

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Metode yang digunakan pada langkah awal penelitian tugas akhir ini dilakukan dengan memulai studi literatur hal-hal yang memiliki kaitan dengan sistem penelitian yang dilakukan.

Pada penelitian [1] membahas tentang pengenalan pola tanda tangan dengan menggunakan metode *Principal Component Analysis* (PCA). Tanda tangan adalah salah satu bentuk identifikasi yang umum digunakan dalam berbagai aplikasi seperti verifikasi dokumen, keamanan, dan transaksi elektronik. Tujuan dari penelitian ini adalah mengembangkan sebuah sistem pengenalan pola tanda tangan yang efektif dan efisien menggunakan metode PCA. Pertama, dataset tanda tangan dikumpulkan dan diproses untuk menghilangkan derau dan memperbaiki kualitas gambar. Kemudian, metode PCA diterapkan untuk mengurangi dimensi dataset tanda tangan. PCA mengidentifikasi kombinasi linier dari fitur-fitur awal yang paling signifikan dan menghasilkan vektor-vektor eigen yang merepresentasikan varian data yang paling tinggi. Vektor-vektor eigen ini digunakan sebagai representasi fitur tanda tangan. Selanjutnya, dilakukan tahap pelatihan menggunakan algoritma klasifikasi seperti *Support Vector Machine* (SVM) atau *Neural Network* (NN) untuk mengenali pola tanda tangan. Selama tahap ini, model dilatih menggunakan contoh-contoh tanda tangan yang sudah diketahui dan diberi label. Proses ini melibatkan komputasi yang intensif dan pemilihan parameter yang tepat untuk mencapai performa yang baik. Sistem pengenalan pola tanda tangan yang telah dibuat dalam penelitian ini mampu mendeteksi pola tanda tangan dengan tingkat akurasi lebih dari 70%. desain penelitian ini mengacu pada tahapan proses yang ada pada sistem pengenalan tanda tangan, yaitu input, proses, dan output. Tahapan penelitian dilakukan mengikuti desain seperti pada gambar yang disajikan. penelitian ini dilakukan dengan menggunakan 300 dataset yang masing-masing berjumlah 200 citra data training dan 100 citra yang diuji. Pengujian ini dilakukan untuk mengetahui akurasi sistem pengenalan tanda tangan dengan menggunakan jumlah data training dan data yang diuji yang berbeda. Hasil pengujian

menunjukkan bahwa sistem pengenalan tanda tangan memiliki tingkat akurasi yang bervariasi tergantung pada jumlah data training dan data yang diuji yang digunakan.

Selanjutnya hasil penelitian [2] membahas tentang implementasi algoritma *Principal Component Analysis* (PCA) pada pengenalan pola tanda tangan dengan menggunakan bahasa pemrograman Python. Dalam konteks pengenalan pola tanda tangan, PCA digunakan untuk mengekstraksi fitur-fitur penting dari tanda tangan yang kemudian digunakan untuk membedakan antara tanda tangan yang berbeda. Dalam jurnal ini, penulis menjelaskan langkah-langkah implementasi PCA dengan rinci, termasuk pemrosesan dataset tanda tangan, perhitungan matriks kovarian, dan pemrosesan matriks eigen. Selanjutnya, dilakukan pelatihan dan pengujian menggunakan dataset tanda tangan yang telah dikumpulkan. Pada tahap ini, model PCA dilatih menggunakan contoh-contoh tanda tangan yang sudah diketahui dan diberi label. Kemudian, pengujian dilakukan untuk menguji kemampuan model dalam mengenali pola tanda tangan yang tidak dikenal. Hasil penelitian menunjukkan bahwa implementasi algoritma PCA menggunakan bahasa pemrograman Python berhasil dalam pengenalan pola tanda tangan. Pengurangan dimensi yang dilakukan oleh PCA membantu meningkatkan efisiensi pemrosesan dan mampu membedakan tanda tangan dengan tingkat akurasi yang tinggi. Selain itu, penggunaan bahasa pemrograman Python mempermudah proses implementasi dan pengembangan sistem.

