**BAB 4. HASIL DAN PEMBAHASAN**

Metode yang digunakan pada penelitian ini adalah *waterfall*. Tahapan-tahapan yang dilakukan berdasarkan metode tersebut adalah sebagai berikut :

**4.1 Identifikasi Masalah**

*Requirement Analysis* atauanalisis kebutuhan merupakan salah satu tahapan dari metode penelitian ini dalam menentukan suatu masalah yang akan diteliti oleh si peneliti agar dapat dijadikan objek penelitian. Pada tahap ini bertujuan untuk menganalisis kebutuhan sistem dan menentukan solusi dari hasil analisis yang didapat. Hal-hal yang dilakukan yaitu :

4.1.1 Studi Literatur

Pada studi literatur, penulis mencari referensi dari jurnal dan *website* resmi yang berhubungan dengan daging, pengolahan citra digital menggunakan fitur warna dan metode K-NN. Selain itu juga mempelajari data citra gambar yang diperoleh dari pedagang dan pemburu sebagai objek penelitian.

4.1.2 Pengumpulan Data

Pengumpulan data merupakan tahapan penelitian, karena pada tahap ini menghasilkan data yang akan dijadikan objek penelitian. Peneliti memilih pedagang daging dan pemburu sebagai narasumber untuk mendapat informasi mengenai objek penelitian, yaitu daging. Peneliti menemui langsung narasumber ke tempat tinggalnya. Kemudian peneliti melakukan wawancara terhadap narasumber tersebut.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pertanyaan | Jawaban | |
| Ya | Tidak |
| 1 | Dari sumber yang saya baca, yaitu dari penelitian | Iya, bahkan ada warnanya yang merah tua | - |
| Roron Wicaksono Hadi, Iwan Setiawan dan Sumardi |
| yang berjudul "Perancangan Alat Pendetaksi Kualitas Daging Sapi |
| Berdasar Warna dan Bau Berbasis Mikrokontroler Atmega32 |
| Menggunakan Logika Fuzzy" |
| apakah benar bahwa daging sapi memiliki ciri : ? |
| 1. Daging sapi memiliki warna merah terang |
| 2. Kadar air sedikit |
| 3. Baunya amis segar |
| 2 | Dari sumber yang saya baca, yaitu dari penelitian | Iya, warnanya lebih pucat dari daging sapi, agak kasar juga | - |
| Roron Wicaksono Hadi, Iwan Setiawan dan Sumardi |
| yang berjudul "Perancangan Alat Pendetaksi Kualitas Daging Sapi |
| Berdasar Warna dan Bau Berbasis Mikrokontroler Atmega32 |
| Menggunakan Logika Fuzzy" |
| apakah benar bahwa daging babi hutan memiliki ciri : ? |
| 1. Dagingnya lebih pucat |
| 2. Kadar air banyak |
| 3. Lebih amis dan cenderung busuuk |
| 3 | Bagaimana cara membedakan keduanya ? | Dari warnanya sudah terlihat, daging sapi lebih merah | |
| 4 | Jadi beda dari keduanya bisa dilihat dari warnanya ? | Iya |  |

Tabel diatas adalah hasil wawancara Bersama pedagang daging sapi dengan nama narasumber Bapak Abu yang bekerja sebagai penjual daging sapi di Pasar Maesan Wawancara dilakukan pada tanggal 31 Juli 2021. Sebagai hasil dari wawancara diperoleh informasi bahwa kedua jenis citra dapat dibedakan berdasarkan warnanya dan membeli beberapa daging untuk dijadikan bahan penelitian.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pertanyaan | Jawaban | |
| Ya | Tidak |
| 1 | Dari sumber yang saya baca, yaitu dari penelitian | Iya, bahkan ada warnanya yang merah tua | - |
| Roron Wicaksono Hadi, Iwan Setiawan dan Sumardi |
| yang berjudul "Perancangan Alat Pendetaksi Kualitas Daging Sapi |
| Berdasar Warna dan Bau Berbasis Mikrokontroler Atmega32 |
| Menggunakan Logika Fuzzy" |
| apakah benar bahwa daging sapi memiliki ciri : ? |
| 1. Daging sapi memiliki warna merah terang |
| 2. Kadar air sedikit |
| 3. Baunya amis segar |
| 2 | Dari sumber yang saya baca, yaitu dari penelitian | Iya, warnanya lebih pucat dari daging sapi | - |
| Roron Wicaksono Hadi, Iwan Setiawan dan Sumardi |
| yang berjudul "Perancangan Alat Pendetaksi Kualitas Daging Sapi |
| Berdasar Warna dan Bau Berbasis Mikrokontroler Atmega32 |
| Menggunakan Logika Fuzzy" |
| apakah benar bahwa daging babi hutan memiliki ciri : ? |
| 1. Dagingnya lebih pucat |
| 2. Kadar air banyak |
| 3. Lebih amis dan cenderung busuuk |
| 3 | Bagaimana cara membedakan keduanya ? | Kalau dari saya bisanya dari warnanya, sapi lebih cerah dari pada babi hutan. Dan pernah merasakan daging babi hutan, daging babi hutan lebih lemak (sedap) | |
| 4 | Jadi beda dari keduanya bisa dilihat dari warnanya ? | Iya |  |

Tabel diatas adalah hasil wawancara Bersama pemburu dengan nama narasumber Bapak Wawan Wawancara dilakukan pada tanggal 7 Agustus 2021. Sebagai hasil dari wawancara diperoleh informasi bahwa kedua jenis citra dapat dibedakan berdasarkan warnanya dan membeli beberapa daging untuk dijadikan bahan penelitian.

Setelah melakukan wawancara, peneliti mendapatkan kesimpulan bahwa objek penelitian yang akan dilakukan, yaitu daging sapi dan daging babi hutan dapat dibedakan berdasarkan warnanya, hanya saja perlu ketelitian dalam melakukannya. Salah satu perbedaannya yaitu warna daging sapi cenderung merah terang sedangkan daging babi cenderung lebih pucat. Setelah itu peneliti membeli daging untuk dijadikan data penelitian. Daging yang diperoleh kemudian dipotong-potong menjadi ukuran yang lebih kecil, kemudian dilakukan pengambilan gambar pada objek yang menghasilkan jumlah data seperti disajikan pada tabel 4.1 sebagai berikut :

|  |  |
| --- | --- |
| Jenis Daging | Jumlah |
| Daging Sapi | 30 Buah |
| Daging Babi Hutan | 30 Buah |

