

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Tanda tangan merupakan salah satu bentuk biometrik atau ciri khas dari setiap manusia, tanda tangan sendiri menjadi sangat penting dalam legalisasi atau verifikasi sebuah dokumen maka diperlukan sebuah penelitian yang dapat membantu dalam mengidentifikasi keaslian dari sebuah tanda tangan.

Terdapat beberapa penelitian mengenai sistem identifikasi keaslian tanda tangan menggunakan beberapa metode ekstraksi fitur dan klasifikasi yang berbeda yang dapat dilihat pada tabel 3.1

Tabel 3.1 Penelitian identifikasi keaslian tanda tangan

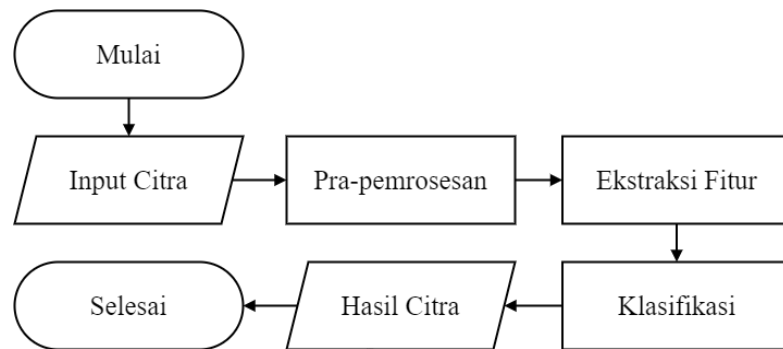
Judul	Permasalahan	Metode	Hasil
Pengenalan Pola Tanda Tangan Menggunakan Metode Jaringan Syaraf Tiruan <i>Backproagation</i>	Pengenalan tanda tangan secara manual memiliki waktu panjang jika terdapat banyak dokumen	Ekstraksi Fitur: vertical splitting dan horizontal splitting Klasifikasi: JST-BP dan SVM	JST-BP dapat mengidentifikasi dengan akurasi 74% dan SVM dengan akurasi mengidentifikasi sebesar 90.5%.
<i>Machine Learning</i> untuk Identifikasi Tanda Tangan Menggunakan GLCM dan <i>Euclidean Distance</i>	Identifikasi tanda tangan masih dilakukan secara manual dengan mencocokkan tanda tangan pada waktu transaksi.	Ekstraksi Fitur: GLCM Klasifikasi: <i>Euclidean Distance</i>	<i>Euclidean Distance</i> dengan piksel 100x100 mengidentifikasi dengan akurasi 67,5% dan piksel 50x50 dengan akurasi 65%

Sumber : Peneliti (Didiary Shandy S)

3.2 Hasil Analisis

Identifikasi tanda tangan dalam penelitian ini, menggunakan salah satu metode ekstraksi fitur *Uniform Local Binary Pattern* (ULBP) dan algoritma yang digunakan mencakup pendekatan *machine learning*, khususnya menggunakan *K-Nearest Neighbor* (KNN) sebagai metode klasifikasi. Metode *Uniform Local Binary Pattern* (ULBP) sebagai ekstraksi fitur dalam mengidentifikasi jenis kayu (Xiaoxia Yang et al., 2023) dalam penelitian tersebut berhasil dalam mengenali jenis kayu berdasarkan tekstur lokal pada kulit kayu menggunakan ekstraksi fitur ULBP dan berkesimpulan bahwa hasil metode ULBP lebih baik dari beberapa metode lanjutan LBP. Metode *K-Nearest Neighbor* digunakan untuk klasifikasi dalam mengidentifikasi telapak tangan (Retnoningrum et al., 2019) pada penelitian tersebut berhasil dalam mengimplementasikan K-NN sebagai klasifikasi dengan kesimpulan memperhatikan parameter K yang diterapkan semakin kecil jarak tetangga semakin stabil hasil yang didapatkan.

Langkah-langkah pemrosesan citra dapat dilihat pada gambar 3.1 memperlihatkan *flowchart* yang merupakan cara kerja sistem secara visual dari pertama sistem dimulai hingga sistem berakhir. dimulai dengan menginput data uji dan latih. Penginputan data ke dalam sistem dilakukan secara manual dengan memilih folder data uji dan data latih yang telah disiapkan dalam penelitian. Tahap selanjutnya akan dilakukan pra-pemrosesan pada data set untuk memperjelas citra, dilanjutkan ekstraksi menggunakan *Uniform Local Binary Pattern* (ULBP) dan klasifikasi pada data latih. Setelah melakukan tahapan tersebut, akan dilakukan prediksi kecocokan tanda tangan antara data latih dan data uji untuk mengidentifikasi keaslian tanda tangan tersebut, Kemudian akan dilakukan akurasi pada setiap citra data uji.

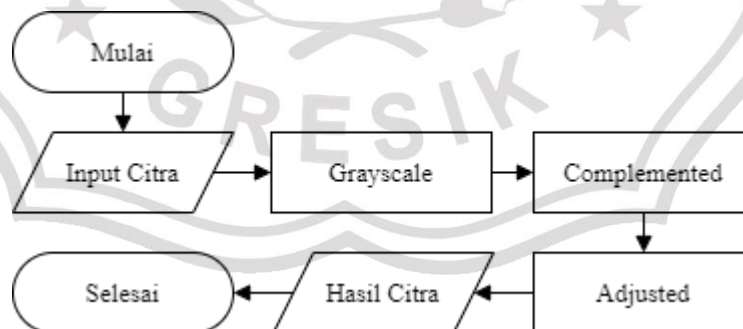


Gambar 3.1 Flowchart pemrosesan citra

Adapun penjelasan mengenai setiap tahapan yang dilakukan dalam pemrosesan citra, sebagai berikut:

3.2.1 Pemrosesan Awal Citra (*Pre-processing*)

Pra-pemrosesan citra akan memperlihatkan pengelolaan citra yang bertujuan untuk meningkatkan kualitas citra dan dapat digunakan dalam proses ekstraksi fitur dengan citra yang baik. Gambar 3.2 menggambarkan urutan dari pra-pemrosesan citra yang dilakukan pada penelitian ini. Terdapat beberapa tahap dalam pra-pemrosesan yang dilakukan yaitu melakukan *Grayscale* dengan mengubah citra awal RGB menjadi citra *Gray*, dilanjutkan dengan melakukan *complemented* dengan mengubah warna *background* yang awal berwarna putih menjadi hitam setelah melakukan *complemented* dilakukan *adjusted* untuk meningkatkan kualitas citra tanda tangan.

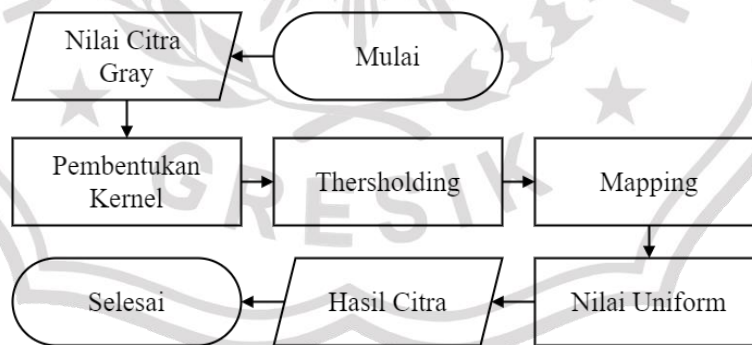


Gambar 3.2 Flowchart pra-pemrosesan citra

3.2.2 Ekstraksi Fitur

Algoritma ULBP (*Uniform Local Binary Pattern*) merupakan salah satu metode yang digunakan untuk mengekstraksi ciri tekstur pada objek citra

grayscale. Dalam *data set* citra tanda tangan yang diterapkan pada penelitian ini, citra awal sudah berupa *grayscale* pada tahap awal pra-pemrosesan. Penggunaan *uniform Local Binary Pattern* (ULBP) untuk menemukan pola-pola tekstur lokal pada setiap citra. Gambar 3.3 menggambarkan urutan dari ekstraksi fitur yang dilakukan setelah melakukan pra-pemrosesan citra. Terdapat beberapa tahapan ekstraksi fitur yang dilakukan menggunakan *uniform Local Binary Pattern* (ULBP) pertama dilakukan pembentukan kernel dengan matriks ukuran 3x3 dengan, dilanjutkan dengan *thersholding* atau konversi nilai piksel menjadi biner dengan cara melakukan perbandingan nilai tetangga piksel terdekat dengan nilai tengah piksel pada citra *grayscale*. Apabila nilai *grayscale* lebih besar atau sama dengan nilai *grayscale* di titik pusat, maka akan diberi nilai 1 dan jika nilai *grayscale* lebih kecil dari nilai *grayscale* yang ada di titik pusat, maka akan diberi nilai 0 begitu juga sebaliknya. Setelah melakukan *thersholding* dilanjutkan tahap *mapping* untuk mengonversi nilai biner kedalam desimal dengan cara menjumlahkan hasil perkalian antara hasil *thersholding* dengan matrik *mapping* pada posisi yang sama. Nilai desimal ini menjadi nilai baru untuk piksel yang berada ditengah ketetanggaan. Setelah didapatkan nilai LBP pada setiap piksel maka akan diambil nilai atau pola *uniform*.

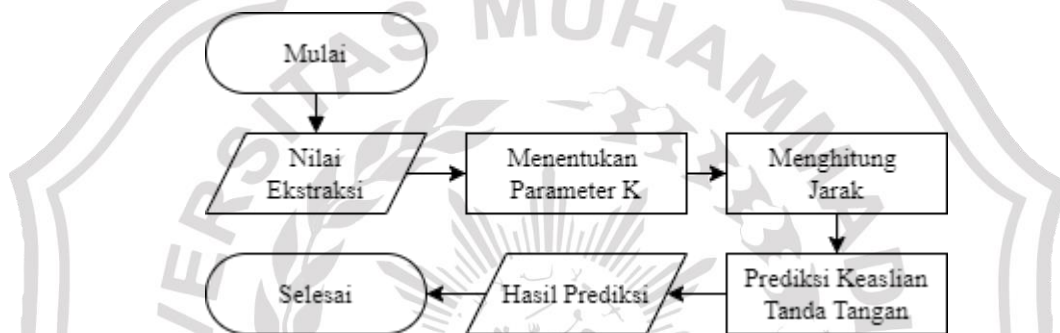


Gambar 3.3 Flowchart ekstraksi fitur menggunakan ULBP

3.2.3 Klasifikasi

Proses dalam pengelompokan untuk mengidentifikasi keaslian tanda tangan pada penelitian ini, dilakukan dengan menggunakan KNN. Setelah melakukan pra-pemrosesan kemudian citra akan diekstraksi menggunakan

