BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebagai upaya dalam menguatkan landasan penelitian, penulis menganalisis serta meriset hasil dari penelitian terdahulu yang berkaitan atau relevan dengan topik penelitian, berikut adalah hasil dari penelitian tersebut:

a. Yuliana Diah Pristanti, Panca Mudjirahardjo, Achmad Basuki dengan judul penelitian "Identifikasi Tanda Tangan dengan Ekstraksi Ciri GLCM dan LBP" pada penelitian ini data yang diujikan sebanyak 15 sampel dengan total tanda tangan 150 bertinta hitam yang terbagi menjadi 105 data latih dan 45 data uji, dan memiliki 20 sampel dengan total jumlah tanda tangan sebanyak 200 bertinta warna selain hitam terbagi menjadi 140 data latih dan 60 data uji, pada proses *Pre-processing* dilakukan dengan tahapan *Croppingg* dan *Resize.* Metode ekstraksi fitur menggunakan *Gray Level Co-Occurrence Matrix* (GLCM) dan *Local Binary Pattern* (LBP) kedua metode tersebut direduksi *menggunakan K-Means Clustering*. Menghasilkan akurasi yang cukup tinggi terhadap 150 tanda tangan berwarna hitam ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) mencapai tingkat akurasi 80,00% sedangkan menggunakan tinta selain berwarna hitam menghasilkan akurasi 78,33%. Pada penelitian ini menyimpulkan bahwa jumlah data set berpengaruh pada hasil akurasi(Diah Pristanti et al., 2019).
b. Angga Dwi Putriana, Dila Seltika Canta, Elvin Leander Hadisaputro, Nuorma Wahyuni dengan judul penelitian "Implementasi Backpropagation untuk Identifikasi Tanda Tangan Digital" penelitian ini memiliki tahapan pra-pemrosesan yang digunakan, tahap pertama dilakukan *resize* untuk meminimalisir lama waktu komputasi, tahap kedua dilakukan *thinning* dilanjutkan dengan *cropping* dan *edge detection* yang digunakan untuk menentukan garis batas objek, hasil citra tersebut digunakan untuk data input dalam proses *train* dan *test* menggunakan

Backpropagation dengan jumlah data 200 citra tanda tangan yang terdiri dari 10 subjek, yang terbagi menjadi 180 data latih dan 20 data uji. Hasil pengujian dilakukan mendapatkan nilai dalam proses klasifikasi adalah 75% pelatihan = 88,1%, 15% *validation* = 60,3%, 15% *test* adalah 59.5% dan 100% *all* adalah 79.7%. pengujian menggunakan 2 sampel setiap kelasnya dengan total data uji adalah 20 dengan tingkat keberhasilan pada proses pengujian adalah 65%. (Dwi Putriana et al., n.d.)

c. Barry Ceasar Octariadi, Yulrio Brianorman dengan judul penelitian "Pengenalan Pola Tanda Tangan Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation" penelitian ini memiliki tahapan pra-pemrosesan dilakukan *resize* dengan ukuran 100x300 piksel, kemudian dilakukan *grayscale*, *median filter,* dilanjutkan dengan metode *otsu*, *thinning,* dan *centerring* yang bertujuan untuk mendapatkan citra yang baik. Setelah dilakukan pra-pemrosesan dilanjutkan ekstraksi fitur berdasarkan *feature feature points based vertical splitting* dan *horizontal splitting, Feature points based on vertical grid and horizontal grid*. Pada penelitian ini menggunakan data latih sebanyak 100 data *input* dan 100 data untuk uji pada setiap sampel tanda tangan dengan hasil pengujian sebesar 74% dalam mengidentifikasi keaslian tanda tangan (Octariadi, 2020)

d. Gresiva Devi Angel, Resty Wulanningrum dengan judul penelitian "Machine Learning untuk Identifikasi Tanda Tangan Menggunakan GLCM dan Euclidean Distance" penelitian ini memiliki tahapan pra-pemrosesan hanya menggunakan *cropping* dengan ukuran 100x100 piksel dan 50x50 piksel dan menggunakan *grayscale* dilanjurkan ekstraksi fitur menggunakan GLCM dengan jumlah citra tanda tangan dari 10 mahasiswa dengan setiap mahasiswa 10 tanda tangan. Terdapat beberapa pengujian yang dilakukan pada citra berukuran 50x50 piksel dan 100x100 piksel dengan uji coba pertama pada 50x50 piksel menggunakan 60 data latih dan 40 data uji menghasilkan akurasi 65%, 40 data uji dan 60 data latih akurasi 14%, 20 data uji dan 80 data latih akurasi 10%. Sedangkan pada 100x100 dengan 60 data latih dan 40 data uji menghasilkan akurasi 67,5%, 40 data

latih dan 60 data uji akurasi 67%, 20 data latih dan 80 data uji akurasi 57,5% peneliti menyimpulkan bahwa pengaruh pembagian data set dapat mempengaruhi hasil akurasi yang didapatkan(Angel & Wulanningrum, n.d.)