Kemudian dari data tersebut, dilakukan proses ekstraksi fitur warna RGB ke HSV untuk dijadikan sebagai *dataset*. Adapun nilai yang diperoleh disajikan pada tabel 4.2 berikut ini :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama File | Nilai H | Nilai S | Nilai V | Label |
| IMG\_0517.JPG | 22,85084386 | 22,5449195 | 21,81181578 | Babi |
| IMG\_0518.JPG | 21,18323402 | 18,34339355 | 18,58545646 | Babi |
| IMG\_0520.JPG | 15,30213009 | 14,69127552 | 13,71130282 | Babi |
| IMG\_0522.JPG | 22,43043243 | 15,91877313 | 18,56630861 | Babi |
| IMG\_0525.JPG | 18,43665622 | 15,76907579 | 15,27913796 | Babi |
| IMG\_0526.JPG | 19,76140782 | 14,57432659 | 14,68870483 | Babi |
| IMG\_0527.JPG | 21,93115599 | 17,84111993 | 20,40222244 | Babi |
| IMG\_0528.JPG | 33,34589954 | 31,21936197 | 33,34854376 | Babi |
| IMG\_0529.JPG | 24,8575559 | 19,99230851 | 22,42111146 | Babi |
| IMG\_0530.JPG | 17,80835681 | 14,80925097 | 12,07782673 | Babi |
| IMG\_0532.JPG | 21,87006818 | 12,40545199 | 18,55851123 | Babi |
| IMG\_0533.JPG | 17,94834493 | 17,69023725 | 15,21386763 | Babi |
| IMG\_0534.JPG | 22,04835199 | 21,86600625 | 21,32688699 | Babi |
| IMG\_0535.JPG | 19,19156671 | 13,82142206 | 13,84411123 | Babi |
| IMG\_0536.JPG | 14,80450077 | 14,1537828 | 11,07007936 | Babi |
| IMG\_0538.JPG | 16,06308495 | 15,73272135 | 13,78781479 | Babi |
| IMG\_0539.JPG | 19,61254287 | 15,91124046 | 16,66043014 | Babi |
| IMG\_0540.JPG | 18,26166666 | 14,75418694 | 15,98823775 | Babi |
| IMG\_0541.JPG | 20,51650068 | 15,76696688 | 17,57942974 | Babi |
| IMG\_0542.JPG | 25,20520257 | 20,86737689 | 21,25077797 | Babi |
| IMG\_0543.JPG | 23,28280576 | 17,77979093 | 16,85139976 | Babi |
| IMG\_0544.JPG | 18,86087957 | 15,07157648 | 13,55852006 | Babi |
| IMG\_0546.JPG | 14,78421464 | 13,80208891 | 11,85272041 | Babi |
| IMG\_0547.JPG | 19,6504562 | 18,0348985 | 14,33942868 | Babi |
| IMG\_0548.JPG | 16,26504915 | 13,06716159 | 11,69666398 | Babi |
| IMG\_0549.JPG | 15,18758714 | 13,06947345 | 12,17646315 | Babi |
| IMG\_0552.JPG | 12,29850053 | 14,13634091 | 9,22749172 | Babi |
| IMG\_0554.JPG | 12,26395793 | 10,36188843 | 7,934520833 | Babi |
| IMG\_0555.JPG | 17,43384728 | 13,27269946 | 14,0693558 | Babi |
| IMG\_0557.JPG | 22,53324843 | 20,53935162 | 20,82707229 | Babi |
| IMG\_0500.JPG | 15,77448777 | 16,67887795 | 10,42115558 | Sapi |
| IMG\_0502.JPG | 13,22927121 | 15,05572288 | 9,614431182 | Sapi |
| IMG\_0504.JPG | 11,6752631 | 12,73324784 | 8,014403534 | Sapi |
| IMG\_0505.JPG | 15,08495938 | 16,04615778 | 12,93246487 | Sapi |
| IMG\_0506.JPG | 24,66031248 | 24,82597519 | 21,79271794 | Sapi |
| IMG\_0507.JPG | 22,20006412 | 28,45313925 | 17,92665286 | Sapi |
| IMG\_0510.JPG | 14,13919103 | 14,01330055 | 9,725459284 | Sapi |
| IMG\_0513.JPG | 14,88372757 | 16,05704058 | 10,89697811 | Sapi |
| IMG\_0514.JPG | 19,53857806 | 22,89281029 | 17,56396146 | Sapi |
| IMG\_0528.JPG | 33,34589954 | 31,21936197 | 33,34854376 | Babi |
| IMG\_0529.JPG | 24,8575559 | 19,99230851 | 22,42111146 | Babi |
| IMG\_0530.JPG | 17,80835681 | 14,80925097 | 12,07782673 | Babi |
| IMG\_0532.JPG | 21,87006818 | 12,40545199 | 18,55851123 | Babi |
| IMG\_0533.JPG | 17,94834493 | 17,69023725 | 15,21386763 | Babi |
| IMG\_0534.JPG | 22,04835199 | 21,86600625 | 21,32688699 | Babi |
| IMG\_0535.JPG | 19,19156671 | 13,82142206 | 13,84411123 | Babi |
| IMG\_0536.JPG | 14,80450077 | 14,1537828 | 11,07007936 | Babi |
| IMG\_0558.JPG | 20,44058579 | 21,75348103 | 21,93839746 | Sapi |
| IMG\_0559.JPG | 21,00155595 | 21,09637162 | 21,2252681 | Sapi |
| IMG\_0561.JPG | 14,05658468 | 14,81903372 | 13,86869165 | Sapi |
| IMG\_0562.JPG | 15,96299259 | 18,22947415 | 14,67822499 | Sapi |
| IMG\_0563.JPG | 16,45632761 | 18,22622402 | 15,7817116 | Sapi |
| IMG\_0564.JPG | 19,37945092 | 19,82301918 | 18,80953922 | Sapi |
| IMG\_0565.JPG | 22,50679146 | 23,15415047 | 24,70889684 | Sapi |
| IMG\_0566.JPG | 14,46748396 | 15,69965822 | 14,48514939 | Sapi |
| IMG\_0567.JPG | 19,57212944 | 21,37085056 | 19,35283218 | Sapi |
| IMG\_0568.JPG | 21,82416335 | 22,28760302 | 23,15200332 | Sapi |
| IMG\_0569.JPG | 19,07970328 | 21,39346032 | 17,40659321 | Sapi |
| IMG\_0570.JPG | 18,7022848 | 20,28158218 | 18,50204714 | Sapi |
| IMG\_0571.JPG | 22,24793962 | 19,86017954 | 23,95019207 | Sapi |
| IMG\_0572.JPG | 25,63479673 | 26,46946345 | 26,27344655 | Sapi |
| IMG\_0573.JPG | 16,85241451 | 16,68277223 | 17,09711576 | Sapi |
| IMG\_0575.JPG | 16,08324755 | 16,68662533 | 13,80308601 | Sapi |
| IMG\_0576.JPG | 19,12196679 | 19,7401893 | 19,36297671 | Sapi |
| IMG\_0577.JPG | 22,67929562 | 23,86804751 | 21,96744866 | Sapi |
| IMG\_0579.JPG | 22,34360238 | 24,59619808 | 23,44392416 | Sapi |
| IMG\_0580.JPG | 21,11654598 | 20,49317031 | 23,32109557 | Sapi |
| IMG\_0581.JPG | 17,65465049 | 17,3978958 | 17,73716271 | Sapi |

4.1.3 Analisis Data

Pada tahapan ini, penulis mengimplementasikan metode yang telah ditentukan yaitu K-NN terhadap data yang diperoleh. Parameter yang digunakan pada penelitian ini adalah nilai hasil ekstraksi fitur warna RGB ke HSV dari data yang diperoleh. Adapun tahap pengolahan data menggunakan metode KNN yaitu sebagai berikut :

1. Menentukan Jumlah Data Latih dan Data Uji

Pada tahap ini, ditentukan jumlah data yang akan digunakan sebagai data latih dan data uji, yaitu sebanyak 60 buah sebagai uji yang terdiri dari 30 buah data daging sapi dan 30 buah data daging babi hutan dan 20 buah sebagai data uji yang terdiri dari 10 buah data daging sapi dan 10 buah data daging babi hutan.

1. Menentukan Nilai N Secara Acak

Tahap selanjutnya yaitu menentukan nilai N secara acak. Nilai N tidak boleh melebihi dari jumlah data latih yang ada. Pada tahap ini penulis melakukan beberapa percobaan untuk menentukan nilai N, yaitu 3, 5, 7, 9 dan 11 untuk mencari hasil akurasi terbaik dari nilai N tersebut.

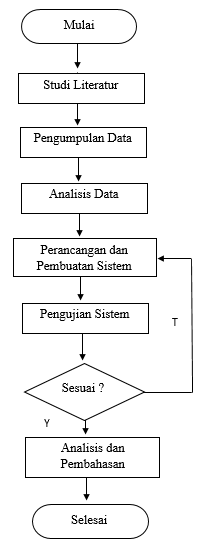
1. Menghitung *Euclidean Distance* dari Data Uji Terhadap Data Latih Berdasarkan Nilai N yang Telah Ditentukan.

**4.2 Desain**

Tahap Desain merupakan tahapan yang dilalui sebelum melakukan proses *coding*. Tahap desain bertujuan untuk memberikan gambaran atau rancangan dari sistem yang akan dibuat secara lengkap seperti misalnya tampilan dan alur dari sistem yang dibuat. Sehingga membantu merincikan spesifikasi kebutuhan *hardware* dan mendefinisikan rancangan dari sistem secara keseluruhan. Untuk melakukan tahap desain, penulis menggunakan *flowchart sistem* dan *Figma*.

4.2.1 *Flowchart Sistem*

Sistem yang dibuat dapat digunakan oleh siapa saja berdasarkan spesifikasi kebutuhan yang telah ditentukan. *Flowchart Sistem* dari sistem yang dibuat disajikan pada gambar 4.xx berikut :

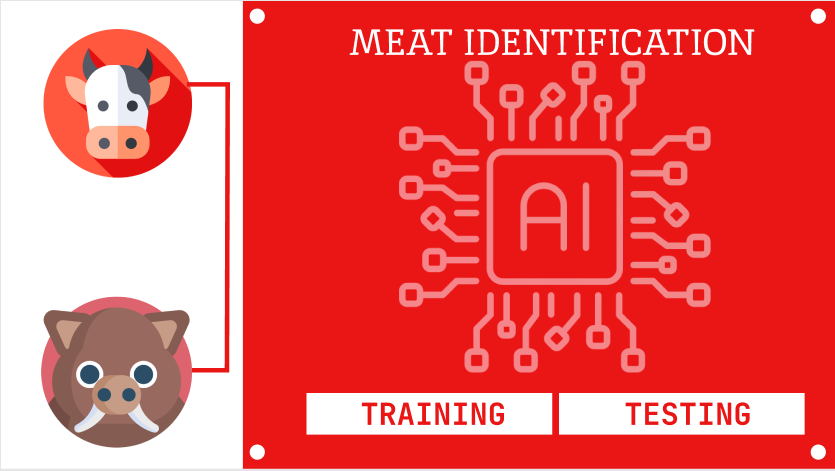


4.2.2 *Figma*

*Figma* digunakan untuk membuat desain yang nantinya akan diimplementasikan pada sistem yang akan dibuat. Berikut ini desain-desain yang telah dibuat yaitu :

