

menggunakan metode GLCM untuk menghasilkan data input yang akan digunakan pada proses selanjutnya sesuai pada gambar 3.3 berikut :



Gambar 3.3 Diagram Blok Perancangan Sistem

Pada diagram diatas, menjelaskan bagaimana tahapan perhitungan ekstraksi ciri pada metode GLCM dari hasil sebuah citra hasil *preprocessing* yaitu citra keabuan (*grayscale*). Langkah pertama yang dilakukan yaitu melakukan proses perhitungan matriks dengan berbagai arah sudut. Pada penelitian ini menggunakan arah sudut $0^\circ, 45^\circ, \text{ dan } 90^\circ$. Selanjutnya melakukan perhitungan pada ciri GLCM dengan 4 fitur yaitu Energi, Entropi, Kontras, dan Homogenitas untuk menghasilkan dataset berupa nilai fitur GLCM yang akan diproses untuk klasifikasi menggunakan metode *Naïve Bayes*.

```
for i in keringl:  
    direktori = f"data_set/kering/{i}"  
    im = Image.open(direktori).convert('L')  
    img = np.array(im)
```

Gambar 3.4 Program Dataset

Dijelaskan pada gambar diatas bahwa dengan banyaknya data image yang digunakan maka perlu menggunakan “for” atau berulang untuk menghitung nilai data secara bergantian.

```
glcm = graycomatrix(img, distances=[jarak], angles=[sudut])
```

Gambar 3.5 Program Matrix GLCM

Kemudian “graycomatrix” adalah matrix pada metode GLCM yang berisi informasi tentang hubungan intensitas piksel pada jarak dan sudut tertentu. Matrix ini digunakan untuk menghitung berbagai fitur pada GLCM seperti *contrast*, *homogeneity*, *energy*, *entropy*, dan lainnya. Jarak yang digunakan dapat berbeda – beda sesuai yang diinginkan contohnya: 1, 2, 3, dan seterusnya. Untuk sudut juga yang digunakan dapat berbeda – beda, yaitu sudut 0°, 45°, 90°, 135°, 180, dan lainnya.

```
contrast = graycoprops(glcm, 'contrast')
homogeneity = graycoprops(glcm, 'homogeneity')
energy = graycoprops(glcm, 'energy')
p = glcm/glcm.sum()
entropy = -np.sum(p*np.log2(p+1e-10))
contrast1.append(np.round(contrast.flatten(),4))
homogeneity1.append(np.round(homogeneity.flatten(),4))
energy1.append(np.round(energy.flatten(),4))
entropy1.append(np.round(entropy,4))
```