e. Dwi Retnoningrum, Agus Wahyu Widodo, Muh. Arif Rahman dengan judul penelitian "Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)" penelitian ini memiliki tahapan *Pre-processing* yaitu untuk mengubah citra berwarna menjadi citra ke abuan *(Grayscale)* dilanjutkan proses *Regioning* atau membagi citra menjadi beberapa *Region* dan dilakukan ekstraksi fitur *Local Binary Pattern* (LBP). Terdapat beberapa hasil akurasi dengan parameter yang berbeda-beda yang dapat mempengaruhi hasil akurasi yang diperoleh. Hasil akurasi terbaik adalah 92.31% dengan jarak ketetanggaan atau radius = 2 dengan jumlah tetangga yang dibandingkan = 8, jumlah *Region* sebesar = 16 dan jumlah pembagian height = 4 dan width = 4(Retnoningrum et al., 2019).
f. M.Fauzi Ishak, Rita Purnamasari, Murnisari Dardjan dengan judul penelitian "Identifikasi Jenis Kelamin Berdasarkan Teraan Gigitan dengan Metode LBP dan Klasifikasi LVQ" pada penelitian ini sistem mengidentifikasi berdasarkan pola bite mark atau pola gigitan manusia menggunakan *Local Binary Pattern* (LBP) dan klasifikasi menggunakan *Learning Vector Quantization* (LVQ) pengujian dilakukan dengan berbagai parameter LBP *sampling point* dan *radius* yang berbeda-beda. Pada percobaan pertama dilakukan dengan *sampling point* (P) = 4 dengan menggunakan *radius* (R) = 1 - 5, didapatkan nilai akurasi tertinggi pada *radius* (R) = 5 sebesar 94,58% dengan waktu komputasi 88,180 detik. Pada percobaan kedua dilakukan dengan *sampling point* (P) = 8 dengan menggunakan *radius* (R) = 1 - 5 menghasilkan akurasi tertinggi sebesar 98,02% dengan waktu komputasi 93,122 detik pada *radius* (R) = 4. Meningkatnya hasil akurasi disebabkan oleh nilai *sampling point* (P), bila semakin besar nilai *sampling point* (P) maka akan meningkatkan informasi mengenai tekstur yang didapatkan(Nasional et al., n.d.). Menyimpulkan bahwa pengenalan pola menggunakan metode *Local Binary Pattern* (LBP)

dengan parameter yang telah ditingkatkan akan menghasilkan nilai akurasi yang tinggi dengan waktu komputasi yang cukup cepat.

g. Afifah Amatulla Suaib, Iwan Iwut Tritosmoro, Nur Ibrahim dengan judul penelitian "Identifikasi COVID-19 berdasarkan citra X-ray paru-paru menggunakan metode *Local Binary Pattern* (LBP) dan algoritma Random Forest" pada penelitian ini telah menyimpulkan berdasarkan hasil dua uji coba yang terdiri dari, pengujian pertama dilakukan dengan menguji pengaruh *Resize* tahapan *Pre-processing* pada tingkat akurasi yang dihasilkan dan pengujian kedua dilakukan dengan menguji beberapa parameter *radius* (R). Pada hasil pengujian pertama citra *Resize* dengan ukuran 200x200 piksel mendapatkan tingkat akurasi tertinggi sebesar 63,67%, sedangkan untuk citra *Resize* dengan ukuran 100x100 piksel memiliki tingkat akurasi terendah sebesar 57%. Pada pengujian kedua peneliti menggunakan citra dengan ukuran 200x200 piksel menghasilkan tingkat akurasi tertinggi pada *radius* (R) = 8 sebesar 84,67% dan akurasi terendah sebesar 63.67% pada *radius* (R) = 1. Bila nilai *radius* dinaikkan dan melebihi 8 tingkat akurasi akan menurun(Suaib et al., 2022). Menyimpulkan bahwa penelitian ini penggunaan *Resize* pada citra yang lebih besar dan lebih kecil tidak terlalu berpengaruh dalam hasil tingkat akurasi menggunakan metode ekstraksi fitur *Local Binary Pattern* (LBP), dan pengujian kedua dapat disimpulkan bila nilai *radius* yang terlalu besar dapat menurunkan hasil tingkat akurasi disebabkan oleh sedikitnya ciri karakteristik yang diperoleh dari citra yang akan diambil ciri karakteristiknya. Hal ini dapat menunjukkan bahwa tingkat *radius* dapat mempengaruhi hasil akurasi sistem.