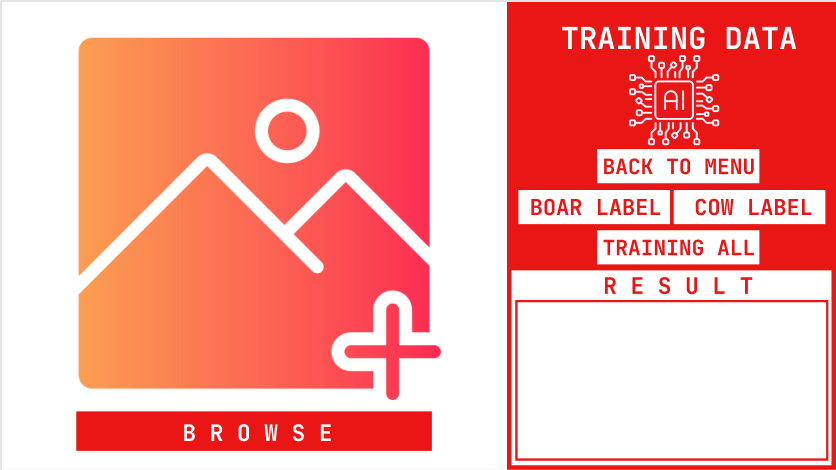
1. Desain Menu Utama

Berikut ini adalah desain menu utama menggunakan *Figma.*



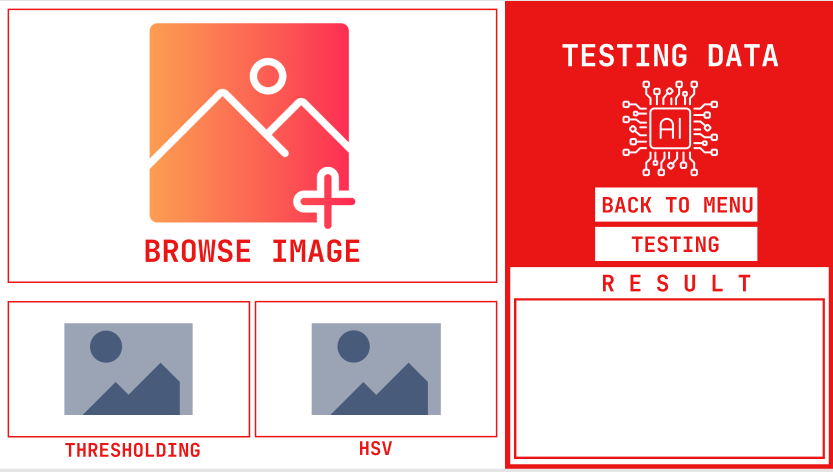
1. Desain *Training*

Berikut ini adalah desain *training* menggunakan *Figma.*



1. Desain *Testing*

Berikut ini adalah desain *testing* menggunakan *Figma.*

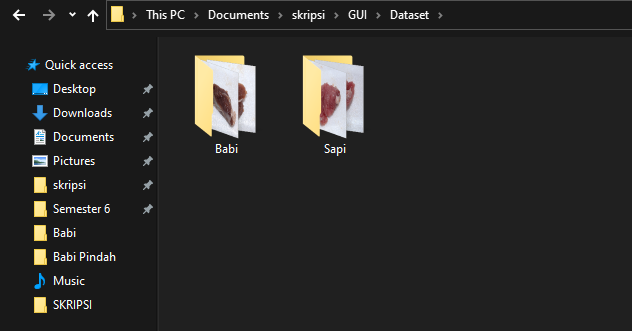


**4.3 *Implementation***

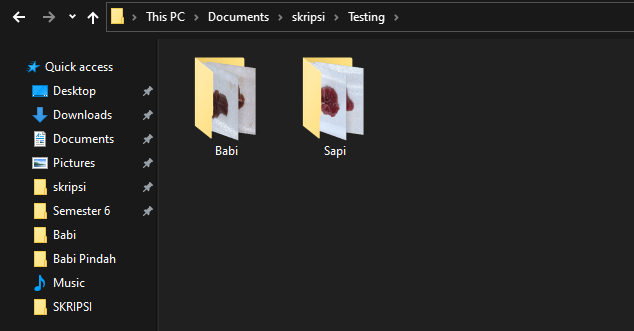
4.3.1 Membuat *Folder*

Hal pertama yang dilakukan pada tahapan implementasi yaitu membuat *folder. Folder* tersebut digunakan untuk tempat menyimpan hal-hal yang diperlukan pada saat proses penelitian. Terdapat beberapa *folder* yang dibuat, diantaranya yaitu :

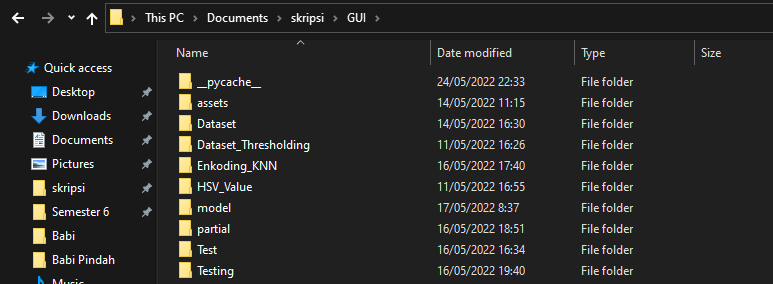
1. *Folder Dataset, folder* tersebut digunakan untuk menyimpan citra daging sapi dan daging babi hutan yang disimpan dalam dua *folder* yang berbeda dalam satu *folder Dataset.* Masing-masing jenis citra disimpan dalam folder dengan nama *folder* sesuai dengan jenis citranya, yaitu Sapi dan Babi. Citra yang terdapat di dalam *folder* tersebut nantinya akan digunakan sebagai data latih pada saat penelitian.



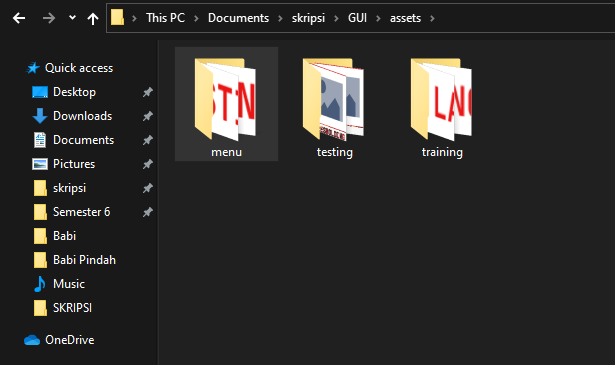
1. *Folder Testing, folder* tersebut digunakan untuk menyimpan data citra daging sapi dan daging babi hutan yang nantinya akan digunakan pada saat pengujian. Citra daging disimpan berdasarkan masing-masing jenis daging, yaitu Sapi dan Babi.



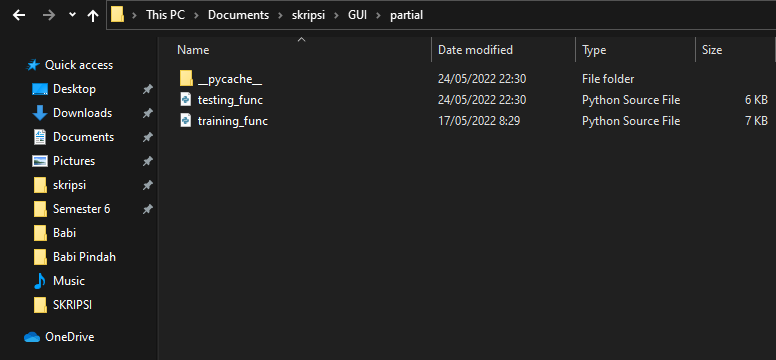
1. *Folder GUI, folder* tersebut digunakan untuk menyimpan segala hal yang diperlukan pada saat membuat *GUI* dari sistem yang akan dibuat.



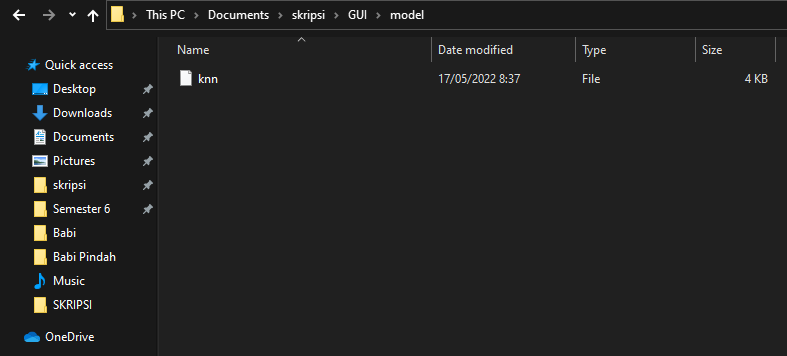
1. *Folder Assets, folder* tersebut berisi *folder* yang menujukkan bagian *GUI* yang dibuat. Masing-masing *folder* berisi gambar yang digunakan pada saat pembuatan antarmuka dari sistem yang dibuat.



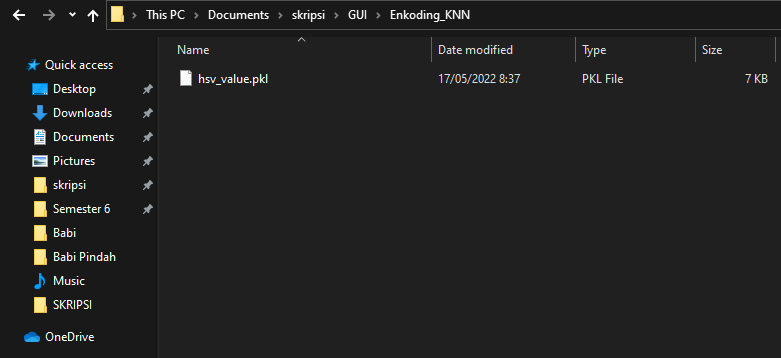
1. *Folder Partial, folder* tersebut digunakan untuk menyimpan file fungsi berekstensi .py yang berisi fungsi-fungsi yang akan digunakan/dipanggil pada fitur *GUI.*



1. *Folder* Model, *folder* tersebut digunakan untuk menyimpan *file* berekstensi .py yang berisi algoritma yang digunakan yaitu K-NN.



1. *Folder Enkoding\_KNN, folder* tersebut digunakan untuk menyimpan *file* berekstensi .pkl yang berisi data nilai HSV dari masing-masing citra daging beserta *label* yang sesuai. *File* tersebut nantinya akan digunakan sebagai pembanding pada saat perhitungan *Euclidean Distance* dan penentuan identifikasi sebagai hasil dari metode K-NN.



4.3.2 Pemasangan *Library*

Pada penelitian ini, Bahasa pemrograman yang digunakan adalah *Python*. Adapun sebelum dapat melanjutkan proses pembuatan sistem, terlebih dahulu dilakukan instalasi *library* untuk memudahkan pada saat proses pembuatan sistem.