Kemudian pada [3] membahas tentang implementasi *Shape Feature* dan *K-Nearest Neighbor* (KNN) untuk klasifikasi tanda tangan. Tujuan dari penelitian ini adalah mengembangkan sebuah sistem klasifikasi tanda tangan yang efektif menggunakan kombinasi metode *Shape Feature* dan KNN. Metode *Shape Feature* fitur ini mencakup aspek bentuk, ukuran, sudut, dan properti geometris lainnya yang dapat membedakan tanda tangan satu dengan yang lain. Dalam konteks pengenalan tanda tangan, *Shape Feature* dapat digunakan untuk mengekstraksi informasi penting yang merepresentasikan karakteristik tanda tangan. Dalam konteks klasifikasi tanda tangan, KNN dapat digunakan untuk membandingkan fitur-fitur *Shape Feature* dari tanda tangan yang belum diketahui dengan tanda tangan yang sudah diketahui, dan kemudian mengklasifikasikan tanda tangan tersebut ke dalam kategori yang sesuai. Pada tahap implementasi, dataset tanda

tangan dikumpulkan dan diproses untuk menghilangkan derau dan memperbaiki kualitas gambar. Selanjutnya, metode Shape Feature diterapkan untuk mengekstraksi fitur-fitur geometris dari tanda tangan. Setelah itu, metode KNN digunakan untuk melakukan klasifikasi tanda tangan berdasarkan fitur-fitur tersebut. Shape Feature membantu dalam mengekstraksi fitur-fitur geometris yang penting, sedangkan KNN memberikan kemampuan klasifikasi yang baik berdasarkan fitur-fitur tersebut. Pengujian yang dilakukan dengan menggunakan dataset tanda tangan yang luas menunjukkan tingkat akurasi yang memuaskan dalam klasifikasi tanda tangan. Dengan melakukan dua pendekatan pemotongan citra hasil yang didapatkan tidak jauh berbeda, baik persegi atau segitiga mempunyai akurasi yang cukup bagus di atas 95%. Artinya tingkat keberhasilan metodologi yang di usulkan sangat tinggi. Hasil lebih bagus didapatkan dari pendekatan segitiga, pendekatan ini secara konsisten menghasilkan akurasi 98.25%. Saran untuk penelitian selanjutnya adalah membandingkan berbagai macam fitur ekstraksi yang telah dikerjakan pada penelitian-penelitian sebelumnya baik dengan tekstur atau dengan pembagian citra untuk membuat klasifikasi tanda tangan lebih akurat lagi.

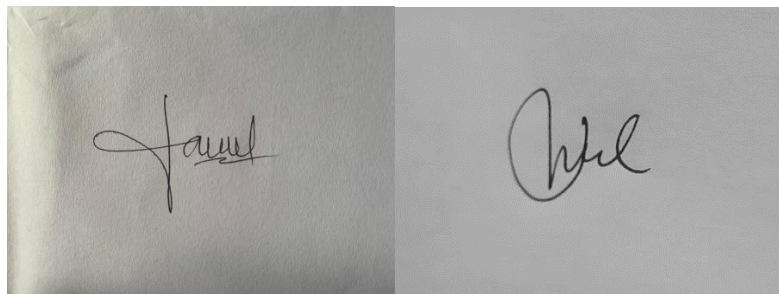
Lalu pada penelitian [4] membahas tentang pengenalan tanda tangan menggunakan metode *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (KNN). Tujuan dari penelitian ini adalah mengembangkan sebuah sistem pengenalan tanda tangan yang efektif menggunakan kombinasi metode PCA dan KNN. Metode KNN adalah algoritma klasifikasi yang sederhana namun efektif. Algoritma ini bekerja dengan mengklasifikasikan sebuah data berdasarkan mayoritas dari kategori-kategori data terdekat. Dalam konteks pengenalan tanda tangan, KNN dapat digunakan untuk membandingkan fitur-fitur tanda tangan yang diekstraksi oleh PCA dengan tanda tangan yang sudah diketahui, dan kemudian mengenali pola tanda tangan yang paling mirip. Pada tahap implementasi, dataset tanda tangan dikumpulkan dan diproses untuk menghilangkan derau dan memperbaiki kualitas gambar. Selanjutnya, metode PCA diterapkan untuk mengurangi dimensi dataset dan mengekstraksi fitur-fitur penting. Setelah itu, metode KNN digunakan untuk melakukan klasifikasi dan pengenalan pola tanda tangan. Berdasarkan hasil penelitian pengenalan tanda tangan menggunakan

metode *Principal Component Analysis* dan *K-Nearest Neighbor*, dapat disimpulkan bahwa fitur PCA belum mencirikan suatu citra sehingga dalam pengujian pengenalan tanda tangan tidak memberikan hasil yang memuaskan. Meskipun demikian, hasil pengujian pada skenario pertama, kedua, dan ketiga menunjukkan bahwa persentase *recall* terbaik ada pada $k=5$ yaitu 4,95%, sedangkan untuk *persentase accuracy* terbaik ada pada $k=10$ dan 15 yaitu 87,55%. Namun, nilai rata-rata *precision* yang didapatkan pada ketiga skenario adalah 6,66%. Oleh karena itu, masih diperlukan penelitian lebih lanjut untuk meningkatkan akurasi pengenalan tanda tangan menggunakan metode ini.