h. Dwi Rizki Yulianti, Iwan Iwut Triastomoro, Sofia Sa'idah dengan judul penelitian "Identifikasi Pengenalan Wajah dengan Menggunakan Metode KNN (K-NEAREST NEIGHBOR) dan LBPH (LOCAL BINARY PATTERN HISTOGRAM) Untuk Sistem Presensi" pada penelitian ini hanya menggunakan pra-pemrosesan *grayscale* dilanjutkan ekstraksi fitur menggunakan LBPH yang termasuk metode baru dalam LBP. Penelitian

ini menggunakan 100 data citra wajah dari beberapa posisi depan, atas, bawah, kanan dan kiri. Dilakukan beberapa pengujian dengan menggunakan parameter (*radius*= 1, *neighbors*= 8, *Grid* X, *Grid* Y= 8x8) dengan posisi wajah (depan, atas (mendongak), bawah (menunduk), kanan dan kiri) dengan jarak sejauh 30 cm mendapatkan akurasi rata-rata 85,32%, nilai FAR rata-rata 8,66% dan nilai FRR rata-rata 6%. Klasifikasi KNN dengan parameter k = 1 akurasi 100% dengan waktu komputasi 34 ms, pada saat k = 3 akurasi 98% dengan waktu komputasi 37 ms dan pada saat k = 5 akurasi 88% dengan waktu komputasi 42 ms. Peneliti menyimpulkan Semakin kecil nilai yang digunakan, semakin besar akurasi yang didapatkan untuk klasifikasi (Yulianti et al., 2022)

i. Andi Farmadi, Ahmad Faris Asy'arie, Irwan Budiman, Dwi Kartini, Ahmad Rusadi Arrahimi, Muliadi dengan judul penelitian "Signature Identification Menggunakan Metode Template Matchingdan Fuzzy K-Nearest Neighbor" pada penelitian ini memiliki tahapan pra-pemrosesan dengan melakukan *cropping* dan format citra yang digunakan semula adalah JPG diubah menjadi PNG yang berguna untuk optimalisasi citra tanpa mengurangi kualitas citra pada citra yang telah dilakukan *cropping,* kemudian dilakukan *resize* menjadi 200x200 piksel dan tahap terakhir dilakukan *grayscale*. Setelah pra-pemrosesan dilakukan *thresholding* digunakan untuk memisahkan objek citra dengan background pada gambar dengan nilai *thresholding* yang digunakan ialah 128. Pada penelitian ini menggunakan data latih 100 citra tanda tangan asli, untuk data uji menggunakan 110 data tanda tangan yang terdiri dari 100 data tanda tangan asli dan 10 data tanda tangan palsu. Terdapat beberapa pengujian yang dilakukan dan peneliti menyimpulkan k=3 untuk tanda tangan asli memiliki nilai kelas pemilik tanda tangan teridentifikasi sebesar 92%, False Rejection Rate (FRR) 12%, dan nilai Positive Predictive Value (PPV) 88% lebih tinggi dari k=5 dan k=9. Serta dipilih k=9 untuk tanda tangan palsu karena memiliki nilai kelas teridentifikasi sebesar 90%, False Acceptance Rate (FAR) 10%, dan memiliki nilai Negative Predictive

Value (NPV) 90% lebih tinggi dari k=3 dan k=5, untuk ukuran 200x200 pixel(Farmadi et al., 2021)

j. Mutiara S. Simanjuntak, Rika Rosnelly, Wanayumini dengan judul penelitian "Identifikasi Tanda Tangan Menggunakan Metode Fitur Ekstraksi Biner dan K-Nearest Neighbor" pada penelitian ini melakukan perbandingan antara fitur ekstrasi biner *Local Binary Pattern* (LBP) dan *Binarised Statistical Image Features* (BSIF) serta menggunakan klasifikasi *K-Nearest Neighbor* (KNN) pada penelitian ini menggunakan dua database yang berisi 100 dan 75 individu. Percobaan dilakukan dengan membandingkan antara metode ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) dan *Binarised Statistical Image Features* (BSIF) dengan melakukan dua percobaan pada database yang berbeda. Percobaan pertama pada data base pertama MCYT-75 menunjukkan bahwa *Binarised Statistical Image Features* (BSIF) memberikan hasil yang lebih baik dari pada LBP. Pada percobaan kedua menggunakan data base kedua GPDS-100 yang berfokus pada pencarian parameter terbaik menunjukkan hasil bahwa metode ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) mendapatkan hasil yang lebih tinggi dari pada *Binarised Statistical Image Features* (BSIF) dengan hasil terbaik mencapai 98,3%(SIMANJUNTAK, 2021). Menyimpulkan bahwa metode ekstraksi fitur menggunakan *Local Binary Pattern* (LBP) lebih baik dari pada menggunakan metode *Binarised Statistical Image Features* (BSIF) pada parameter tertentu dengan menggunakan klasifikasi KNN.

Berdasarkan penelitian yang telah dilakukan, LBP dapat mengekstraksi fitur tanda tangan dan memberikan hasil identifikasi yang cukup baik dengan menggunakan mesin pembelajaran KNN. Untuk itu, penelitian ini menyarankan penggunaan metode ekstraksi fitur Uniform *Local Binary Pattern* (ULBP) dan klasifikasi dengan menggunakan metode *K-Nearest Neighbors* (KNN) untuk identifikasi keaslian tanda tangan. Hasil analisis dari penelitian terdahulu menyimpulkan bahwa penggunaan metode tersebut dapat bekerja dengan baik untuk mengidentifikasi keaslian tanda tangan

dengan menerapkan parameter yang sesuai dengan kondisi atau keadaan citra yang akan diidentifikasi.