**Kode Program 4.1 Instalasi Package**

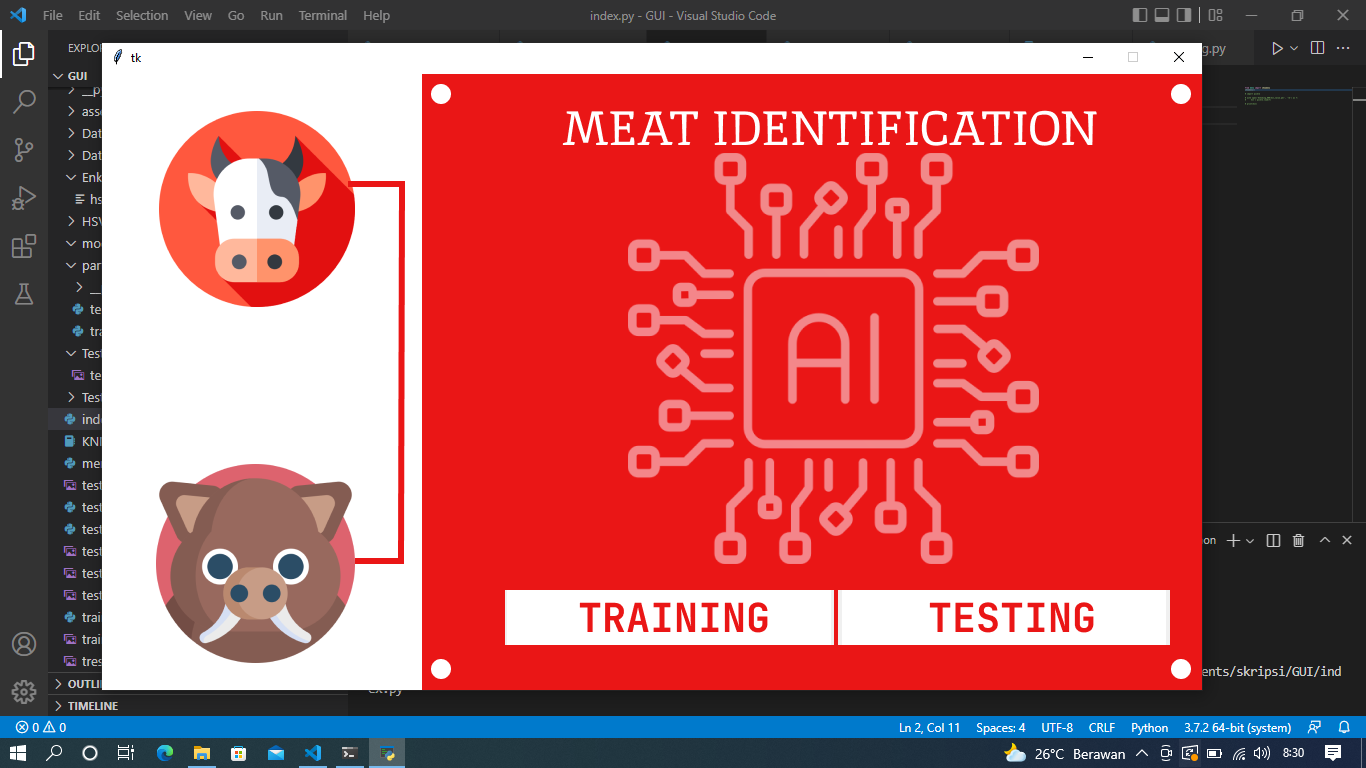
1. pip install numpy
2. pip install PIL
3. pip install pickle
4. pip install os
5. pip install pandas
6. pip install cv2
7. pip install sckit-learn
8. pip install sys

Kode dasar pemasangan *library* dari *python* yaitu pip install (nama *library*). Kode di atas adalah *library* yang digunakan pada saat pembuatan sistem ini.

4.3.3 Implementasi Sistem

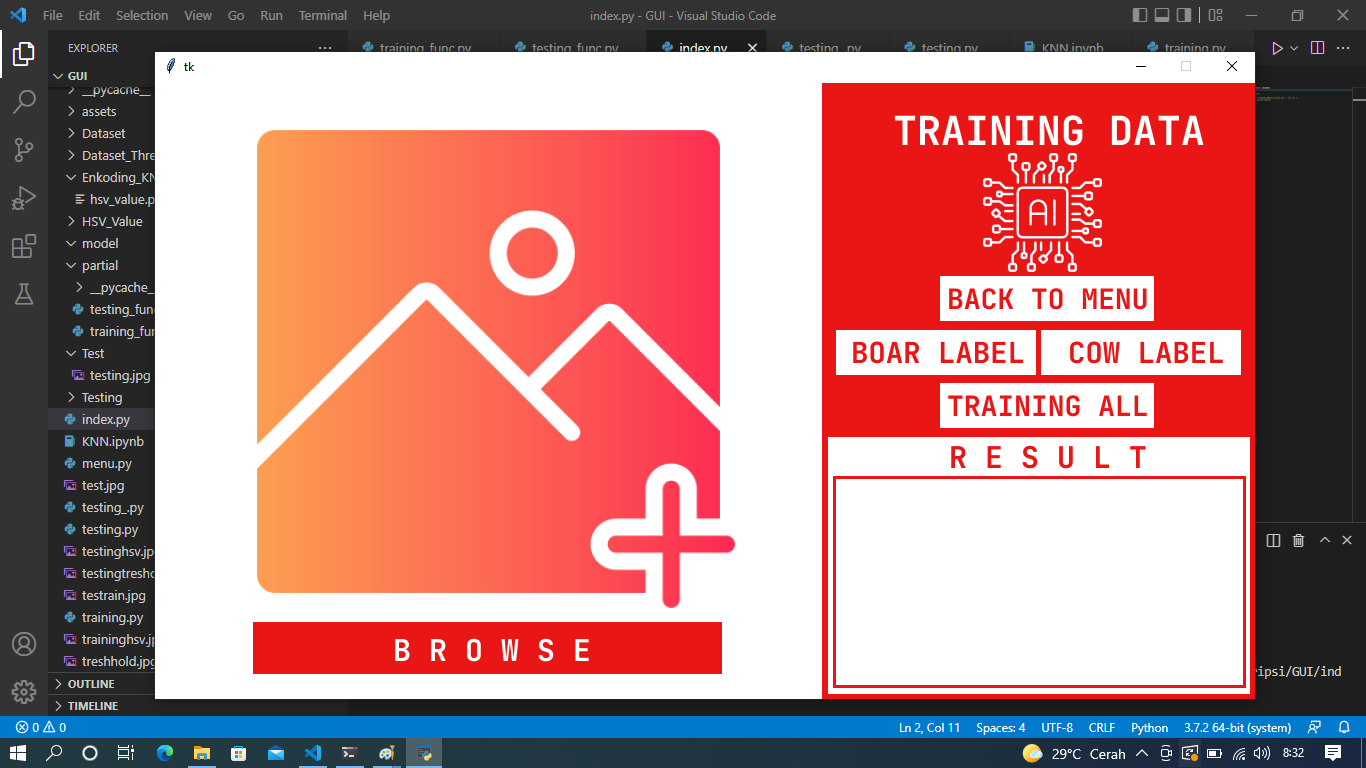
Berikut ini adalah hasil implementasi desain yang telah dibuat ke dalam bentuk sistem :

1. Antarmuka Sistem Menu Utama



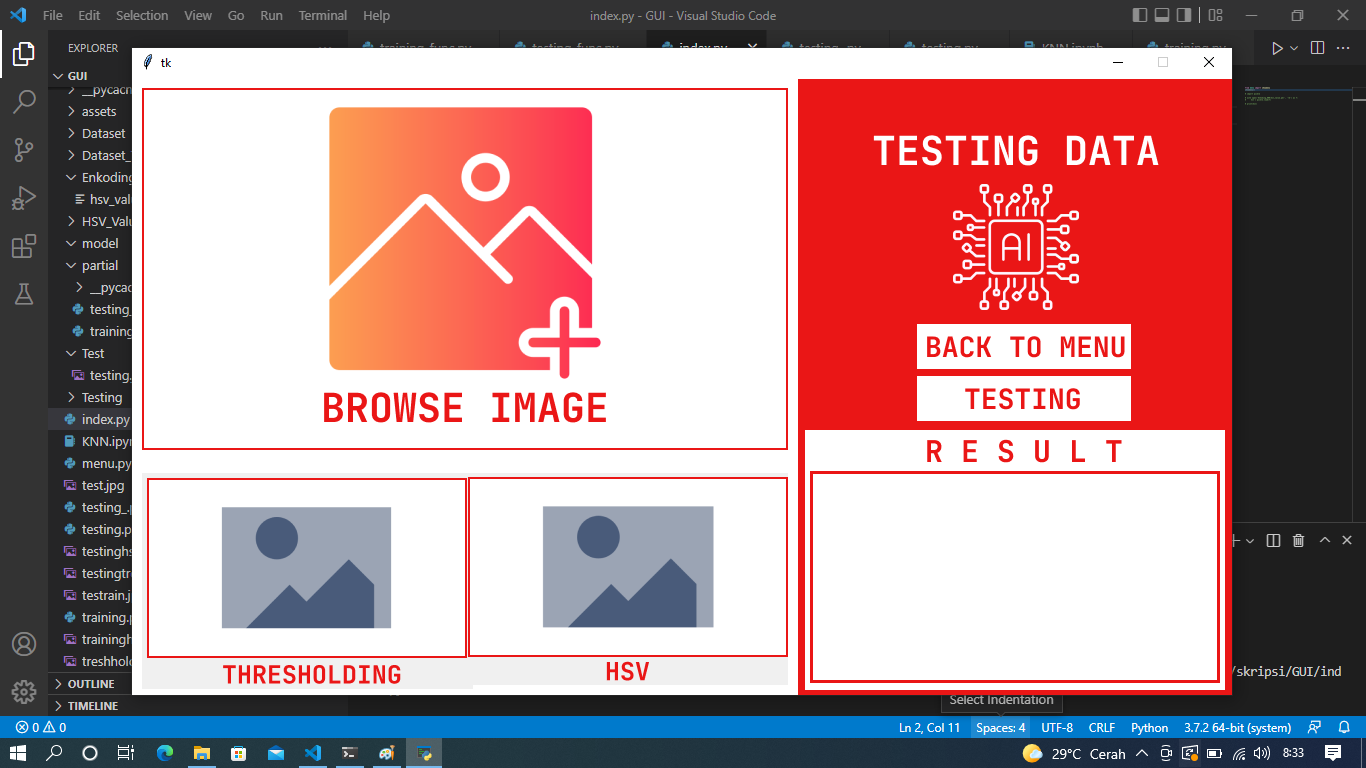
Pada gambar 4.xx tersebut adalah tampilan menu utama, pada menu tersebut terdapat dua tombol, yaitu tombol *TRAINING* untuk melakukan proses *training data* dan tombol *TESTING* untuk melakukan proses uji data.