2.2 DASAR TEORI

2.2.1 Pengenalan Citra Tanda Tangan

Pengenalan citra tanda tangan adalah proses pengidentifikasian dan klasifikasi tanda tangan secara otomatis menggunakan metode komputasional. Pada dasarnya tanda tangan symbol yang merupakan versi miniature dari pemiliknya atau tanda tangan bukanlah sembarang tanda atau symbol. Sehingga tujuan dari pengenalan pola tanda tangan adalah untuk menghasilkan model yang dapat membedakan tanda tangan yang berbeda dan mengklasifikasikan tanda tangan yang tidak dikenal [6].

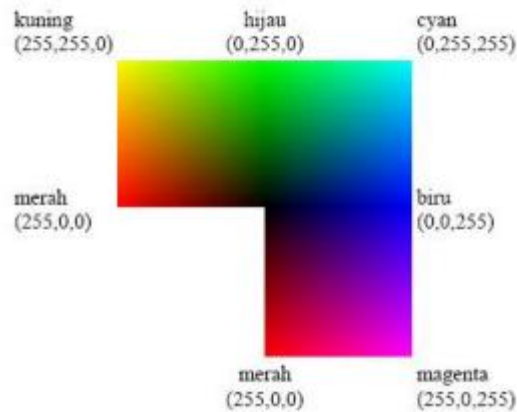


Gambar 2. 1 Tampilan Contoh Tanda Tangan

2.2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah suatu bidang dalam ilmu komputer yang berkaitan dengan pemrosesan dan analisis citra yang diambil dari sumber digital, seperti kamera atau sensor digital lainnya atau bisa dikatakan proses pengolahan gambar digital yang menggunakan komputer, dengan kata lain pengolahan citra digital menggunakan algoritma komputer untuk mengekstrak berbagai informasi berharga atau menghasilkan gambar yang lebih baik [7]. Tujuannya adalah untuk mengubah, memperbaiki, atau menganalisis citra secara komputerisasi untuk mendapatkan informasi yang lebih berguna atau untuk menghasilkan tampilan visual yang lebih baik. Pengolahan citra digital memiliki aplikasi yang luas, termasuk dalam bidang kedokteran, pemrosesan gambar dan video, pengenalan karakter dan tulisan tangan, pengawasan keamanan, pengenalan pola geospasial, augmented reality, dan banyak lagi. Pengembangan teknologi pengolahan citra digital terus berlanjut dan memainkan peran penting dalam berbagai aspek kehidupan sehari-hari dan industry.

Pada pengolahan citra digital pada penelitian ini menggunakan citra RGB dan RGB dimana citra RGB itu sendiri adalah ciri warna asli (juga dikenal sebagai citra RGB) terdiri dari matriks data berukuran $m \times n \times 3$ yang penekanan setiap piksel dengan warna merah, hijau, dan biru. Setiap warna memiliki rentang nilai yang berbeda, dengan nilai paling kecil 0 dan paling besar 255 untuk monitor komputer. Pilihan skala 256 ini didasarkan pada cara komputer mengungkap bilangan biner 8 digit. Jadi, ada lebih dari 16 juta warna yang dapat diperoleh. Intensitas merah, hijau, dan biru digunakan untuk menentukan warna masing-masing piksel [8]. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 1677726$ jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(30, 75, 255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0) [9].



Gambar 2. 2 Warna RGB pada Citra Digital

Sistem warna aditif seperti RGB telah menjadi standar dalam materi digital. Ini disebut RGB karena ketika beberapa warna ditambahkan secara bersamaan, warnanya akan terlihat lebih terang hingga mendekati warna putih [10]. Sangat penting untuk diingat bahwa warna aditif terbuat dari cahaya; semakin banyak cahaya yang ditambahkan, semakin terang dan cerah warna tersebut [11]. Mode warna additive RGB didasarkan pada cara kerja penglihatan manusia, dimana dalam mata memiliki tiga jenis sel kerucut fotoreseptor: gelombang cahaya merah, hijau dan biru. Untuk kemudian otak mengambil cahaya dari ketiga reseptor tersebut dan menerjemahkan menjadi jutaan warna yang dilihat. Cara kerja RGB to Grayscale adalah dengan menghitung intensitas keabuan dari masing-masing piksel berdasarkan nilai komponen warna RGB-nya. Nilai keabuan ini bisa diperoleh dengan berbagai metode, salah satunya adalah dengan menjumlahkan komponen warna RGB dengan bobot tertentu. Citra *grayscale* sering disebut juga dengan citra beraras keabuan. Citra ini memiliki nilai tunggal dengan nilai intensitas berkisar antara 0 sampai 255. Nilai intensitas yang mendekati 255 memiliki derajat keabuan yang semakin terang begitu juga sebaliknya nilai yang mendekati 0 akan semakin gelap. Salah satu metode yang umum digunakan adalah rata-rata dari nilai RGB, yaitu:

$$y = 0.299R + 0.587G + 0.114B \quad (2.1)$$

Kerangan:

R adalah nilai komponen merah (Red) dalam rentang 0 hingga 255.

G adalah nilai komponen hijau (Green) dalam rentang 0 hingga 255.

B adalah nilai komponen biru (Blue) dalam rentang 0 hingga 255.

Y adalah nilai kecerahan suatu piksel pada citra grayscale, dengan persentase 29,9% warna merah, 58,7% dari warna hijau, dan 11,4 dari warna biru [9].

2.2.3 Principal Component Analysis (PCA):

Principal Component Analysis (PCA) adalah sebuah metode dalam analisis statistik yang digunakan untuk mengurangi dimensi dari suatu dataset atau bisa dikatakan suatu teknik untuk menemukan pola dalam data dan menunjukkan adanya perbedaan atau kesamaan dalam kumpulan data. Biasanya digunakan untuk mereduksi dimensi data menjadi bentuk yang berada di bidang nilai yang berbeda. Metode ini bekerja dengan menghitung covariance matrix data sebelum menemukan eigenvectors dan eigenvalues.. PCA mencari kombinasi linear baru dari variabel asli yang disebut komponen utama (*principal components*) yang menjelaskan sebagian besar variasi dalam data. Algoritma dari metode PCA sebagai berikut :

1. Membentuk matriks parameter klasifikasi dari data yang akan dilatih, persamaan (2).

$$X = \begin{matrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \\ \dots & \dots & \dots \\ X_{m1} & X_{m2} & X_{m3} \end{matrix} \quad (2.2)$$

2. Menghitung rata-rata nilai dari seluruh parameter terhadap semua data yang digunakan, m jumlah data, persamaan (3).

$$x = \frac{X_{12}+X_{21}+X_{31}+\dots+X_{m1}}{m} = \sum_j^m = 1^x j i = [T_1, T_2, \dots, \dots, \dots, T_n] \quad (2.3)$$

3. Penggandaan rata-rata atau means. Proses ini dilakukan untuk membuat dimensi nilai rata-rata sama dengan dimensi nilai pada citra sampel. μ adalah matriks kolom yang memiliki dimensi $m \times 1$, dengan Rumus yang digunakan seperti pada Persamaan (4).

$$\mu = axT \quad (2.4)$$

4. Melakukan perhitungan terhadap nilai yang memiliki rata-rata nol. Hal itu dilakukan agar hasil perhitungan PCA menjadi lebih akurat. Rumus yang digunakan seperti pada Persamaan (5).

$$\phi = x - \mu \quad (2.5)$$

5. Memperoleh nilai matriks kovarian berdasarkan hasil perhitungan rata-rata nol. Dapat dihitung menggunakan Persamaan (6).

$$c = \frac{1}{m-1} \phi_{ji} * \phi^T ji \quad (2.6)$$

6. Menentukan matriks eigenvalue dan eigenvector. Dapat dihitung dengan Persamaan (7). Dimana C adalah matriks kovarian, sedangkan Z adalah matriks eigenvalue dengan λ adalah matriks skalarnya [11].

$$C - Z = |C - I| = |C - I| = 0 \quad (2.7)$$

Penggunaan PCA dalam Pengenalan Pola Tanda Tangan:

Metode PCA dapat diterapkan dalam pengenalan pola tanda tangan dengan langkah-langkah sebagai berikut:

- a. Pemrosesan Data:

Data tanda tangan dalam bentuk citra harus diproses terlebih dahulu sebelum diterapkan PCA. Proses ini melibatkan segmentasi citra tanda tangan, ekstraksi fitur, dan normalisasi untuk memastikan bahwa data tanda tangan memiliki format yang seragam.