2.2 Landasan Teori

2.2.1 Biometrik Tanda Tangan

Pada dasarnya manusia memiliki karakteristik yang terbagi menjadi dua yaitu karakteristik secara *fisiologi* atau fisik *(physiological/physical characteristic)* dan karakteristik perilaku *(behavioral characteristic).* Biometrika berdasarkan karakteristik *fisiologi* atau fisik menggunakan bagian struktur fisik dari tubuh sebagai kode atau pola unik untuk pengenalan setiap individu, seperti telinga, flush atau jejak panas pada wajah manusia, *Deoxyribo Nucleic Acid* (DNA), serta geometri pada retina, tangan, dan bau atau komposisi kimia dari keringat yang dihasilkan tubuh. Untuk biometrika karakteristik perilaku menggunakan perilaku manusia sebagai kode atau pola unik untuk melakukan pengenalan pada seseorang, seperti gaya berjalan, suara, hentakan tombol dan tanda tangan. Secara umum sistem biometrika terbagi menjadi dua model, yaitu sistem identifikasi *(Identification System)* dan sistem verifikasi *(Verification System)*. Sistem identifikasi digunakan untuk tujuan memecahkan identitas seseorang, sedangkan untuk sistem verifikasi memiliki tujuan untuk menerima atau menolak identitas yang telah diklaim seseorang(Haryadi, n.d.).

2.2.2 Citra Digital

Citra digital adalah representasi dari suatu gambar dua dimensi yang menggunakan jumlah point terbatas yang disebut sebagai elemen gambar atau piksel. Setiap piksel pada suatu citra diwakili oleh satu atau lebih nilai numerik, seperti pada gambar monokrom *(graysacale)* memiliki nilai tunggal yang mempresentasikan kecerahan atau intensitas dari warna abu-abu, secara umum nilai-nilai intensitas ini memiliki rentan 0 mewakili warna hitam hingga 255 mewakili warna putih, sebagai contoh nilai 0 menghasilkan piksel yang lebih gelap atau berwarna hitam, sementara untuk nilai 255

menghasilkan piksel yang memiliki intensitas kecerahan yang tinggi atau berwarna putih. Secara umum gambar monokrom *(graysacale)* juga menggabungkan tiga nilai citra warna yang merepresentasikan jumlah warna merah (R), hijau (G), dan biru (B) dengan tingkat nilai yang sama dapat menghasilkan gambar atau citra yang tidak memiliki warna atau hanya memiliki intensitas yang berbeda-beda pada setiap pikselnya. Pada definisi lain citra digital adalah suatu matriks dua dimensi yang terdiri dari baris dan kolom, setiap elemen dalam matriks mewakili titik tunggal pada citra yang disebut dengan piksel atau *(picture element)*(Kadar et al., 2020).

Citra digital merupakan representasi dari citra analog *continue* yang telah diubah ke dalam bentuk diskret, secara dasar citra digital menggambarkan representasi diskrit dari citra analog *continue.* citra analog *continue* didefinisikan sebagai fungsi dua dimensi *f(x, y),* di mana *x* dan *y* adalah koordinat *special* di dalam citra, dan nilai atau amplitudo dari fungsi f pada setiap titik koordinat *(x, y)* disebut dengan intensitas atau tingkat keabuan *(gray level)* di level tertentu. Ketika *(x, y)* dan nilai intensitas dari semua f terbatas serta nilainya diskrit maka disebut sebagai citra digital(Gonzalez & Woods, 2018).

2.2.3 *Grayscale*

Citra *grayscale* merupakan citra berwarna keabuan dengan memiliki variasi warna 8 bit serta disebut dengan skala keabuan dikarenakan pada umumnya di dalam warna keabuan terdapat hitam sebagai warna minimal dan putih sebagai warna maksimal, sehingga terdapat warna di antara keduanya adalah abu-abu. Namun pada implementasinya warna yang terpakai tidak terbatas pada warna abu-abu sebagai contoh dipilih warna minimal adalah putih dan warna maksimal adalah merah, maka semakin besar nilai semakin besar juga intensitas warna merahnya. Untuk mengonversi citra RGB ke dalam citra *grayscale* dilakukan menggunakan rumus persamaan 2.1

$$GS_{(x,y)} = 0.2989 * R_{(x,y)} + 0.5870 * G_{(x,y)} + 0.1141 * B_{(x,y)} \quad (2.1)$$