1. Antarmuka Sistem Menu *Training*



Pada gambar 4.xx tersebut adalah tampilan *TRAINING*, pada menu tersebut terdapat lima tombol, yaitu tombol *BROWSE* untuk mencari dan mengunggah gambar yang akan disimpan baik itu sebagai data latih maupun data uji, tombol *BACK TO MENU* untuk kembali ke menu utama, tombol *BOAR LABEL* untuk melakukan menyimpan gambar menjadi data latih daging babi hutan, tombol *COW LABEL* untuk menyimpan gambar menjadi data latih daging sapi, dan tombol *TRAINING ALL* untuk melakukan proses *training data* yang akan menghasilkan nilai HSV untuk digunakan pada saat pengujian sistem. Selain itu terdapat satu label *RESULT* yang nantinya akan menampilkan jalannya proses *training* sampai proses *training* selesai dilakukan.

1. Antarmuka Sistem Menu *Testing*

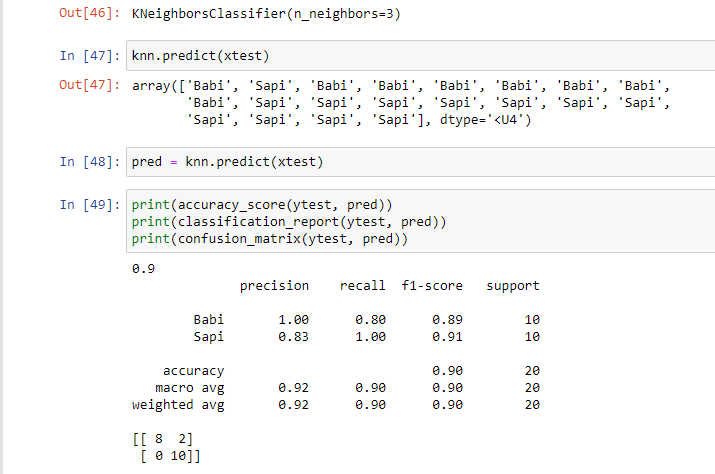


Pada gambar 4.xx tersebut adalah tampilan *TESTING*, pada menu tersebut terdapat tiga tombol, yaitu tombol *BROWSE IMAGE* untuk mencari dan mengunggah gambar yang akan diuji, tombol *BACK TO MENU* untuk kembali ke menu utama, dan tombol *TESTING* untuk melakukan proses *testing data* yang akan menghasilkan hasil identifikasi terhadap gambar yang diuji pada saat pengujian sistem. Selain itu terdapat satu label *RESULT* yang nantinya akan menampilkan hasil identifikasi setelah proses *testing* selesai dilakukan.

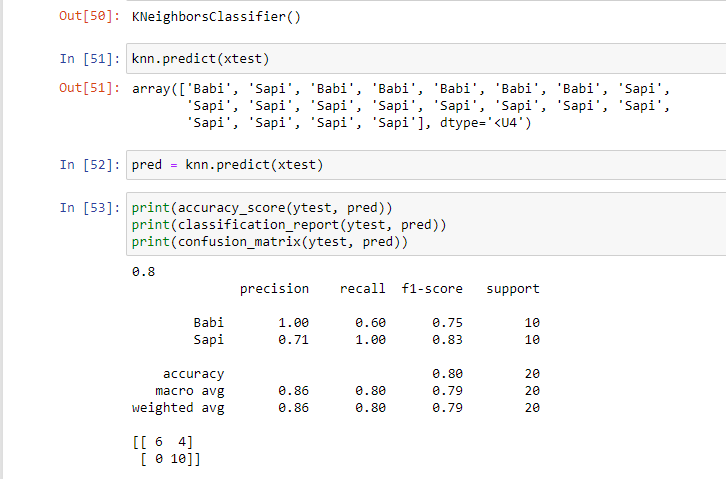
4.4 *Training*

Proses *training* dilakukan untuk mendapatkan model dari metode yang akan digunakan yaitu K-NN. Proses tersebut juga dilakukan untuk menentukan nilai N yang terbaik untuk diimplementasikan. Pada tahap ini data yang digunakan untuk proses *training* yaitu dari *dataset* sebanyak 60 buah, kemudian dilakukan *split* membagi antara nilai dari citra dan label dari citra. Hasil dari proses ini adalah *file* berekstensi .pkl dan model K-NN yang disimpan pada *folder* yang telah dibuat sebelumnya. *File* tersebut nantinya akan digunakan untuk proses *testing*.

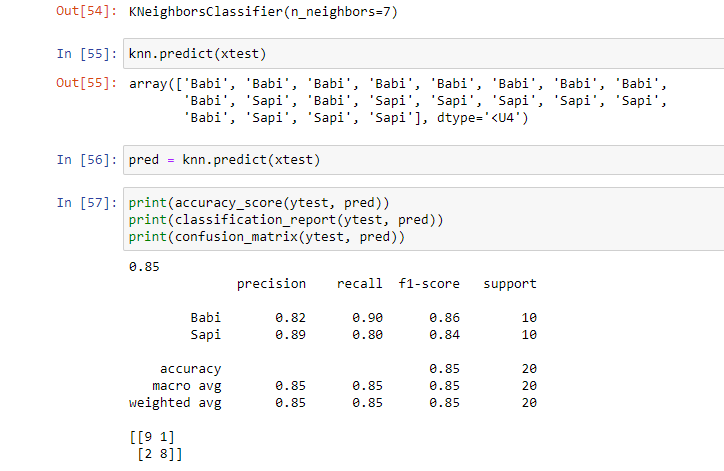
Untuk mendapatkan model yang sesuai saat implementasi metode K-NN, terlebih dahulu harus menentukan nilai N yang sesuai untuk menghasilkan akurasi yang terbaik. Pada penelitian kali ini dilakukan percobaan menggunakan 60 data latih dan 20 data uji, kemudian ditentukan nilai N secara acak yaitu 3, 5, 7, 9 dan 11.



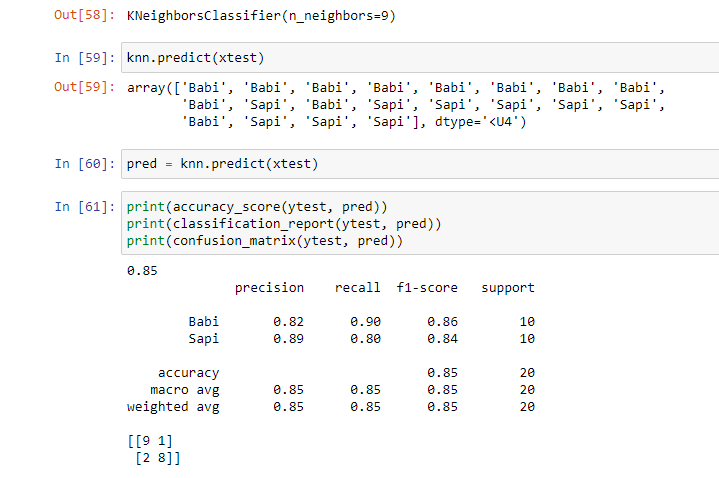
Pada saat percobaan menggunakan nilai N = 3, menghasilkan akurasi sebesar 90% atau terdapat dua kesalahan dari 20 data uji yang ada.



Pada saat percobaan menggunakan nilai N = 5, menghasilkan akurasi sebesar 80% atau terdapat empat kesalahan dari 20 data uji yang ada.



Pada saat percobaan menggunakan nilai N = 7, menghasilkan akurasi sebesar 85% atau terdapat tiga kesalahan dari 20 data uji yang ada.



Pada saat percobaan menggunakan nilai N = 9, menghasilkan akurasi sebesar 85% atau terdapat tiga kesalahan dari 20 data uji yang ada.