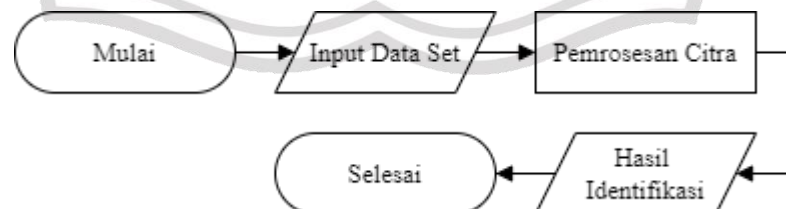
Uniform Local Binary Pattern (ULBP) dan mendapatkan hasil biner yang tergolong pola *uniform*, dilanjutkan dengan melakukan pengelompokan menggunakan KNN yang dapat dilihat pada gambar 3.4 yang menjelaskan tahapan dari klasifikasi menggunakan KNN yang dilakukan pada penelitian ini. Terdapat beberapa tahap dalam klasifikasi yang dilakukan pertama dengan menentukan parameter K untuk ketetangaan, dilanjutkan menghitung jarak menggunakan rumus persamaan 2.4. Setelah mendapatkan jarak dari setiap data uji terhadap data latih maka akan dilakukan pemrediksian tanda tangan pada setiap data uji dan dilanjutkan menghitung hasil akurasi menggunakan akurasi *Confusion Matrix* rumus persamaan 2.5.



Gambar 3.4 Flowchart Klasifikasi Menggunakan KNN

3.2.4 Diagram Alur (Flowchart)

Diagram alur dalam sistem identifikasi keaslian tanda tangan menggunakan *Uniform Local Binary Pattern* (ULBP) dan algoritma *K-Nearest Neighbors* (KNN) yang berdasarkan pola biner yang dihasilkan oleh metode LBP dan kemudian melakukan klasifikasi menggunakan algoritma KNN.



Gambar 3.5 Flowchart Sistem Identifikasi Tanda Tangan

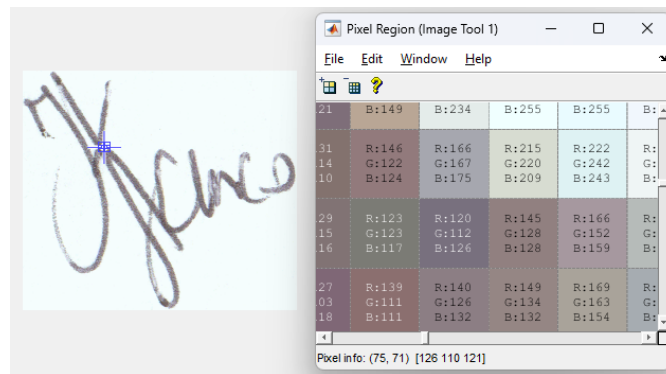
Berikut penjelasan mengenai *flowchart* pada gambar 3.5 menggambarkan sistem identifikasi tanda tangan menggunakan metode *Uniform Local Binary Pattern* (ULBP) dan algoritma *K-Nearest Neighbors* (KNN):

1. *User* membuka sistem identifikasi menggunakan Matlab
2. *User* memasukkan data set yang terdiri dari data latih dan data uji
3. Sistem akan memproses data set tersebut dengan menggunakan ULBP dan KNN
4. Sistem akan mengeluarkan hasil *output* berupa hasil identifikasi keaslian tanda tangan

3.3 Representasi Model

Citra tanda tangan yang diambil dari situs website kaggle.com dengan link tanda tangan <https://www.kaggle.com/datasets/robinreni/signature-verification-dataset/data> dengan citra tanda tangan sebanyak 1.193 dengan format PNG yang telah diproses *cropping* dengan ukuran yang telah disesuaikan. Citra tanda tangan memiliki total 46 varian tanda tangan yang telah memiliki pola tanda tangan asli dan palsu pada setiap varian dengan total 708 tanda tangan asli dan 485 tanda tangan palsu. Penelitian ini mengambil beberapa varian tanda tangan yang akan digunakan dengan total 13 varian tanda tangan. Data set terbagi menjadi dua dengan jumlah data latih 202 citra tanda tangan asli dan data uji 134 citra tanda tangan palsu. Representasi model dibuat untuk menjelaskan mengenai tahapan pra-pemrosesan, ekstraksi fitur dan klasifikasi dalam hasil analisis. Sebagai berikut:

1. Pada pra-pemrosesan dapat dilihat dalam gambar 3.2 yang menjelaskan mengenai langkah-langkah pra-pemrosesan dalam penelitian. Adapun penjelasan mengenai setiap tahapan pra-pemrosesan, sebagai berikut:
 - a. tahap awal pra-pemrosesan ini dilakukan *Grayscale* untuk mengonversi citra berwarna dalam format RGB yang memiliki intensitas tiga saluran warna merah (*red*), hijau (*green*), dan biru (*blue*) seperti pada gambar 3.6.

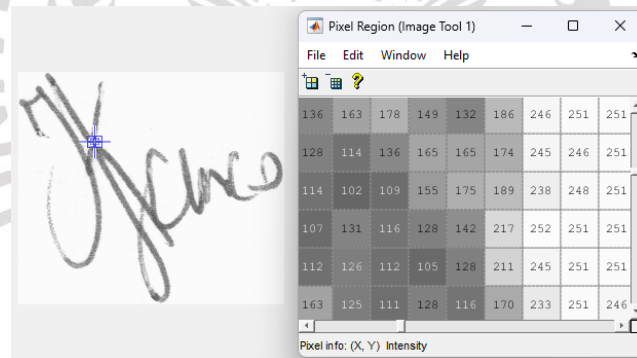


Gambar 3.6 Citra RGB dan nilai kanal penyusunnya

Untuk mengubah menjadi *Grayscale* dilakukan dengan persamaan sederhana contoh rata-rata nilai kecerahan dalam tiga intensitas warna RGB yang berbeda-beda menggunakan rumus persamaan (2.1)