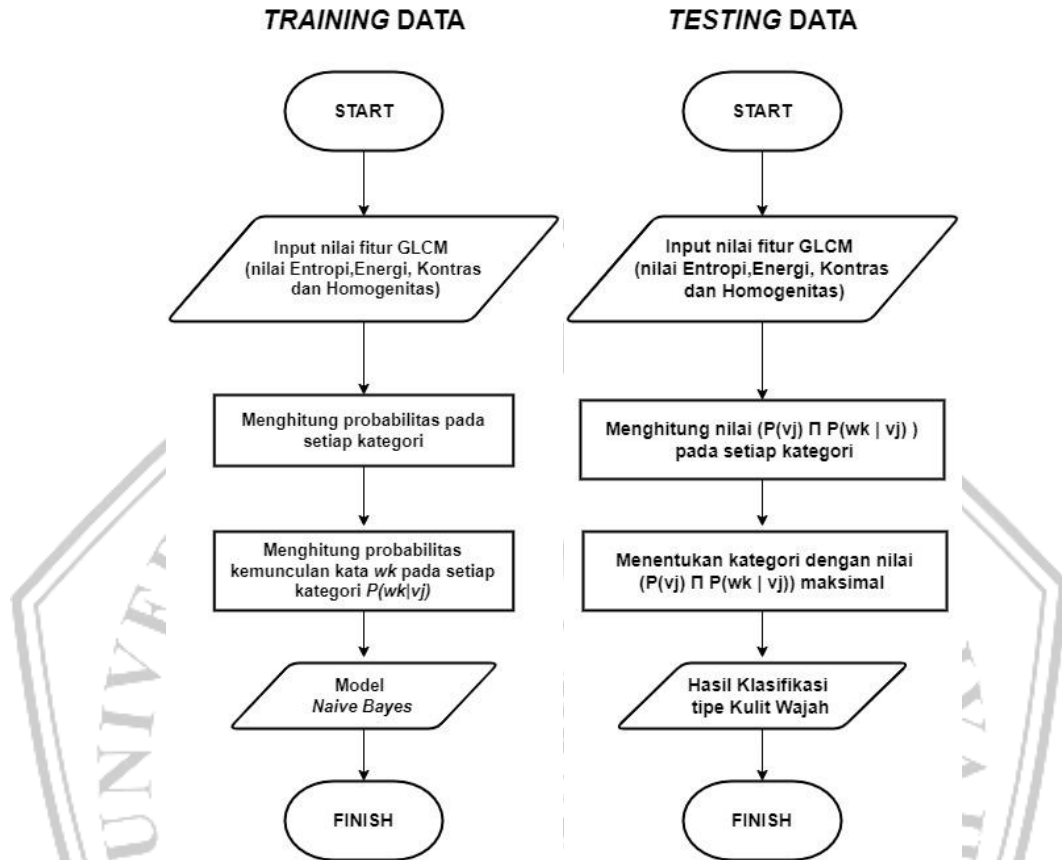
Gambar 3.6 Program Parameter GLCM

Pada penelitian ini, parameter GLCM yang digunakan ada 4 yaitu *contrast*, *homogeneity*, *energy*, dan *entropy* yang mana pada program cukup ditulis “graycoprops” dan diberikan input berupa nilai matrix GLCM dan perintah sesuai dengan parameter tersebut. Dan pada program tersebut akan menghasilkan sebuah data set yang mana data tersebut dapat digunakan untuk diproses pada metode Naïve Bayes.

3.2.5. Alur Perancangan Algoritma Naïve Bayes

Kemudian, pada proses klasifikasi penelitian ini menggunakan metode

Naïve Bayes yang bekerja dengan alur pembagian data menjadi dua proses yaitu proses data *training* dan proses data *testing* sesuai pada gambar 3.4 berikut :



Gambar 3.7 Diagram Blok Proses Klasifikasi Metode *Naïve Bayes*

Pada diagram alir diatas, terdapat sistem perancangan klasifikasi dengan metode *Naïve Bayes*. Proses klasifikasi ini dimulai dengan memberikan sejumlah data berupa nilai dari fitur GLCM yang menjadi acuan untuk membuat klasifikasi data. Data ini disebut dengan tahap *training* yang kemudian akan dibuat suatu model untuk menentukan nilai probabilitas dari sampel data yang sudah didapatkan.

Kemudian pada data *testing* dimulai juga dengan memberikan data berupa berupa nilai dari fitur GLCM yang kemudian akan diproses untuk penentuan kelas dari data uji yang belum diketahui kelasnya dengan menggunakan rumus probabilitas. Hasil dari klasifikasi dengan metode *Naïve*

Bayes akan menghasilkan 3 kelas yaitu kulit normal, kulit kering dan kulit berminyak serta akan menampilkan nilai akurasi dari metode *Naïve Bayes*.

```
model = GaussianNB()  
model.fit(X_train,y_train)
```

Gambar 3.8 Program Naïve Bayes

Dijelaskan pada gambar diatas bahwa untuk metode Naïve Bayes dengan menggunakan VSCode cukup memberikan perintah berupa “GaussianNB()”. Kemudian pada gambar juga terdapat perintah “X_train” yang berfungsi untuk menampung data yang akan di *training*, sedangkan perintah “Y_train” berfungsi untuk menampung data target yang akan di *testing*.

3.2.6. Perhitungan Akurasi Dalam Program

```
false = all-true  
acc = true/all*100  
xtest = X_test.tolist()  
print(f"Akurasi = {acc}")
```

Gambar 3.9 Program Menghitung Akurasi

Dijelaskan pada program diatas bahwa untuk menghitung nilai akurasi pada metode *Naïve Bayes*. Terlihat pada program, untuk mengitung akurasi yaitu “Akurasi = {acc}” dengan “acc = true/all*100” yang mana nilai acc didapat dengan membagi jumlah data prediksi yang benar dengan jumlah data total kemudian dikalikan dengan 100%. Hasilnya akan dalam bentuk presentasi akurasi.