- b. Ekstraksi Fitur:

Setelah pemrosesan data, fitur-fitur yang relevan dari citra tanda tangan diekstraksi. Fitur-fitur ini dapat mencakup informasi seperti tekanan pena, kecepatan, sudut, panjang garis, dan sebagainya. Ekstraksi fitur bertujuan untuk mengidentifikasi pola unik dalam tanda tangan yang dapat dibedakan secara statistik.

- c. Reduksi Dimensi menggunakan PCA:

Setelah ekstraksi fitur, metode PCA dapat diterapkan untuk mengurangi dimensi dari dataset fitur. PCA akan menghasilkan komponen utama yang merupakan kombinasi linear dari fitur-fitur asli. Komponen utama ini disusun berdasarkan tingkat variabilitasnya, dengan komponen pertama menjelaskan variasi terbesar dalam data.

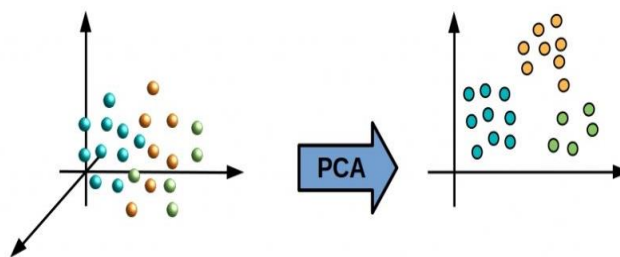
- d. Klasifikasi:

Setelah reduksi dimensi menggunakan PCA, teknik klasifikasi dapat diterapkan untuk mengklasifikasikan tanda tangan yang tidak dikenal berdasarkan model yang telah dibangun sebelumnya. Teknik klasifikasi yang

umum digunakan termasuk metode jarak terdekat, jaringan saraf tiruan (neural networks), atau metode klasifikasi lainnya.

e. Evaluasi dan Validasi:

Akurasi sistem pengenalan pola tanda tangan yang menggunakan PCA dapat dievaluasi dan divalidasi dengan menggunakan metrik evaluasi seperti tingkat keberhasilan pengenalan, tingkat kesalahan, kecepatan pengenalan, dan metrik kinerja lainnya



Gambar 2. 3 Tampilan Metode PCA

Gambar 2.3 merupakan tampilan metode PCA, dimana secara sederhananya memiliki 5 langkah cara kerjanya. Dimana pada tahap pertama, seluruh variabel disamakan dengan standarnya. Oleh karena itu, setiap variabel dapat memberikan dampak yang sama pada hasil analisis. Selanjutnya menghitung matriks kovarians. Proses ini memungkinkan peneliti untuk mengidentifikasi hubungan antara variabel dalam kumpulan input. Setelah itu, peneliti dapat menemukan nilai dan vektor eigen matriks kovarians. Tujuannya adalah agar peneliti dapat menemukan elemen utama. Vektor fitur adalah bagian terakhir dari analisis. Peneliti dapat menemukan komponen yang kurang penting dan dapat dibuang, dengan melakukan komputasi nilai dan vektor eigen. Vektor fitur adalah matriks vektor yang tersisa. Setelah itu, analisis ditutup dengan perubahan pada sumbu komponen utama [12].

Pada metode PCA ini memiliki keuntungan dibandingkan dengan metode lain diantaranya PCA secara efektif mengurangi dimensi kumpulan data kompleks sambil mempertahankan informasi yang paling penting. Ini memungkinkan representasi data yang lebih sederhana tanpa kehilangan fitur penting, dan PCA membantu visualisasi data dimensi tinggi dalam ruang dimensi yang diperkecil

dengan memplot komponen utamanya. Ini memungkinkan untuk melihat pola, kelompok, dan hubungan antara titik data, yang memudahkan interpretasi dan pemahaman data serta PCA dapat menemukan elemen kumpulan data yang paling informasi, yang memungkinkan peneliti untuk fokus pada elemen utama yang menjelaskan sebagian besar varian. Ini memungkinkan peneliti untuk memilih fitur dengan tepat dan mengurangi kebisingan atau redundansi saat melakukan analisis lebih lanjut [13].

2.2.4 K-Nearest Neighbor (K-NN)

K-Nearest Neighbor (K-NN) adalah algoritma klasifikasi yang sederhana namun efektif. Algoritma ini mengklasifikasikan objek baru berdasarkan mayoritas label dari K tetangga terdekatnya dalam ruang fitur. Dalam konteks pengenalan tanda tangan, K-NN dapat digunakan untuk membandingkan tanda tangan yang tidak dikenal dengan tanda tangan yang telah diklasifikasikan sebelumnya .