$$GS_{(x,y)} = 0.2989 * R_{(x,y)} + 0.5870 * G_{(x,y)} + 0.1141 * B_{(x,y)}$$

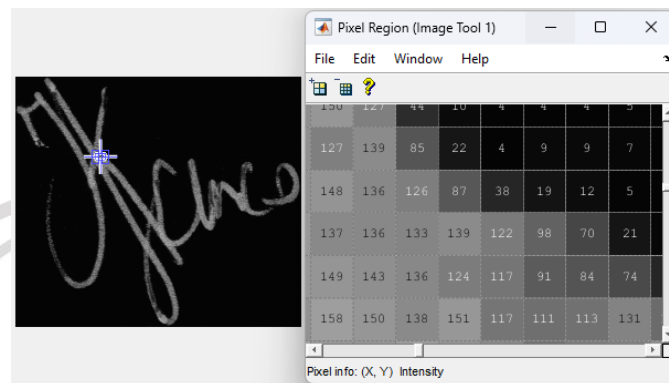
Proses *Grayscale* dilakukan untuk mempermudah proses gambar lebih lanjut, karena hanya memiliki satu nilai kanal pada setiap pikselnya yang bernilai 0-255.



Gambar 3.7 Hasil *Grayscale* dengan nilai piksel

- b. Selanjutnya dilakukan proses *complemented*, untuk menghasilkan nilai *complemented* dengan mengurangi nilai maksimum terhadap nilai piksel asli serta bertujuan untuk melakukan negasi atau kebalikan warna (hitam menjadi putih, maupun sebaliknya). Citra tanda tangan yang akan diidentifikasi akan di lakukan proses ekstraksi fitur untuk mencirikan objek tersebut. Dalam Matlab, objek akan dikenali berdasarkan warna yang lebih terang dibandingkan dengan warna latar belakang. Oleh karena itu, diperlukan proses konversi di mana tanda tangan dalam data citra pada

penelitian ini menggunakan tinta hitam pada kertas putih akan dibalik kondisinya sehingga kertas putih yang merupakan latar akan diberikan warna lebih gelap dibandingkan tanda tangan yang merupakan objek penelitian. Penggunaan metode *Complemented* dilakukan untuk meningkatkan kemampuan pengenalan objek dan membantu membedakan antara tanda tangan dan latar belakang.



Gambar 3.8 Hasil *Complemented* dengan nilai piksel

- c. dilanjutkan proses *adjusted* yang bertujuan untuk memperbaiki kontras pada citra dengan mengubah intensitas piksel pada citra asli. Penggunaan *adjusted* dalam penelitian ini adalah untuk mempertegas pola atau garis dari citra tanda tangan dengan melakukan penyesuaian intensitas agar garis-garis tanda tangan yang terlihat samar atau terputus dapat menjadi terlihat lebih jelas dan teridentifikasi dengan lebih baik. Serta agar citra tanda tangan dapat lebih mudah diinterpretasi pada saat ekstraksi fitur dan garis-garis tanda tangan dapat dikenali dengan lebih baik.



Gambar 3.9 Hasil *Complemented* dan *Adjusted*

- Setelah melakukan tahap pra-pemrosesan hasil citra gray akan masuk pada tahap ekstraksi. Jenis spesifik yang digunakan dalam penelitian ini adalah ULBP yang menggunakan titik ketetanggaan (*sampling point*) = 8, dengan

jarak titik ketetangaan (*radius*) = 1. Gambar 3.10 menggambarkan ilustrasi LBP dengan contoh sebuah citra yang memiliki matriks berukuran 5 x 5

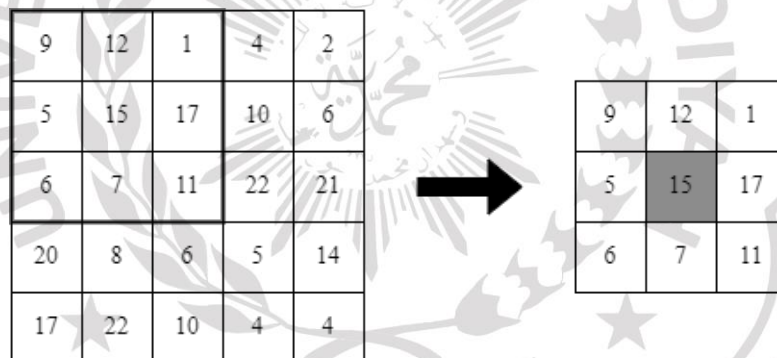
A =

9	12	1	4	2
5	15	17	10	6
6	7	11	22	21
20	8	6	5	14
17	22	10	4	4

Gambar 3.10 Contoh matriks

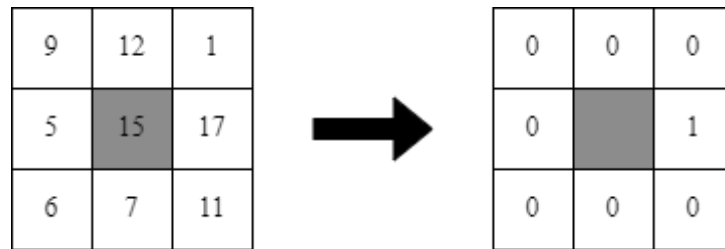
Adapun langkah-langkah untuk membentuk matriks *Local Binary Pattern* (LBP) sebagai berikut:

- Pada awal proses akan membagi matriks citra menjadi sejumlah bagian-bagian yang lebih kecil atau disebut sebagai kernel, yang berukuran matriks 3 x 3 pada gambar 3.11



Gambar 3.11 Pembentukan kernel

- Setelah terbentuk kernel dengan ukuran matriks 3 x 3 dilakukan perbandingan nilai piksel pada titik pusat yang berada pada 8 titik tetangga sekelilingnya. Jika nilai piksel tetangga lebih besar atau sama besar dari nilai piksel titik pusat ($17 > 15 = 1$), maka nilai LBP yang dihasilkan piksel tetangga adalah 1, namun sebaliknya. Jika nilai piksel tetangga lebih kecil dari nilai titik pusat ($9 < 15 = 0$), maka dihasilkan nilai 0 pada piksel tetangga seperti pada gambar 3.12