**Kode Program 4.XX Proses *Training Data***

1. def training\_all(result):
2. path = 'Dataset'
3. data\_hsv = {}
4. imgname = []
5. for filename in os.listdir(path):
6. for imagename in os.listdir(os.path.join(path, filename)):
7. if len(imgname) > 8 :

imgname=[]

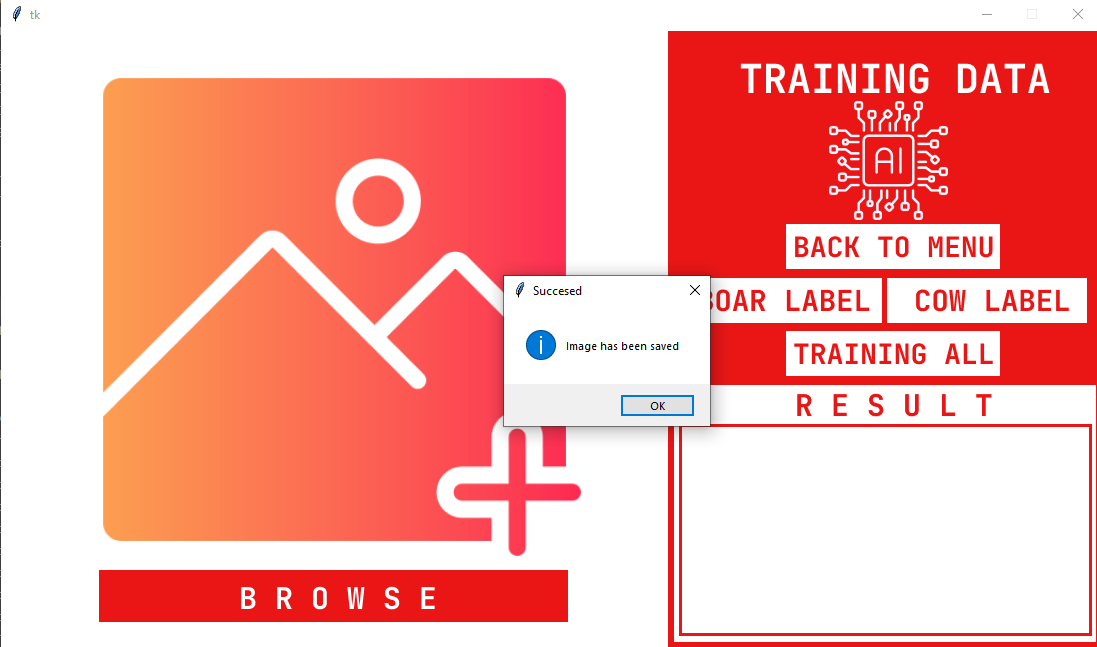
1. imgname.append(imagename + " Succes To Trained")
2. image = imread(os.path.join(path, filename, imagename))
3. gray = rgb2gray(image)
4. thresh = threshold\_otsu(gray)
5. masking  = gray < thresh
6. filtered = masked\_image(image, masking)
7. image = Image.fromarray(filtered)
8. image = image.resize((778, 437))
9. image = np.asarray(image)
10. hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)
11. h,s,v = cv2.split(hsv)
12. hmean = h.mean()
13. smean = s.mean()
14. vmean = v.mean()
15. data\_hsv[f'{imagename}'] = (hmean, smean, vmean, filename)
16. # result
17. result.configure(text = '\n'.join(x for x in imgname))
18. result.update()
19. file = open('Enkoding\_KNN/hsv\_value.pkl', 'wb')
20. pickle.dump(data\_hsv, file)
21. file.close()
22. # create model knn
23. xtrain, ytrain = [], []
24. for key, data in data\_hsv.items():
25. xtrain.append([data[0], data[1], data[2]])
26. ytrain.append(data[3])
28. xtrain = np.array(xtrain)
29. ytrain = np.array(ytrain)
30. knn = KNeighborsClassifier(n\_neighbors=3)
31. knn.fit(xtrain, ytrain)
32. with open('model/knn', 'wb') as f:
33. pickle.dump(knn, f)

4.4 *Testing*

Proses yang dilakukan setelah sistem selesai dibuat adalah *testing* atau pengujian sistem untuk mengetahui hasil dari sistem yang telah dibuat. Adapun hal-hal yang dilakukan dalam proses *testing* yaitu :

4.4.1 Fitur Menyimpan Citra pada *Folder Dataset*

Proses ini dilakukan ada menu *Training.* Pada menu tersebut dilakukan proses memasukkan citra yang mungkin akan dijadikan sebagai tambahan *dataset.* Cara menggunakannya yaitu dengan meng-klik tombol *COW LABEL* untuk menyimpan citra sebagai *dataset* daging sapi atautombol *BOAR LABEL* untuk menyimpan citra sebagai *dataset* daging babi hutan.



**Kode Program 4.XX Proses Menyimpan Citra Daging Sapi**

1. def savecow():
2. if label\_original != "":
3. value = randint(0, 1000)
4. convert = cv2.cvtColor(np.array(image\_original), cv2.COLOR\_RGB2BGR)
5. # print(image\_original)
6. save\_babi =cv2.imwrite("../Dataset/Sapi/"+str(value)+".jpg", convert)
7. label\_original.configure(image="")
8. label\_original.destroy()
9. messagebox.showinfo(title= "Succesed", message= "Image has been saved")
10. else:
11. messagebox.showwarning(title= "Warning", message= "Image must be choosen")

Kode di atas adalah fungsi yang dipanggil untuk menjalankan perintah simpan citra daging sapi pada fitur *Training*. Sebelumnya terlebih dahulu mengunggah citra dengan meng-klik tombol *BROWSE.* Setelah mengunggah citra, maka citra akan ditampilkan pada variable *label\_original.* Kemudian citra akan disimpan menggunakan *cv2* pada *folder dataset* yang telah dibuat sebelumnya. Kemudian setelah citra berhasil disimpan, maka akan muncul pemberitahuan gambar berhasil disimpan dan gambar yang ditampilkan pada original akan di hapus. Namun jika meng-klik tombol sebelum citra diunggah, maka akan muncul pemberitahuan gambar harus dipilih.

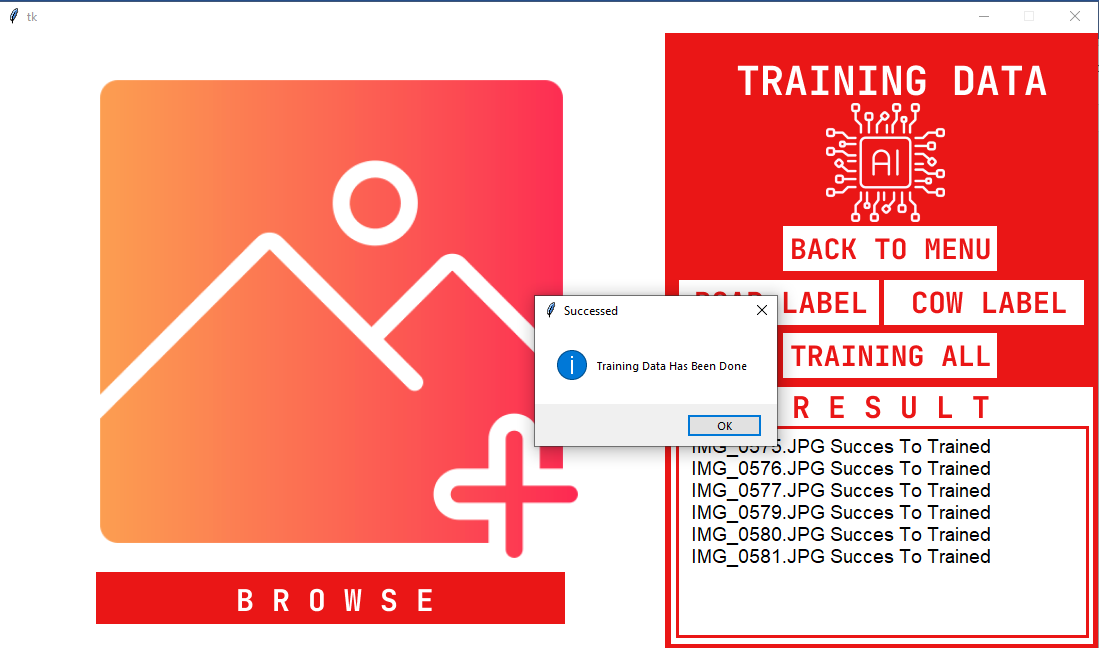
**Kode Program 4.XX Proses Menyimpan Citra Daging Babi**

1. def savecow():
2. if label\_original != "":
3. value = randint(0, 1000)
4. convert = cv2.cvtColor(np.array(image\_original), cv2.COLOR\_RGB2BGR)
5. # print(image\_original)
6. save\_babi =cv2.imwrite("../Dataset/Babi/"+str(value)+".jpg", convert)
7. label\_original.configure(image="")
8. label\_original.destroy()
9. messagebox.showinfo(title= "Succesed", message= "Image has been saved")
10. else:
11. messagebox.showwarning(title= "Warning", message= "Image must be choosen")

Kode di atas adalah fungsi yang dipanggil untuk menjalankan perintah simpan citra daging sapi pada fitur *Training*. Sebelumnya terlebih dahulu mengunggah citra dengan meng-klik tombol *BROWSE.* Setelah mengunggah citra, maka citra akan ditampilkan pada variable *label\_original.* Kemudian citra akan disimpan menggunakan *cv2* pada *folder dataset* yang telah dibuat sebelumnya. Kemudian setelah citra berhasil disimpan, maka akan muncul pemberitahuan gambar berhasil disimpan dan gambar yang ditampilkan pada original akan di hapus. Namun jika meng-klik tombol sebelum citra diunggah, maka akan muncul pemberitahuan gambar harus dipilih.