Metode Pengenalan Tanda Tangan dengan PCA dan K-NN [14]:

a. Pemrosesan Data:

Data tanda tangan dalam bentuk citra harus diproses terlebih dahulu sebelum diterapkan PCA. Proses ini melibatkan segmentasi citra tanda tangan, ekstraksi fitur, dan normalisasi untuk memastikan bahwa data tanda tangan memiliki format yang seragam.

b. Ekstraksi Fitur:

Setelah pemrosesan data, fitur-fitur yang relevan dari citra tanda tangan diekstraksi. Fitur-fitur ini dapat mencakup informasi seperti tekanan pena, kecepatan, sudut, panjang garis, dan sebagainya.

c. Reduksi Dimensi menggunakan PCA:

Metode PCA diterapkan untuk mengurangi dimensi dari dataset fitur tanda tangan. PCA akan menghasilkan komponen utama yang merupakan kombinasi linear dari fitur-fitur asli, dan komponen utama tersebut disusun berdasarkan tingkat variabilitasnya.

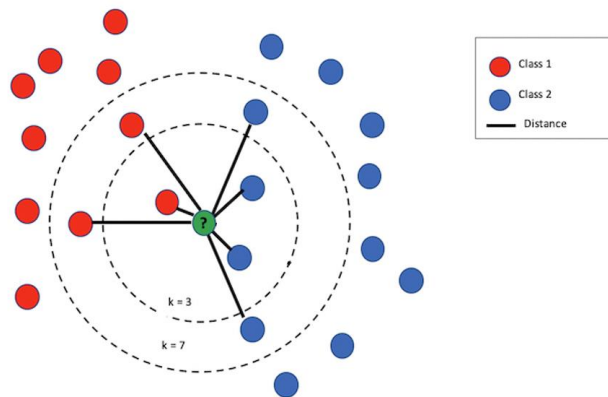
d. Klasifikasi Menggunakan K-NN:

Setelah reduksi dimensi menggunakan PCA, tanda tangan yang tidak dikenal dapat diklasifikasikan menggunakan algoritma K-NN. Jarak antara tanda tangan yang tidak dikenal dengan tanda tangan yang telah diklasifikasikan sebelumnya

dihitung, dan label dari K tetangga terdekat digunakan untuk menentukan label klasifikasi.

e. Evaluasi dan Validasi:

Sistem pengenalan tanda tangan dengan menggunakan metode PCA dan K-NN dapat dievaluasi dan divalidasi dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Metode validasi silang (cross-validation) juga dapat digunakan untuk menguji kinerja sistem pada data.



Gambar 2. 4 Tampilan Metode KNN [14].

Pada gambar 2.4 merupakan tampilan metode KNN, dimana algoritma knn digunakan untuk mengklasifikasikan data. Oleh karena itu, terlebih dahulu harus menentukan nilai k jika ingin mengetahui kelas sebuah objek. Sebagai contoh, pada gambar di atas, knn akan digunakan untuk mengetahui apakah objek berbentuk lingkaran hijau termasuk ke dalam kelas lingkaran merah atau lingkaran biru. Nilai k dapat ditetapkan dengan cara apa pun sehingga dalam hal ini, nilai k ditetapkan menjadi 2 yang berarti bahwa sebuah lingkaran dengan objek bintang hijau sebagai pusat hanya mengandung dua data saja. Kemudian untuk metode KNN sendiri memiliki kelebihan diantaranya algoritma KNN dapat menangani data bising, mudah digunakan, dan dapat menangani banyak data. Sedangkan untuk kekurangan metode KNN sebagai berikut pembelajaran berdasarkan jarak tidak jelas tentang jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, nilai parameter K (jumlah tetangga terdekat) harus ditentukan oleh KNN, dan perhitungan jarak dari tiap sample uji pada keseluruhan sample latih membutuhkan banyak daya komputasi. Rumus yang digunakan untuk perhitungan KNN seperti pada persamaan (7) $d(i,j)$ adalah jarak, x_1 adalah nilai pada fitur 1, dan x_j nilai pada fitur 2 [15].

$$d(i, f) = \sqrt{|X_{i1} - X_{j1}|^2 + |X_{i2} - X_{j2}|^2 + \dots + |X_{ip} - X_{jp}|^2} \quad (2.8)$$

2.2.5 Confusion Matrix

Confusion matrix adalah salah satu metode evaluasi yang umum digunakan dalam penilaian kinerja algoritma klasifikasi. Ini menggunakan matriks yang dibuat dari perbandingan prediksi dengan kelas asli dari data latih pada data yang diuji. Dalam matriks ini, hasil pengujian dikelompokkan menjadi tanda positif dan negatif, dengan baris dan kolom positif menunjukkan hasil yang sebenarnya, sementara hasil negatif mencerminkan kesalahan prediksi. *Confusion matrix* memungkinkan evaluasi terhadap presisi, nilai presisi, dan tingkat *recall* dari hasil tes. Akurasi adalah ukuran seberapa baik nilai prediksi menyamai nilai sebenarnya, dan digunakan untuk menilai keakuratan tes [16].