Gambar 3.12 Konversi nilai piksel ke biner

- c. Nilai akan di susun mengikuti arah jarum jam sehingga menjadi nilai bilangan biner 8 bit seperti pada gambar 3.13



Gambar 3.13 Nilai bilangan biner 8 bit

- d. Mengonversikan bilangan menjadi desimal pada penelitian ini menggunakan *mapping* dengan mengalikan setiap digit nilai biner dengan nilai eksponensial dari 2, kemudian menjumlahkan semua hasil perkalian tersebut dapat dilihat pada gambar 3.14 di bawah. Nilai desimal yang dihasilkan akan menjadi nilai LBP pada satu kernel



$$00001000 = 0 \times 128 + 0 \times 64 + 0 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1 = 8$$

Gambar 3.14 Mapping

Sehingga nilai konversi bilangan biner ke desimal yang menghasilkan nilai 8 akan menjadi representasi nilai *Local Binary Pattern* (LBP) dan akan diletakkan pada kernel seperti pada gambar 3.15 di bawah ini

	8	

Gambar 3.15 Nilai titik pusat berubah menjadi nilai LBP

- e. Ulangi langkah-langkah pada tahapan a – d kembali untuk menghitung LBP pada piksel lainnya, dan akan memperoleh hasil akhir seperti pada gambar 3.16

A =

24	48	248	112	224
62	8	16	240	224
60	110	11	0	128
16	244	231	207	3
10	1	128	146	131

Gambar 3.16 Hasil nilai piksel perhitungan LBP

- d. Kemudian dilakukan pengambilan nilai atau pola *Uniform Local Binary Pattern* (ULBP) akan mengambil nilai yang memiliki maksimum dua transisi (perubahan dari 0 – 1 maupun sebaliknya) atau disebut dengan pola biner *uniform*.

Tabel 3.2 Hasil pola setiap biner

Desimal	Biner	Pola
24	00011000	Non-Uniform
48	00110000	Non-Uniform
248	11111000	Non-Uniform
112	01110000	Non-Uniform
224	11100000	Non-Uniform
62	00111110	Non-Uniform
8	00001000	Uniform

16	00010000	Uniform
240	11110000	Non-Uniform
60	00111100	Non-Uniform
110	01101110	Non-Uniform
11	00001011	Non-Uniform
0	00000000	Uniform
128	10000000	Uniform
244	11110100	Non-Uniform
231	11100111	Non-Uniform
207	11001111	Non-Uniform
3	00000011	Uniform
10	00001010	Uniform
1	00000001	Uniform
146	10010010	Non-Uniform
131	10000011	Non-Uniform

Sumber : Peneliti (Didiary Shandy S)

Tabel 3.2 memperlihatkan hasil biner dari matriks gambar 3.16 beserta pola yang dihasilkan dari setiap biner. Setelah mengetahui pola dari setiap biner maka akan diambil pola *uniform* sebagai hasil dari ekstraksi yang akan digunakan untuk klasifikasi menggunakan algoritma *K-Nearest Neighbor (KNN)*. Nilai-nilai biner yang tergolong *uniform* dapat dilihat pada tabel 3.3

Tabel 3.3 Nilai biner pola uniform

Desimal	Biner	Pola
8	00001000	Uniform
16	00010000	Uniform
0	00000000	Uniform
128	10000000	Uniform
3	00000011	Uniform
10	00001010	Uniform

1	00000001	Uniform
---	----------	---------

Sumber : Peneliti (Didiary Shandy S)

3. Setelah melakukan ekstraksi fitur dan mendapatkan hasil fitur, dilanjutkan dengan pencarian nilai jarak *euclidean* menggunakan rumus persamaan 2.4.

$$f(v_1, v_2) = \sqrt{\sum_{k=1}^N (v_1(k) - v_2(k))^2}$$

Apabila telah menemukan jarak *euclidean* akan diurutkan berdasarkan jarak terdekat. Kemudian dilakukan proses K-NN yang di mana akan bekerja mencari jarak terdekat dari data latih dengan data uji yang diambil terbanyak dari K yang telah ditentukan.

Langkah-langkah proses dalam algoritma klasifikasi K-NN, sebagai berikut:

- Menentukan data latih yang digunakan, contoh salah satu citra data latih hasil ekstraksi fitur ULBP
Fitur data latih 1: [10444,1669,266,273,1484,285,519,367,476,....., 5922]
Menentukan data uji yang digunakan, contoh salah satu citra data uji hasil ekstraksi fitur ULBP
Fitur data uji 1: [16678,2951,559,655,2527,606,897,608,764,.....,13272]
- Menentukan parameter K, dalam proses untuk menentukan K dilakukan dengan mempertimbangkan dalam jumlah data yang berjumlah genap maka sebaiknya menggunakan nilai K ganjil dan begitu juga sebaliknya.
- Menghitung jarak *euclidean* dengan menggunakan persamaan 2.4, contoh perhitungan menggunakan *euclidean* untuk menemukan jarak terdekat, dengan perhitungan dilakukan antara fitur data latih 1 dan fitur data uji 1

$$d1 = \sqrt{\begin{aligned} &(uji1_{fitur1} - latih1_{fitur1})^2 + (uji1_{fitur2} - latih1_{fitur2})^2 + \\ &(uji1_{fitur3} - latih1_{fitur3})^2 + (uji1_{fitur4} - latih1_{fitur4})^2 + \dots + \\ &(uji1_{fitur59} - latih1_{fitur59})^2 \end{aligned}}$$

$$d1 = \sqrt{(10444 - 16678)^2 + (1669 - 2951)^2 + (266 - 559)^2 + (273 - 655)^2 + \dots + (5922 - 13272)^2}$$

$$d1 = \sqrt{47207200} \approx 6866.1981$$

Perhitungan manual yang telah dilakukan di atas merupakan perhitungan data uji 1 dengan data latih 1 maka selanjutnya akan menghitung nilai *euclidean* dari data uji 1 terhadap data latih 2, data uji 1 terhadap data latih 3 dan seterusnya, hasil perhitungan dari setiap data latih terhadap data uji dapat dilihat di tabel 3.4.