4.4.2 Fitur *Training Data*

Proses ini dilakukan ada menu *Training.* Pada menu tersebut dilakukan proses *training* terhadap citra yang terdapat pada *folder dataset.* Cara menggunakannya yaitu dengan meng-klik tombol *TRAINING ALL* untuk melakukan.*training* terhadap semua *dataset.* Proses tersebuut nantinya menghasilkan *file* berekstensi .pkl yang akan menyimpan nilai HSV dari masing-masing citra yang akan digunakan pada saat menghitung *Euclidean Distance.*



**Kode Program 4.XX Proses *Training Data***

1. def training\_all(result):
2. path = 'Dataset'
3. data\_hsv = {}
4. imgname = []
5. for filename in os.listdir(path):
6. for imagename in os.listdir(os.path.join(path, filename)):
7. if len(imgname) > 8 :

imgname=[]

1. imgname.append(imagename + " Succes To Trained")
2. image = imread(os.path.join(path, filename, imagename))
3. gray = rgb2gray(image)
4. thresh = threshold\_otsu(gray)
5. masking  = gray < thresh
6. filtered = masked\_image(image, masking)
7. image = Image.fromarray(filtered)
8. image = image.resize((778, 437))
9. image = np.asarray(image)
10. hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)
11. h,s,v = cv2.split(hsv)
12. hmean = h.mean()
13. smean = s.mean()
14. vmean = v.mean()
15. data\_hsv[f'{imagename}'] = (hmean, smean, vmean, filename)
16. # result
17. result.configure(text = '\n'.join(x for x in imgname))
18. result.update()
19. file = open('Enkoding\_KNN/hsv\_value.pkl', 'wb')
20. pickle.dump(data\_hsv, file)
21. file.close()

Pada kode di atas *path* diarahkan pada *folder dataset,* kemudian citra-citra yang terdapat di dalamnya akan melalui proses segmentasi citra, yaitu memisahkan objek dengan latar belakang, atau biasa dikenal dengan proses *thresholding.* Setelah itu citra akan melalui proses ekstraksi fitur warna RGB ke HSV. Kemudian nilai HSV dari citra akan diambil nilai rata-rata untuk mendapatkan masing-masing satu nilai HSV terhadap citra yang ada. Setelah itu nilai HSV dari citra akan disimpan dalam *array data\_hsv.* Terakhir nilai yang ditampung *array data\_hsv* akan disimpan dalam *file* berekstensi .pkl pada *folder* yang telah disiapkan sebelumnya

4.4.3 *Input* Citra dan Kelas Citra

Objek yang dijadikan bahan masukan adalah citra daging sapi dan daging babi yang telah diperoleh dari narasumber. Adapun banyaknya daging yang diperoleh dipotong-potong sehingga mendapatkan dua kelas yaitu kelas daging babi dan daging sapi, kemudian dibagi menjadi 2 bagian yaitu sebagai data uji dan data latih.

Adapun banyaknya data yang digunakan sebagai data uji yaitu 20 data yang terdiri dari 10 buah daging babi dan 10 buah daging sapi. Sedangkan data latih yang digunakan yaitu sebanyak 60 buah yang terdiri dari 30 daging sapi dan 30 daging babi.

Berikut contoh data yang digunakan sebagai data uji, ditampikan pada gambar 4.xx



**Kode Program 4.XX Proses Mengunggah Gambar**

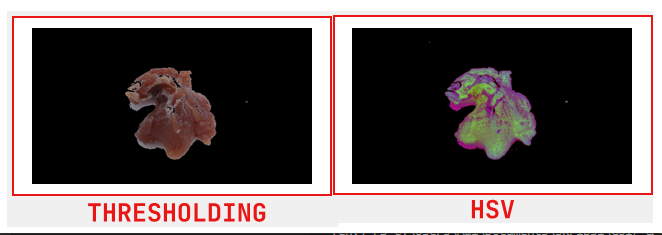
1. def uploadimage():
2. global image\_original, label\_original, filepath
3. filename =filedialog.askopenfilename(initialdir="C:/Users/Adi/Documents/skripsi")
4. if filename !="":
5. img = Image.open(filename)
6. image\_original = img
7. img = img.resize((516, 290))
8. load = ImageTk.PhotoImage(img)
9. label1 = Label(image = load)
10. label1.image = load
11. label1.place(
12. x = 75, y = 20,
13. width= 516,
14. height= 290
15. )
16. label\_original = label1

Kode diatas adalah fungsi yang dijalankan ketika mengunggah citra. Pertama dibuat variabel global agar dapat diakses secara global. Kemudian menggunakan *askopenfilename* akan diarahkan pada *folder* untuk memilih citra yang diinginkan dan akan disimpan pada variabel *filename.* Setelah gambar dipilih, maka gambar asli akan ditampung pada variabel *image\_original* untuk menampung citra sesuai dengan keadaan yang sebenarnya. Kemudian citra juga akan ditampilkan pada *label* agar dapat melihat citra apa yang diunggah. Namun sebelum ditampilkan pada label, citra akan di *resize* ukutannya sesuai dengan kebutuhan.

4.4.4 Pre-processing

Tahapan yang dilakukan sebelum citra dilakukan proses identifikasi yaitu pre-processing. Pada tahap ini data-data yang ada akan dibuat seragam. Adapun yang dilakukan adalah proses *resize* dari ukuran citra, yaitu dari ukuran 5184 X 2912 pixel menjadi 778 X 437 pixel. Proses ini dilakukan agar pada saat proses perhitungan identifikasi, tidak memerlukan waktu yang lama dan memori yang besar.

Namun sebelum itu, citra yang ada akan melalui proses *thresholding\_otsu. Thresholding* dilakukan untuk mendapatkan gambar yang hanya berisi objek saja, karena bagian yang digunakan pada proses penelitian dari data yang ada adalah objeknya saja, sedangkan *background* tidak digunakan. Setelah gambar melalui proses *thresholding,* selanjutnya citra akan dilakukan proses ekstraksi fitur warna RGB ke HSV yang menghasilkan nilai HSV untuk dihitung *Euclidean Distancenya.* Selengkapnya pada gambar 4.xx di bawah ini .



**Kode Program 4.2 Resize dan Thresholding**

1. def masked\_image(image, mask):
2. r = image[:,:,0] \* mask
3. g = image[:,:,1] \* mask
4. b = image[:,:,2] \* mask
5. return np.dstack([r,g,b])
6. def testing(result):
7. global label\_original, image\_original, filepath
8. if label\_original != "":
9. image = imread(filepath)
10. gray = rgb2gray(image)
11. thresh = threshold\_otsu(gray)
12. masking  = gray < thresh
13. filtered = masked\_image(image, masking)
14. image = Image.fromarray(filtered)
15. image = image.resize((778, 437))
16. image = np.asarray(image)

Kode diatas adalah dua fungsi yang dijalankan ketika melakukan proses *thresholding\_otsu* dan *resize.* Citra yang telah diunggah akan dilakukan segmentasi citra, yang menghasilkan citra berisi objek saja, sedangkan latar belakang citra akan menjadi hitam. Selain itu citra juga akan di *resize* sesuai dengan kebutuhan.

**Kode Program 4.2 Ekstraksi Fitur Warna RGB ke HSV**

1. hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

Kode diatas adalah perintah untuk mengubah warna RGB menjadi HSV sebelum diambil nilai masing-masingnya, karena pada penelitian ini, nilai yang digunakan adalah HSV dari citra tersebut.

4.4.5 Identifikasi K-NN

Metode yang digunakan pada penelitian ini adalah K-NN. Metode tersebut melakukan proses identifikasi berdasarkan k terdekat yang telah ditentukan. Proses penentuan nilai k itu sendiri secara default adalah 5. Proses menentukan letak k terdekat tersebut, dilakukan perhitungan jarak terdekat atau yang dikenal sebagai *Euclidean Distance.*

Sebelum citra dapat dilakukan proses Setelah citra melalui proses ekstraksi fitur perhitungan *Euclidean Distance,* terlebih dahulu harus mendapat nilai HSV dari citra tersebut, karena parameter perhitungan yang digunakan adalah nilai rata-rata HSV dari citra.

Untuk mendapat nilai rata-rata HSV dari citra, terlebih dahulu akan diambil nilai HSV dari masing-masing pixel pada citra, kemudian seluruh nilai yang didapat akan dihitung rata-ratanya, sehingga mendapat masing-masing nilai HSV yang diinginkan.

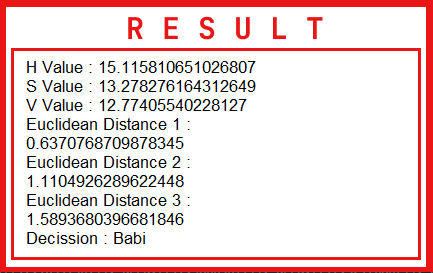
**Kode Program 4.4 Mendapatkan Nilai HSV Citra**

1. hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)
2. h,s,v = cv2.split(hsv)
3. hmean = h.mean()
4. smean = s.mean()
5. vmean = v.mean()

Proses tersebut menghasilkan nilai ratarata masing-masing HSV dari citra yang telah diunggah.

Setelah mendapat nilai masing-masing HSV, maka proses identifikasi dapat dilakukan, yaitu dengan menghitung *Euclidean Distance* antara data uji dangan data latih, kemudian menentukan hasil identifikasi berdasarkan k terdekat atau selisih terdekat dari perhitungan tersebut.

Berikut contoh hasil ekstraksi fitur warna RGB ke HSV, beserta hasil perhitungan *Euclidean Distance* dan hasil identifikasi terhadap data yang diujipada gambar 4.xx berikut.



**Kode Program 4.xx Identifikasi**

1. decission = knn.predict(xtest)
2. ed = euclidean\_distances(xtrain, xtest)
3. ed\_sort = sorted(ed, key=lambda x:x[0])
4. print(ed\_sort)
5. # result
6. result.configure(text = f'H Value : {xtest[0][0]}\nS Value : {xtest[1][0]}\nV Value : {xtest[2][0]}\nEuclidean Distance 1 : {ed\_sort[0][0]}\nEuclidean Distance 2 : {ed\_sort[0][1]}\nEuclidean Distance 3 : {ed\_sort[0][2]} \nDecission : {decission[0]}')

Pada kode di atas setelah melakukan perhitungan *Euclidean Distance* terhadap seluruh data latih, maka nilai yang dihasilkan akan diambil 3 terkecil sebagai jarak terpendek. Selanjutnya nilai dan hasil identifikasi ditampilkan pada label *Result*.