2.2.4 Ekstraksi Fitur

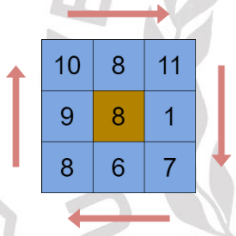
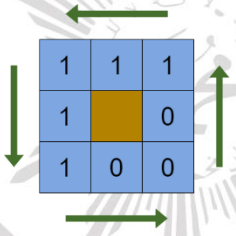
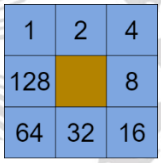
Fitur adalah karakteristik unik dari sebuah objek di dalam citra yang akan dibedakan dengan objek lainya. Informasi yang telah diekstrak menghasilkan parameter yang dapat membedakan sebuah objek pada tahapan identifikasi atau klasifikasi, serta terdapat beberapa fitur-fitur yang diamati adalah fitur warna, fitur tekstur, dan fitur bentuk. Fitur-fitur tersebut dapat dikenali dan dibedakan karena memiliki karakteristik atau pola-pola tertentu(Kosasih et al., 2021). Fitur memiliki susunan bilangan yang dapat dipakai dalam mengidentifikasi objek. Susunan bilangan adalah hasil *descriptor,* sedangkan *descriptor* merupakan parameter yang memiliki karakteristik tertentu pada objek. Untuk mengenali objek pada citra membutuhkan parameter dan karakteristik yang bisa mencirikan objek tanda tangan. Ekstraksi fitur adalah proses yang digunakan untuk mengambil ciri dan karakteristik dari objek tanda tangan yang digunakan sebagai pembeda dari objek tanda tangan lainya.

2.1.4.1 Ekstraksi Fitur Tekstur

Ekstraksi fitur tekstur adalah salah satu aspek atau fitur penting dalam sebuah gambar, yang memiliki informasi mengenai pola susunan struktur yang ada di permukaan gambar. Proses ekstraksi fitur tekstur umumnya menggunakan metode *Local Binary Pattern* (LBP) atau *Gray Level Co-occurrence Matrix* (GLCM), di mana fungsi ini berguna sebagai matriks yang mengekstrak nilai keabuan dari gambar(Amatullah et al., 2021). Ekstraksi fitur tekstur pada dasarnya merupakan proses untuk mengidentifikasi, mengekstrak, dan merepresentasikan informasi mengenai pola, struktur, dan tekstur pada sebuah gambar. Serta berfungsi untuk algoritma pengolahan citra untuk memahami ciri tekstur pada gambar, seperti pola halus, kasar, atau berulang yang mungkin sulit untuk didapatkan menggunakan metode ekstraksi fitur warna maupun bentuk.

2.2.5 *Local Binary Pattern* (LBP)

Local Binary Pattern (LBP) merupakan salah satu metode yang dipakai untuk ekstraksi fitur tekstur yang telah terbukti efektif serta dalam invariant terhadap cahaya yang berbeda. Penggunaan *Local Binary Pattern* (LBP) untuk mencari pola-pola tekstur pada citra dan nilai *Local Binary Pattern* (LBP) didapatkan dengan melakukan *thresholding* pada piksel tetangga dengan menggunakan nilai pusat. *Local Binary Pattern* (LBP) pada dasarnya bekerja dengan delapan piksel tetangga dengan pusat piksel di tengah dan menggunakan nilai pusat piksel sebagai *threshold* yang didapatkan dengan memberikan label pada setiap piksel tetangga. Jika nilai tetangga lebih kecil dari nilai pusat piksel maka nilai piksel 0, jika nilai tetangga lebih besar dari nilai pusat piksel maka nilai piksel(Rahayu et al., 2021).

Operasi Thresholding	Hasil Thresholding	Mapping	Nilai baru (Desimal)
	 11000111		= 1*1 + 1*2 + 1*4 + 0*8 + 0*16 + 0*32 + 1*64 + 1*128 = 199

Gambar 2.1 Ilustrasi tahapan perhitungan LBP

Proses kerja perhitungan *Local Binary Pattern* (LBP) adalah untuk mencari nilai pusat atau tengah dari suatu kernel, seperti gambar 2.1 memperlihatkan contoh kernel berukuran 3 x 3 dengan nilai titik pusat kernel adalah 8 pada blok berwarna kuning, maka akan dilakukan perhitungan *Local Binary Pattern* (LBP) yang akan menggantikan nilai pusat kernel 8 dengan nilai baru.

$$LBP_{P,R} = \sum_{p-0}^{p-1} s(g_{p-} g_e)2^p \qquad (2.2)$$

Tahap pertama perhitungan *Local Binary Pattern* (LBP) dengan melakukan perbandingan nilai tetangga piksel terdekat pada nilai tengah piksel atau nilai titik pusat kernel pada citra hasil pra-pemrosesan dengan nilai *grayscale*.

$$s(x) = \begin{cases} 0, \ jika\ x < 0 \\ 1, \ jika\ x \geq 0 \end{cases} \quad (2.3)$$

Pada proses perbandingan nilai *grayscale,* apabila nilai tetangga piksel lebih besar atau sama dengan nilai pusat piksel, maka hasil perbandingan akan diberikan nilai 1 pada nilai tetangga piksel dan sebaliknya apabila nilai tetangga piksel lebih kecil dari nilai pusat piksel maka hasil perbandingan akan diberikan nilai 0 pada nilai tetangga piksel. Proses ini dilakukan secara berurutan dimulai dari pojok kiri atas pada (perbandingan nilai pusat 8 dan nilai tetangga 10) dan seterusnya mengikuti perputaran arah jarum jam (dilihat pada panah merah) atau sebaliknya berlawanan dengan arah jarum jam.