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<p>TP (True Positive)</p>	<p>FP (False Positive) <small>Type I Error</small></p>
	0 (Negative)	<p>FN (False Negative) <small>Type II Error</small></p>	<p>TN (True Negative)</p>

Gambar 2. 5 *Confusion matrix* [16].

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (2.9)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (2.10)$$

$$\text{Preccision} = \frac{TP}{TP+FP} \times 100\% \quad (2.11)$$

Keterangan :

TP = *True Positive*

TN = *True Negative*

FP = *False Positive*

FN = *Fals Negative*

2.2.6 Artificial Intelligence (AI)

Artificial Intelligence (AI) merupakan salah satu yang sering digunakan untuk pengenalan obyek. *Artificial Intelligence* (AI) atau kecerdasan buatan adalah cabang ilmu komputer yang berfokus pada pengembangan sistem yang dapat

melakukan tugas-tugas yang membutuhkan kecerdasan manusia atau bisa dikatakan kemampuan mesin untuk meniru atau mengimitasi kecerdasan manusia. Hal ini termasuk berbagai metode dan pendekatan yang memungkinkan komputer untuk memahami, belajar, dan menggunakan data untuk membuat keputusan. AI bertujuan untuk menciptakan mesin yang dapat belajar, memahami, merencanakan, dan melakukan tindakan yang cerdas seperti yang dilakukan oleh manusia [16].

Kemudian sistem AI mempunyai kelebihan dan kekurangan, dimana untuk kelebihanya diantaranya bisa menganalisis data dengan cepat dan akurat meskipun dalam jumlah besar yang memungkinkan pengambilan keputusan dan prediksi yang lebih baik, dan dapat menyelesaikan tugas dengan cepat dan dengan akurat, yang menghemat waktu dan biaya dalam banyak hal. Sedangkan kekurangannya yaitu untuk pelatihan, kecerdasan buatan sangat bergantung pada data yang digunakan. Jika data tidak representatif atau terkontaminasi oleh bias, sistem dapat menghasilkan hasil yang tidak akurat atau tidak adil [17].

2.2.7 Deep Learning

Deep learning adalah sub-bidang pembelajaran mesin yang berfokus pada penggunaan jaringan saraf tiruan yang dalam (*deep neural networks*) untuk memodelkan dan mempelajari representasi hierarkis dari data. Deep learning telah memberikan kemajuan signifikan dalam bidang pengenalan suara, pengenalan wajah, pemrosesan bahasa alami, dan bidang lainnya. Deep Learning adalah bagian dari pembelajaran mesin yang memilih fitur. Ini menggunakan Jaringan Saraf Tiruan dan lebih banyak daya komputasi. Pembelajaran Mesin membutuhkan lebih sedikit waktu pelatihan dan lebih banyak waktu pengujian sedangkan Deep Learning membutuhkan lebih banyak waktu pelatihan dan lebih sedikit waktu pengujian yang membuatnya lebih cocok untuk penyebaran. Pembelajaran mendalam mengeksplorasi semua fitur, parameter dan hyperparameters dan kemudian terapkan filter untuk memutuskan fitur mana yang harus dilanjutkan [18].

2.2.8 Google Colab

Google Colaboratory merupakan layanan layanan sumber terbuka yang disediakan oleh Google untuk setiap orang yang memiliki akun *Gmail* atau bisa dikatakan program yang dapat dilaksanakan yang dapat disimpan, ditulis, dan dibagikan melalui Google Drive [19]. *Google Colab* menyediakan GPU untuk

penelitian kepada orang-orang yang tidak memiliki sumber daya yang cukup atau tidak mampu membelinya. Layanan *Google Colab* menyediakan 12,72 GB RAM dan 358,27 GB ruang *harddisk* dalam satu runtime. Setiap *runtime* berlangsung selama 12 jam setelah itu runtime diatur ulang dan pengguna harus membuat koneksi lagi. Ini untuk memastikan bahwa orang tidak menggunakan GPU layanan untuk penambangan mata uang kripto dan ilegal lainnya ilegal lainnya. Setelah pengguna membuka file *Google Colab*, mereka harus memilih jenis runtime. Ada 3 opsi yang tersedia untuk hal yang sama,