4.4.5 Analisis dan Pembahasan

Setelah program selesai dibuat, maka proses selanjutnya adalah melakukan pengujian terhadap sistem yang mengimplementasikan metode K-NN. Adapun pengujian yang dilakukan yaitu dengan memasukkan data latih ke dalam program, kemudian data latih tersebut akan diidentifikasi apakah termasuk kelas daging bagi atau daging babi.

Data yang dimasukkan sebagai data uji, akan melalui tahapan *pre-processing* terlebih dahulu untuk mendapatkan data yang sesuai. Sebagai contoh diambil satu data random sebagai data uji.

Kemudian akan dilakukan ekstraksi fitur warna RGB ke HSV pada data uji dan akan diambil nilai mean dari nilai HSV masing-masing pixel. Berikut ini adalah perhitungan manual pada sistem tersebut :

1. Pada saat melakukan pengujian terhadap salah satu gambar, akan menghasilkan nilai HSV. Pada data yang dipilih didapat nilai H = 15,11581065, S = 13,27827616, V = 12,7740554 dan kelas termasuk kelas Babi.
2. Selanjutnya diambil nilai HSV dari data uji, kemudian akan dihitung nilai *Euclidean Distance* terhadap nilai HSV data latih seperti ditampikan pada tabel 4.xx . Berikut ini contoh perhitungan manual untuk menghitung nilai *Euclidean Distance.*

Euclidean Distance (ED) terhadap data latih pertama :

ED = 22,8508438582765-15,1158106510268)^2 + (22,5449194966852-

15,1158106510268)^2 + (21,8118157806498-12,7740554022813)^2

ED= (7,7350332072497)^2+(7,4291088456584)^2+(9,0377603783 685)^2

ED = 59,83073871725558 + 55,19165824063988+ 81,68111265680753

ED = 14,0251028379368

Euclidean Distance (ED) terhadap data latih kedua :

ED = 21,1832340155183-15,1158106510268)^2 + (18,3433935514992-

15,1158106510268)^2 + (18,5854564599719-12,7740554022813)^2

ED= (6,0674233644915)^2+(3,2275829004724)^2+(5,8114010576906)^2

ED = 36,81362628397735 + 10,41729137942183+ 33,77238225332742

ED = 9,000183326839882

Proses perhitungan tersebut dilakukan pada semua data latih, sehingga mendapat nilai masing-masing *Euclidean Distance* seperti pada tabel 4.xx berikut ini*.*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Gambar | Nilai ED | Urutan |
| 1 | IMG\_0517.JPG | 14,02510284 | 49 |
| 2 | IMG\_0518.JPG | 9,000183327 | 37 |
| 3 | IMG\_0520.JPG | 1,045647037 | 2 |
| 4 | IMG\_0522.JPG | 9,364755082 | 39 |
| 5 | IMG\_0525.JPG | 4,210725501 | 19 |
| 6 | IMG\_0526.JPG | 5,053776855 | 26 |
| 7 | IMG\_0527.JPG | 10,58608403 | 43 |
| 8 | IMG\_0528.JPG | 31,85859503 | 60 |
| 9 | IMG\_0529.JPG | 14,55154708 | 53 |
| 10 | IMG\_0530.JPG | 2,797948899 | 12 |
| 11 | IMG\_0532.JPG | 9,296556787 | 38 |
| 12 | IMG\_0533.JPG | 4,539119567 | 23 |
| 13 | IMG\_0534.JPG | 12,91418593 | 47 |
| 14 | IMG\_0535.JPG | 4,408202441 | 20 |
| 15 | IMG\_0536.JPG | 1,981399956 | 7 |
| 16 | IMG\_0538.JPG | 1,518425338 | 3 |
| 17 | IMG\_0539.JPG | 5,996433756 | 28 |
| 18 | IMG\_0540.JPG | 4,512000653 | 21 |
| 19 | IMG\_0541.JPG | 7,258311081 | 31 |
| 20 | IMG\_0542.JPG | 14,37814901 | 52 |
| 21 | IMG\_0543.JPG | 9,509013468 | 40 |
| 22 | IMG\_0544.JPG | 3,826601969 | 18 |
| 23 | IMG\_0546.JPG | 1,638498978 | 5 |
| 24 | IMG\_0547.JPG | 5,615556746 | 27 |
| 25 | IMG\_0548.JPG | 2,584276375 | 11 |
| 26 | IMG\_0549.JPG | 2,133017654 | 10 |
| 27 | IMG\_0552.JPG | 4,634081491 | 24 |
| 28 | IMG\_0554.JPG | 7,358935744 | 32 |
| 29 | IMG\_0555.JPG | 3,232360709 | 14 |
| 30 | IMG\_0557.JPG | 12,21819384 | 45 |
| 31 | IMG\_0500.JPG | 2,900546937 | 13 |
| 32 | IMG\_0502.JPG | 3,680471007 | 16 |
| 33 | IMG\_0504.JPG | 6,337843396 | 30 |
| 34 | IMG\_0505.JPG | 0,944241042 | 1 |
| 35 | IMG\_0506.JPG | 16,33159773 | 58 |
| 36 | IMG\_0507.JPG | 15,9568243 | 55 |
| 37 | IMG\_0510.JPG | 3,385742565 | 15 |
| 38 | IMG\_0513.JPG | 2,11262763 | 9 |
| 39 | IMG\_0514.JPG | 10,14819171 | 42 |
| 40 | IMG\_0558.JPG | 12,50588116 | 46 |
| 41 | IMG\_0559.JPG | 11,90932841 | 44 |
| 42 | IMG\_0561.JPG | 1,551858471 | 4 |
| 43 | IMG\_0562.JPG | 3,746795893 | 17 |
| 44 | IMG\_0563.JPG | 4,529641582 | 22 |
| 45 | IMG\_0564.JPG | 8,761478492 | 35 |
| 46 | IMG\_0565.JPG | 16,1765863 | 57 |
| 47 | IMG\_0566.JPG | 1,920689491 | 6 |
| 48 | IMG\_0567.JPG | 10,11259637 | 41 |
| 49 | IMG\_0568.JPG | 14,28770117 | 51 |
| 50 | IMG\_0569.JPG | 8,751099171 | 34 |
| 51 | IMG\_0570.JPG | 8,506343617 | 33 |
| 52 | IMG\_0571.JPG | 14,08127588 | 50 |
| 53 | IMG\_0572.JPG | 20,53747943 | 59 |
| 54 | IMG\_0573.JPG | 4,915283557 | 25 |
| 55 | IMG\_0575.JPG | 2,112414944 | 8 |
| 56 | IMG\_0576.JPG | 8,99155431 | 36 |
| 57 | IMG\_0577.JPG | 14,77587339 | 54 |
| 58 | IMG\_0579.JPG | 15,99890054 | 56 |
| 59 | IMG\_0580.JPG | 13,27271177 | 48 |
| 60 | IMG\_0581.JPG | 6,023790703 | 29 |

Dengan menggunakan nilai N = 3, kemudian nilai tersebut akan dipilih berdasarkan 3 nilai terkecil atau jarak terpendek, kemudian akan ditentukan identifikasi berdasarkan label paling banyak muncul, seperti tabel 4.xx berikut ini.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Gambar | Nilai ED | Urutan | Label |
| 34 | IMG\_0505.JPG | 0,944241042 | 1 | Sapi |
| 3 | IMG\_0520.JPG | 1,045647037 | 2 | Babi |
| 16 | IMG\_0538.JPG | 1,518425338 | 3 | Babi |

Pada tabel tersebut, label yang paling banyak muncul adalah Babi, sehingga data tersebut diidentifikasi sebagai daging Babi.

Proses pengujian dilakukan pada seluruh data uji, sebanyak 20 buah. Hasilnya ditampilkan pada tabel 4xx berikut ini.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Gambar | Hasil Identifikasi | Kesesuaian |
| 1 | IMG\_0519.JPG | Babi | Benar |
| 2 | IMG\_0521.JPG | Sapi | Salah |
| 3 | IMG\_0523.JPG | Babi | Benar |
| 4 | IMG\_0524.JPG | Babi | Benar |
| 5 | IMG\_0531.JPG | Babi | Benar |
| 6 | IMG\_0537.JPG | Babi | Benar |
| 7 | IMG\_0545.JPG | Babi | Benar |
| 8 | IMG\_0550.JPG | Babi | Benar |
| 9 | IMG\_0551.JPG | Babi | Benar |
| 10 | IMG\_0556.JPG | Sapi | Salah |
| 11 | IMG\_0503.JPG | Sapi | Benar |
| 12 | IMG\_0509.JPG | Sapi | Benar |
| 13 | IMG\_0560.JPG | Sapi | Benar |
| 14 | IMG\_0574.JPG | Sapi | Benar |
| 15 | IMG\_0582.JPG | Sapi | Benar |
| 16 | IMG\_0583.JPG | Sapi | Benar |
| 17 | IMG\_0584.JPG | Sapi | Benar |
| 18 | IMG\_0585.JPG | Sapi | Benar |
| 19 | IMG\_0586.JPG | Sapi | Benar |
| 20 | IMG\_0587.JPG | Sapi | Benar |

Proses pengujian sebanyak 20 data menggunakan nilai N = 3, mendapatkan hasil identifikasi benar sebanyak 18 dan salah sebanyak 2 , sehingga akurasi sistem menggunakan nilai N = 3 yaitu sebesar 90 %.