Proses perbandingan antara nilai tetangga piksel dengan nilai pusat piksel disebut dengan nama *thresholding.* Proses *thresholding* menghasilkan nilai-nilai piksel antara 0 – 1. Setelah dilakukan proses *thresholding,* maka nilai piksel tetangga akan diambil sebagai nilai biner. Pengambilan nilai piksel tetangga dimulai dari posisi terakhir dan akan bergerak mundur sampai ke posisi awal, berlawan dengan arah urutan posisi saat proses *thresholding* (ditunjukkan pada panah hijau) . Hasil biner kemudian diubah menjadi angka desimal, setelah itu angka desimal akan digunakan untuk menggantikan nilai intensitas piksel pada nilai titik pusat kernel.

Terdapat cara lain untuk mendapatkan nilai *Local Binary Pattern* (LBP) dengan menggunakan *mapping*. Tahap awal proses *mapping* dengan mendapatkan matriks *mapping* yang diperoleh dengan menghitung nilai 2 pangkat n, dengan n berada dalam rentang 0, 1, … , hingga jumlah tetangga (*sample*). Jumlah tetangga di sini sesuai dengan jumlah piksel tetangga yang digunakan dalam proses *Local Binary Pattern* (LBP). Setelah mendapatkan matriks mapping, proses mapping akan dilakukan dengan mengalikan hasil dari *thresholding* (metode untuk mengonversi nilai piksel menjadi nilai biner, seperti 0 atau 1) dengan matriks *mapping* yang sesuai dengan posisi. Hasil

perkalian tersebut akan dijumlahkan menghasilkan nilai desimal yang menjadi nilai baru untuk titik pusat kernel (kotak kuning) dan sebagai nilai *Local Binary Pattern* (LBP) untuk piksel yang diproses. Hasil nilai ini merepresentasikan pola tekstur pada citra

2.2.6 *Uniform Local Binary Pattern* (ULBP)

Uniform Local Binary Pattern (ULBP) merupakan salah satu varian dari *Local Binary Pattern* (LBP) yang hanya mengambil *output* uniform. *Uniform Local Binary Pattern* (ULBP) sendiri adalah pola dengan maksimal dua transisi *bitwise* 0-1 dan 1-0 dengan memilihnya pola uniform dapat berkontribusi dalam mengurangi panjang vektor fitur dan meningkatkan kinerja klasifikasi yang menggunakan fitur *Local Binary Pattern* (LBP)(Nyoman Indra et al., n.d.). Pada *Local Binary Pattern* (LBP) dengan tetangga sebanyak 8 dengan radius bernilai 1 yang memiliki kombinasi pola 256 diantara-Nya adalah pola uniform sebanyak 58, sehingga Panjang vector fitur dapat berkurang dari 256 menjadi hanya 59 fitur.

Local Binary Pattern (LBP) dengan nilai 11 memiliki pola biner (00001011), karena pola ini mengandung lebih dari dua transisi bitwise (perubahan nilai dari 0 ke 1 atau sebaliknya), maka pola biner ini tidak termasuk pola uniform atau non-uniform. Pola non-uniform seperti (00001011) akan diberikan label khusus untuk mengidentifikasi, contoh sebagai label 58, maka semua pola non-uniform akan diberikan label unik, misalnya label 58 untuk membedakan dari pola lain. *Local Binary Pattern* (LBP) dengan nilai 7 memiliki pola biner (00000111), karena pola ini hanya mengandung kurang dari dua transisi bitwise, maka pola ini dikategorikan sebagai pola uniform. Pola-pola uniform dengan pola biner (00000111) akan diberikan label yang sesuai dengan urutan dalam kategori uniform, contoh urutan 56 dari 0 hingga 57 label pola uniform.

2.2.7 Klasifikasi

Klasifikasi adalah suatu proses bertujuan untuk mengelompokkan atau menempatkan sebuah objek ke dalam kelas atau kategori yang telah ditentukan sebelumnya. Klasifikasi juga didefinisikan sebagai proses pembangunan model yang digunakan untuk mengklasifikasikan objek sesuai dengan atribut-atributnya. Proses klasifikasi data maupun dokumen dapat dimulai dengan membangun aturan klasifikasi tertentu yang menggunakan data latihan atau yang sering disebut sebagai tahapan pembelajaran, setelah aturan telah dibangun maka data pengujian atau data testing digunakan untuk menguji model yang telah dibuat(Setiawan et al., 2018).

Pada dasarnya, langkah klasifikasi melibatkan penggunaan atribut-atribut yang dimiliki oleh objek atau citra untuk dapat memutuskan kelas atau kategori yang memiliki tingkat kemiripan atau kecocokan tertinggi yang sesuai dengan objek atau citra tersebut. Melibatkan penggunaan model pembelajaran yang sesuai dengan yang diperlukan, serta model atau algoritma yang dipelajari dari data latihan atau data *training* untuk digunakan dalam mengklasifikasikan data baru.