Tabel 3.4 Nilai jarak euclidean pada setiap data latih

Data Latih	Jarak Euclidean	Kelas
A.1	6866.1981	Asli
A.2	19164.0313	Asli
A.3	19451.7333	Asli
A.4	18827.0023	Asli
A.5	19639.1894	Asli
P.1	19312.0824	Palsu
P.2	19247.6641	Palsu
P.3	19385.1755	Palsu
P.4	19349.0707	Palsu
P.5	19252.0903	Palsu

Sumber : Peneliti (Didiary Shandy S)

- d. Setelah menemukan semua jarak dilanjutkan dengan mengurutkan data dari data terdekat sampai dengan data terjauh, dapat dilihat pada tabel 3.5.

Tabel 3.5 Pengurutan nilai terdekat sampai terjauh

Data Latih	Jarak Euclidean	Kelas
A.1	6866.1981	Asli
A.4	18827.0023	Asli
A.2	19164.0313	Asli
P.2	19247.6641	Palsu

P.5	19252.0903	Palsu
P.3	19385.1755	Palsu
P.1	19312.0824	Palsu
P.4	19349.0707	Palsu
A.3	19451.7333	Asli
A.5	19639.1894	Asli

Sumber : Peneliti (Didiary Shandy S)

- e. Langkah selanjutnya setelah melakukan perhitungan jarak pada seluruh data maka akan diproses untuk mendapatkan prediksi kelas untuk menentukan data uji 1 terletak pada kelas asli atau palsu. Hasil klasifikasi menggunakan parameter $K=1$ dapat dilihat pada tabel 3.6 tetangga terdekat menunjukan kelas asli, sehingga dengan menggunakan jumlah 10 data latih dan 1 data uji dengan $K=1$ teridentifikasi kelas asli.

Tabel 3.6 Hasil klasifikasi dengan parameter $K=1$

Data Latih	Jarak Euclidean	Kelas
A.1	6866.1981	Asli
A.4	18827.0023	Asli
A.2	19164.0313	Asli
P.2	19247.6641	Palsu
P.5	19252.0903	Palsu
P.3	19385.1755	Palsu
P.1	19312.0824	Palsu
P.4	19349.0707	Palsu
A.3	19451.7333	Asli
A.5	19639.1894	Asli

Sumber : Peneliti (Didiary Shandy S)

4. Setelah seluruh data uji diproses untuk mendapatkan nilai prediksi kelas maka akan mencari nilai akurasi dengan melakukan perhitungan menggunakan persamaan 2.5.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \%$$

Keterangan

True Positive (TP) = Jumlah data bernilai positif dan diprediksi benar sebagai positif.

False Positive (FP) = Jumlah data bernilai negatif tetapi diprediksi benar sebagai positif.

False Negative (FN) = Jumlah data bernilai positif tetapi diprediksi benar sebagai negatif.

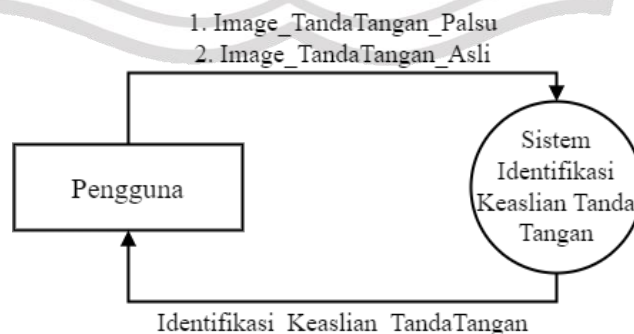
True Negative (TN) = Jumlah data bernilai negatif dan diprediksi benar sebagai negatif.

3.4 Perancangan Sistem

Perancangan sistem merupakan suatu proses yang melibatkan perencanaan dan elemen-elemen yang dapat membentuk suatu sistem yang digunakan untuk mencapai tujuan tertentu. Perancangan sistem bertujuan untuk membuat kerangka kerja sebuah sistem yang dapat mendukung dan mencapai tujuan dari sistem yang dibuat, serta untuk memberikan gambaran secara umum mengenai sistem yang dibuat.

3.4.1 Diagram Konteks

Diagram konteks (*context diagram*) untuk sistem identifikasi tanda tangan menggunakan metode Uniform Local Binary Pattern (LBP) dan algoritma K-Nearest Neighbors (KNN). Gambar 3.17 menjelaskan bahwa hanya terdapat 1 entitas eksternal yang berhubungan dengan sistem, yaitu pengguna yang dapat memasukkan data latih dan data uji, serta akan mendapatkan *output* dari sistem berupa hasil identifikasi