1. Tidak ada (yang akan menggunakan CPU komputer yang digunakan pengguna)
2. GPU
3. TPU (terutama pemrosesan tensor)

Google Colab pada dasarnya adalah sebuah buku catatan *Jupiter* online dan memiliki semua fungsi yang ada di dalamnya. Pengguna dapat menghubungkan *Google Colab* ke *Google Drive* mereka. Setelah *Google Drive* dipasang ke *Google Colab*, kemudian user dapat menggunakannya sebagai *Jupyter Notebook* dan menulis kode kerja ke dalam sel atau memiliki file *python* di *Google Drive* dan menjalankan file-file tersebut dari *Google Colab* seperti halnya menjalankannya dari terminal *Linux* tetapi dengan tanda '!' sebelum pernyataan



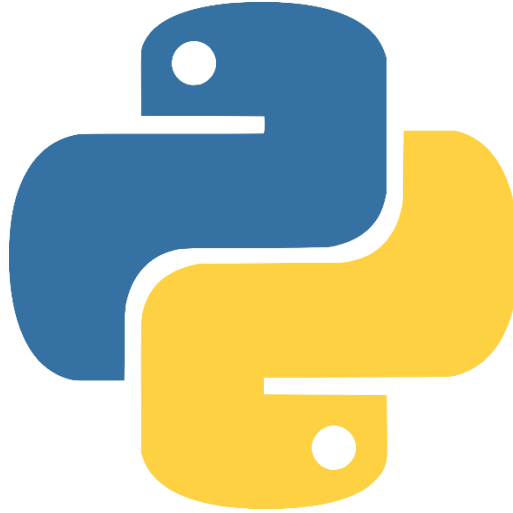
Gambar 2. 6 Tampilan *Google Colab* [19].

2.2.9 PYTHON

Python merupakan *Python* merupakan bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multiguna secara langsung (*interpretatif*) dengan berorientasi pada objek serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan kode atau *syntax*. Sebagian lain mengartikan *Python* sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif atau bisa dikatakan pemrograman multifungsi yang dapat diterapkan pada hampir semua arsitektur sistem, mulai dari pengembangan web hingga pengajaran mesin, dan dapat digunakan untuk berbagai aplikasi [20]. *Python* sendiri menampilkan fitur-fitur menarik salah satunya *python* memiliki tata bahasa dan script yang sangat mudah untuk dipelajari. *Python* juga memiliki sistem pengelolaan data dan memori otomatis. Selain itu modul pada *Python* selalu di-*update*. Ditambah lagi, *Python* juga memiliki banyak fasilitas pendukung. *Python* banyak diaplikasikan pada berbagai sistem operasi seperti *Linux*, *Microsoft Windows*, *Mac OS*, *Android*, *Symbian OS*, *Amiga*, *Palm* dan lain-lain. *Python* merupakan bahasa pemrograman yang bersifat dinamis dan tingkat tinggi. Bahasa ini berfungsi sebagai bahasa pemrograman interpreter, yang artinya ia mengubah source code menjadi machine code langsung saat program berjalan. Selain itu, *Python* juga mendukung pendekatan pemrograman Berorientasi Objek dalam pengembangan aplikasi, dan ia mempunyai tingkat kemudahan dalam pembelajarannya. *Python* juga menawarkan beragam struktur data tingkat tinggi.

Python juga dikenal sebagai bahasa skrip yang mudah diajarkan, tetapi memiliki kekuatan dan fleksibilitas yang luar biasa, menjadikannya menarik dalam Pengembangan Aplikasi. Sifat dinamis dari sintaks dan pengetikan *Python* membuatnya cocok untuk skrip dan pengembangan aplikasi yang berjalan dengan cepat. Bahasa ini mampu mendukung berbagai pola pemrograman, termasuk pendekatan berorientasi objek, imperatif, fungsional, dan prosedural. Keunikan *Python* terletak pada kemampuannya untuk bekerja dalam berbagai domain, tidak terbatas pada pemrograman web saja. Inilah sebabnya mengapa *Python* dikenal sebagai Bahasa multiguna, mampu digunakan dalam lingkup web, enterprise, CAD 3D, dan lainnya. Salah satu aspek menarik dari *Python* adalah deklarasi variabelnya

yang tidak memerlukan tipe data karena pengenalan tipe dilakukan secara dinamis. Misalnya, kita dapat langsung menetapkan nilai bilangan bulat ke variabel dengan menulis `a=10`.



Gambar 2. 7 Tampilan *Logo Python* [20].