2.2.8 *K-Nearest Neighbors* (KNN)

Klasifikasi *K-Nearest Neighbor* (KNN) merupakan metode pembelajaran improvisasi yang di mana hanya dilakukan pada saat data uji diprediksi serta sebuah metode klasifikasi yang berdasarkan data pembelajaran yang memiliki jarak paling dekat antara objek. Prinsip dasar dari *K-Nearest Neighbor* (KNN) untuk mencari jarak terdekat antara data baru yang akan dievaluasi dengan K ketetanggan terdekat dalam data pembelajaran(Zhang & Li, 2023). *K-Nearest Neighbor* (KNN) merupakan salah satu metode sederhana dalam pembelajaran dan disebut (*lazy learning*). Metode ini digunakan dalam bidang pengenalan karakter, klasifikasi teks, dan pengenalan gambar dengan memberikan data uji, algoritma hanya perlu memprediksi keadaan data uji berdasarkan keadaan K tetangga terdekatnya dalam data pelatihan. KNN menggunakan pembelajaran supervised.

Supervised learning digunakan untuk menemukan pola baru dalam sebuah data dengan menghubungkan pola data yang sudah ada dengan data yang baru(Li et al., 2023).

Algoritma *K–Nearest Neighbor* (KNN) secara umum nilai K yang tinggi dapat mengurangi *effect noise* pada saat klasifikasi, tetapi dapat membuat batasan antara setiap klasifikasi menjadi lebih kabur karena mempertimbangkan lebih banyak ketetanggaan. Sedangkan nilai K yang baik didapatkan melalui optimasi parameter dengan menggunakan *cross-validation* yang di mana klasifikasi diprediksi berdasarkan data pembelajaran yang paling dekat (K = 1) disebut algoritma *k-nearest neighbor*. Digunakan rumus pada persamaan 2.4.

$$f(v_{1,}v_2) = \sqrt{\sum_{k=1}^{N}(v_1(k) - v_2(k))^2} \qquad (2.4)$$

Keterangan persamaan di atas:

$f(v_{1,}v_2)$ = Jarak *Euclidean*

N = Jumlah *sample*

v_1 = Proses perhitungan data v ke-1

v_2 = Proses perhitungan data v ke-2

Pada persamaan di atas adalah persamaan jarak *Euclidean* yang akan digunakan dalam metode KNN. Jarak *Euclidean* digunakan untuk melakukan klasifikasi berdasarkan jarak sebuah data dengan data lainya. Jarak akan ditentukan berdasarkan nilai K ketetanggaan. Nilai K sendiri diperoleh secara *random* agar hasil akhir memiliki *presentase* yang dinilai tinggi.

2.2.9 *Confusion Matrix*

Confusion Matrix merupakan sebuah analisis *predictif* yang digunakan untuk menampilkan serta membandingkan antara nilai aktual dengan nilai hasil prediksi model yang digunakan untuk mendapatkan hasil matrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *Error rate.* Dalam tabel *Confusion Matrix* menghasilkan 4 nilai yaitu, *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN).

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

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.

a. *Accuracy*

Akurasi didapatkan melalui jumlah data yang bernilai positif yang diprediksi positif *True Positive* (TP) dan data yang bernilai negatif yang diprediksi negatif *True Negative* (TN) dibagi dengan seluruh jumlah data menggunakan rumus persamaan 2.5.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.5)$$

b. *Precision*

Precision merupakan peluang kasus yang diprediksi positif yang dalam kenyataannya termasuk kasus kategori positif *True Positive* (TP) menggunakana rumus persamaan 2.6.

$$Precision = \frac{TP}{TP+FP} \quad (2.6)$$

c. *Recall*

Recall merupakan peluang kasus dengan positif yang dengan benar prediksi positif menggunakan rumus persamaan 2.7.

$$Recall = \frac{TP}{TP+FN} \quad (2.7)$$

d. *Error Rate*

Error Rate adalah presentase jumlah *record* data yang diprediksi secara salah oleh algoritma yang digunakan menggunakan rumus persamaan 2.8.

$$Error\ Rate = \frac{(FN+FP)}{(TP+FN+FP+TN)} \quad (2.8)$$

2.2.10 Matlab

Matlab adalah salah satu program komputer yang berguna sebagai "laboratorium" untuk melakukan perhitungan matriks dan memberikan akses ke perangkat lunak matriks yang telah dikembangkan oleh LINPACK dan EISPACK, program ini memiliki kemampuan untuk menyelesaikan tugas-tugas standar, seperti persamaan linier hingga permasalahan nilai eigen pada matriks serta digunakan dalam aljabar linier dan analisis numerik, dan bermanfaat sebagai kalkulator matriks. Matlab dikembangkan menggunakan bahasa FORTRAN dan dirancang untuk dapat diistal dengan mudah pada sistem operasi mana pun yang memungkinkan interaktif pada bahasa-bahasa FORTRAN. Terdapat sekitar 6000 basis kode sumber FORTRAN, termasuk didalam-Nya subtruin LINPACK dan EISPACK yang digunakan, meskipun matlab menggunakan referensi dari perangkat lunak matriks LINPACK dan EISPACK, maltab lebih memprioritaskan kemudahan pada pengguna daripada efisiensi waktu eksekusi maupun penghematan penyimpanan. Oleh sebab itu, matlab hanya menggunakan sebagian kecil dari subrutin yang dimiliki oleh LINPACK dan EISPACK(Moler & Little, 2020).