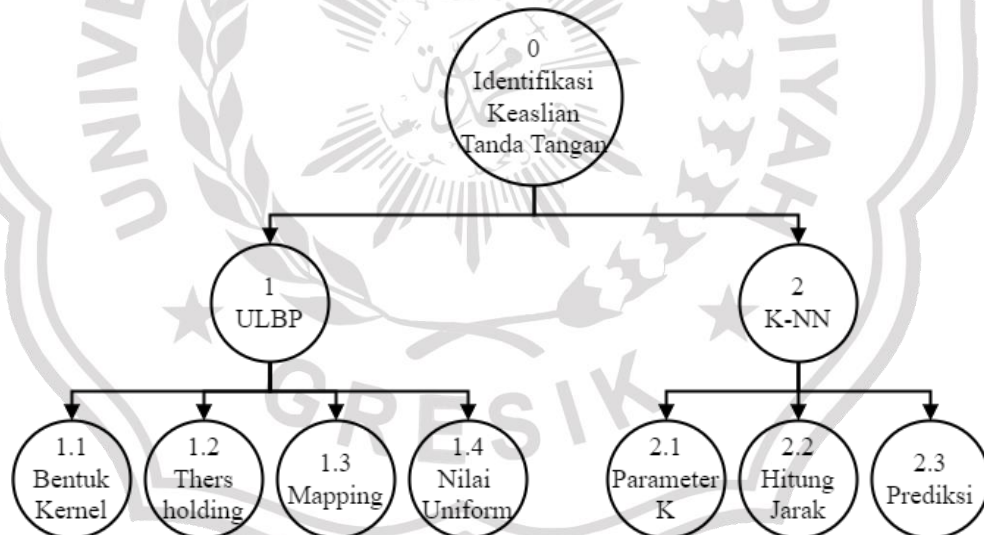


Gambar 3.17 Diagram Konteks Sistem Identifikasi Tanda Tangan

3.4.2 Diagram *Hierarchy Plus Input-Process-Output (HIPO)*

Diagram HIPO menggambarkan hierarki modul dalam sistem serta input, proses, dan output yang terkait dengan setiap modul. HIPO membantu dalam memahami struktur sistem secara keseluruhan dan bagaimana setiap bagian saling berhubungan dan berinteraksi. Gambar 3.18 menjelaskan alur keseluruhan sistem, sebagai berikut:

1. *Top level*: Sistem identifikasi keaslian tanda tangan dengan menerapkan ULBP dan K-NN
2. *Level 2*: Berisi proses yang ada di dalam sistem, penerapan ULBP sebagai ekstraksi fitur dan K-NN sebagai klasifikasi
3. *Level 3*: Penerapan ULBP dan K-NN
 - a. ULBP dengan proses pembentukan kernel, *thersholding*, *mapping*, dan nilai uniform
 - b. K-NN dimulai dengan proses menentukan parameter K, perhitungan jarak, dan prediksi tanda tangan.



Gambar 3.18 Diagram HIPO Sistem Identifikasi Tanda Tangan

3.4.3 Data *Flow Diagram (DFD)*

Data *Flow Diagram (DFD)* adalah representasi grafis dari aliran data dalam sistem. DFD menggambarkan bagaimana data diproses dalam sistem,

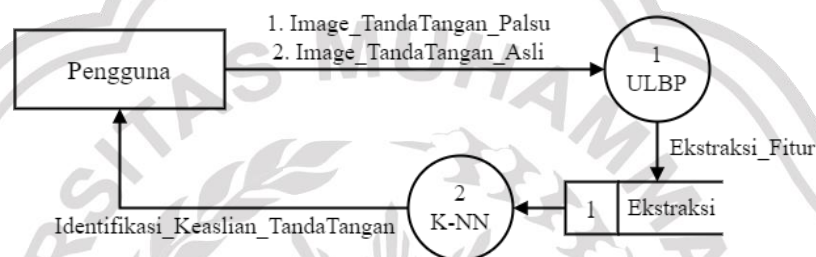
mulai dari *input*, proses, hingga *output*. DFD dapat membantu pengembang dalam memahami secara visual alur kerja sistem.

3.4.3.1 DFD Level 1

Pada gambar 3.19 akan menjelaskan mengenai beberapa tahapan proses dalam sistem, sebagai berikut:

Proses 1 untuk melakukan ekstraksi fitur menggunakan ULBP dari data gambar yang telah dimasukkan

Proses 2 akan melakukan klasifikasi menggunakan K-NN dari hasil dari ekstraksi fitur.



Gambar 3.19 DFD Level 1 Sistem Identifikasi Tanda Tangan

3.4.3.2 DFD Level 2

Pada DFD level 2 akan menjelaskan mengenai tahapan proses 1 dan 2 dari DFD level 1 yang terbagi menjadi dua penjelasan dari ULBP dan K-NN, sebagai berikut:

a. DFD level 2 *Uniform Local Binary Pattern* (ULBP)

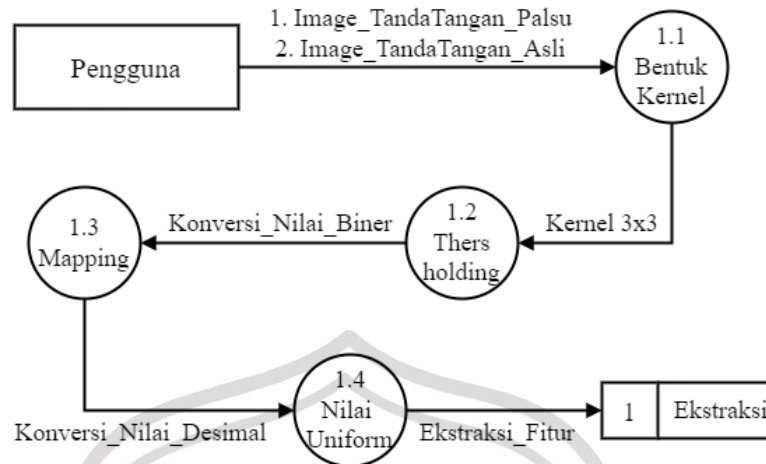
Pada gambar 3.20 akan terjadi proses ULBP sebagai berikut:

Proses 1.1 Pembentukan kernel dengan matriks 3x3 pada setiap image data yang telah dimasukkan

Proses 1.2 Proses *Thesholding* bertujuan untuk menemukan ambang batas dengan melakukan konversi nilai piksel ke dalam biner

Proses 1.3 Proses *Mapping* bertujuan untuk melakukan konversi nilai biner ke dalam nilai desimal sebagai nilai LBP

Proses 1.4 Nilai uniform bertujuan untuk mengambil nilai yang hanya termasuk ke dalam uniform



Gambar 3.20 DFD Level 2 ULBP Sistem Identifikasi Tanda Tangan

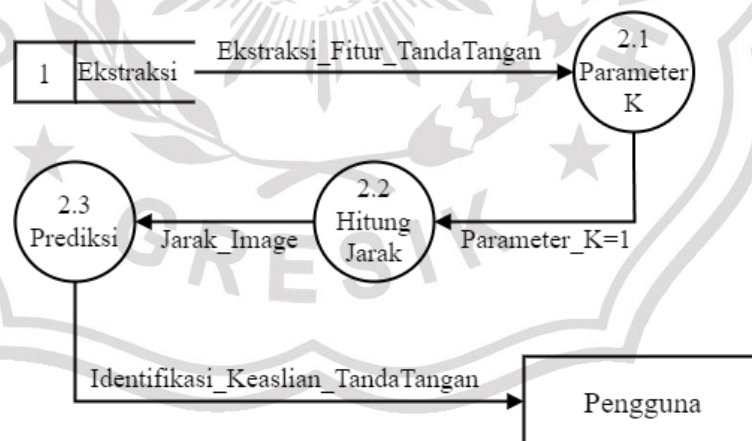
b. DFD level 2 *K-Nearest Neighbor* (K-NN)

Pada gambar 3.21 akan terjadi proses K-NN sebagai berikut:

Proses 2.1 Menentukan parameter K yang dilakukan untuk menentukan ketetangaan sesuai dengan skenario pengujian

Proses 2.2 Perhitungan jarak dilakukan untuk menemukan jarak antara data uji dan setiap data latih

Proses 2.3 Proses prediksi bertujuan untuk menentukan kecocokan image tanda tangan kepada setiap data latih



Gambar 3.21 DFD Level 2 K-NN Sistem Identifikasi Tanda Tangan

3.5 Skenario Pengujian

Pengujian dalam penelitian ini dilakukan untuk menguji model yang dihasilkan dari ekstraksi fitur LBP dan klasifikasi KNN sebelumnya. Pada

penelitian ini model yang dibangun akan melihat keakuratannya dengan memperhatikan parameter pengujian. Pada metode ekstraksi fitur ULBP parameter yang akan diperhatikan pada *radius* dan jumlah tetangga sedangkan pada metode klasifikasi KNN akan memperhatikan pada K ketetangaan. Berikut merupakan jenis skenario pengujian.

1. Pengujian akurasi berdasarkan parameter pada metode yang digunakan dalam metode ULBP, parameter yang digunakan dalam pengujian menggunakan nilai $R = 1$ dan $P = 8$, di mana R adalah jarak tetangga atau *radius* dan P merupakan jumlah tetangga. Nilai ini diambil berdasarkan penelitian “Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)” (Retnoningrum et al., 2019). Serta kesimpulan dari penelitian sebelumnya dengan judul “Identifikasi COVID-19 berdasarkan citra X-ray paru-paru menggunakan metode *Local Binary Pattern* (LBP) dan algoritma Random Forest” (Suaib et al., 2022).
2. Pengujian akurasi berdasarkan parameter KNN yang akan diujikan adalah K ketetangaan di mana K yang digunakan adalah $K = 1$ dan $K = 3$. Tetapi pada penelitian ini akan memperhatikan nilai $K = 1$ berdasarkan hasil akurasi tertinggi yang telah dilakukan penelitian sebelumnya dengan judul “Identifikasi Pengenalan Wajah dengan Menggunakan Metode KNN (K-NEAREST NEIGHBOR) dan LBPH (LOCAL BINARY PATTERN HISTOGRAM) Untuk Sistem Presensi” (Yulianti et al., 2022)
3. Pada proses pengujian yang dilakukan akan menggunakan alat uji akurasi yang diambil dari perbandingan nilai dengan *True Positif* dan *True Negative* terhadap seluruh data uji menggunakan rumus 2.5. Contoh terdapat data *dummy* seperti pada Tabel 3.7.

Tabel 3.7 *Confusion matrix* dengan data *dummy*.

Kelas Sebenarnya	Hasil Klasifikasi	Positive	Negative
	Positive	20	5
Negative	3	72	

Sumber : Peneliti (Didiary Shandy S)

Berdasarkan confusion matrix di atas, dapat dihitung nilai akurasi menggunakan *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Nilai hitung akurasi dilakukan dengan menggunakan persamaan 2.5.

$$Accuracy = \frac{20 + 72}{20 + 5 + 3 + 72} \times 100\% = \frac{92}{100} \times 100\% = 92\%$$

3.5.1 Kebutuhan Perangkat Keras dan Perangkat Lunak

Pada penelitian ini alat dan bahan yang digunakan berupa *software* dan *hardware*. Alat – alat yang digunakan untuk melakukan penelitian ini menggunakan komputer dengan spesifikasi *processor* i3-10105F 3.70GHz, RAM 16GB dan menggunakan *Video Graphics Array* (VGA) RX 580 SP, VRAM 8GB, *Operating System* Windows 11 pro 64-bit serta untuk software yang digunakan adalah Matlab R2021a yang diperlukan untuk membuat program dan komputasi.