LINPACK dan EISPACK adalah teknologi terkini dalam perangkat lunak untuk pengelolaan komputasi matriks. EISPACK adalah suatu paket yang berisi lebih dari 70 subrutin FORTRAN untuk digunakan di berbagai komputasi nilai eigen matriks yang sebagian besar berdasarkan pada prosedur-prosedur algol yang diterbitkan oleh Wilkinson dan Reinsch pada tahun 1971, LINPACK adalah suatu paket yang memiliki 40 subrutin FORTRAN yang masing-masing memiliki empat tipe data untuk dapat

menyelesaikan dan menganalisis persamaan linier simultan dan masalah-masalah matriks lainya. Matlab tidak menggunakan keseluruhan subrutin FORTRAN, dan hanya menggunakan 8 subrutin dari LINPACK dan 5 dari EISPACK yang terlibat pada matlab(Moler & Little, 2020).

2.1.10.1 LINPACK

Linear Equation Package (LINPACK) adalah sebuah perangkat lunak matematika yang dikembangkan pada tahun 1970 oleh Jack Dongarra, Pete Stewart, Jim Bunch, dan Cleve Moler dengan tujuan untuk menyelesaikan sistem persamaan linier dengan presisi yang berbeda-beda meliputi, REAL, DOUBLE, COMPLEX, Dan COMPLEX*16. Pengembangan *Linear Equation Package* (LINPACK) dilakukan di Laboratorium Nasional Argonne dan *National Science Foundation* (NSF) Amerika Serikat. *Linear Equation Package* (LINPACK) dikembangkan langsung menggunakan bahasa pemrograman FORTRAN. *Linear Equation Package* (LINPACK) memiliki 44 subrutin dalam setiap presisinya serta di implementasikan dengan gaya pemrograman yang terstruktur. Pengembangan *Linear Equation Package* (LINPACK) menghindari penggunaan "go-to's" dan nomor pernyataan untuk meningkatkan kejelasan dalam keterbacaan kode, serta menunjukkan dengan jelas ruang lingkup dari perulangan dan konstruksi if-then-else dengan indentis yang teratur(Moler & Little, 2020).

Aspek penting dalam *Linear Equation Package* (LINPACK) adalah penggunaan *Basic Linear Algebra Subprogram* (BLAS) yang telah dikembangkan secara bersamaan oleh Chuck Lawson yang memfasilitasi perulangan-perulangan penting pada perkalian dalam *(inner product)* dari dua vektor (DDOT) dan penjumlahan vektor dengan scalar (DAXPY). Implementasi *Basic Linear Algebra Subprogram* (BLAS) dapat memungkinkan kinerja yang lebih

efisien pada sistem yang tidak memiliki kompilator FORTRAN yang dioptimalkan(Moler & Little, 2020).

2.1.10.2 EISPACK

Matrix Eigensystem Package (EISPACK) merupakan perangkat lunak matematika yang telah dikembangkan pada awal tahun 1970. Perangkat lunak diciptakan sebagai hasil dai usaha bersama yang dilakukan di Laboratorium Nasional Argonne dan *National Science Foundation* (NSF) tujuan utama dikembangkannya *Matrix Eigensystem Package* (EISPACK) untuk menghasilkan, menguji dan menyebarkan perangkat lunak yang berkualitas tinggi untuk menyelesaikan masalah-masalah yang berkaitan dengan nilai eigen matriks. Pengembangan *Matrix Eigensystem Package* (EISPACK) dimulai dengan penerjemahan prosedur-prosedur dalam bahasa pemrograman algoritma yang membahas mengenai masalah nilai eigen dari bagian II Handbook ke dalam Bahasa FORTRAN serta melakukan pengujian menyeluruh untuk memastikan kemampuan probabilitas(Moler & Little, 2020).

Perilisan pertama *Matrix Eigensystem Package* (EISPACK) dilakukan pada tahun 1971 ke dua universitas dan laboratorium nasional Amerika Serikat yang telah menyetujui dan sepakat untuk menguji dan menganalisis kembali perangkat lunak *Matrix Eigensystem Package* (EISPACK). Proyek terus dikembangkan, perilisan kedua *Matrix Eigensystem Package* (EISPACK) sudah pada tahap dapat didistribusikan secara publik. Pada perangkat lunak *Matrix Eigensystem Package* (EISPACK) inklusinya tidak hanya pada prosedur-prosedur dari bagian II Handbook, tetapi juga pada algoritma *Singular Value Decomposition* (SVD) dari bagian I serta algortima QZ dalam masalah nilai eigen generalisasi yang melibatkan dua matriks secara langsung(Moler & Little, 2020).