

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Dari hasil penelitian yang telah dilakukan sebelumnya pada tahun 2022 oleh Agus Mulyadi mengenai pengenalan wajah menggunakan metode PCA dan KNN, didapatkan tingkat akurasi tertinggi sebesar 93.5% dengan waktu komputasi 1.4 detik pada dataset AT&T [8]. Penelitian pengenalan wajah oleh Eko Setiawan dkk. menggunakan metode KNN pada dataset ORL mendapatkan akurasi sebesar 91.5% Yaleface mendapatkan akurasi sebesar 78.8%, dan MUCT mendapatkan akurasi sebesar 70%, dengan nilai $k=1$ dan rata-rata waktu komputasi untuk ketiga dataset 0.152 detik pada *high-power processor* dan 2.66 detik pada *low-power processor* [9]. Hasil penelitian lain oleh Tumpa Dey dkk. menggunakan ORL database dan *Facial Expression Database - Japanese Female* (JAFFE), teknik pengelompokan K-Means dapat mencapai tingkat pengenalan yang baik sebesar 90% dan 85% untuk klasifikasi dan pengenalan wajah manusia dengan waktu komputasi yang lebih sedikit dibandingkan metode lainnya [10]. Hasil penelitian dengan menggunakan metode serupa untuk identifikasi wajah oleh Ayu Wirdiani dkk. pada tahun 2019 menggunakan PCA dan KNN untuk klasifikasi mendapatkan hasil terbaik dengan tingkat akurasi 81% pada $k=1$ pada dataset mandiri[11]. Pada dataset Georgia Tech, Agus Mulyadi mendapatkan tingkat akurasi 78% dengan waktu komputasi 5 detik. Sedangkan pada dataset Yale B, penelitian yang sama menghasilkan tingkat akurasi sebesar 84% dengan waktu komputasi 7 detik.

2.2 Pengolahan Citra (*Image Processing*)

Sebuah citra adalah gambar dalam bidang dua dimensi yang dihasilkan oleh suatu perangkat yang disebut kamera. Secara matematis, citra dapat dianggap sebagai fungsi dua dimensi yang menggambarkan intensitas cahaya dalam bidang tersebut, disimbolkan sebagai $f(x,y)$. Di sini, $f(x,y)$ mewakili intensitas cahaya pada koordinat titik (x,y) . Citra digital pada dasarnya merupakan representasi citra yang berasal dari pencuplikan (*sampling*) secara ruang dan waktu, dan juga direpresentasikan sebagai matriks berukuran $M \times N$. setiap elemen matriks menyatakan sebuah piksel (*picture element*). $M \times N$ adalah resolusi citra [12].

Pengolahan Citra adalah prosedur mengonversi sebuah gambar ke dalam ben-

tuk digital dan melakukan beberapa operasi pada gambar tersebut, dengan tujuan untuk mendapatkan gambar yang lebih baik dan mengekstrak beberapa informasi yang berguna dari gambar tersebut. Pengolahan Citra umumnya merujuk pada pemrosesan gambar digital [13]. Berikut representasi dari citra digital:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Pengolahan Citra Digital digunakan untuk mengubah gambar menjadi bentuk digital sekaligus melakukan operasi di atasnya, agar didapatkan gambar yang disempurnakan atau untuk tujuan mengekstrak informasi yang berguna dari gambar tersebut [14]. Gambar 2.1 merupakan contoh dari suatu citra dengan resolusi 256×256 pixels.

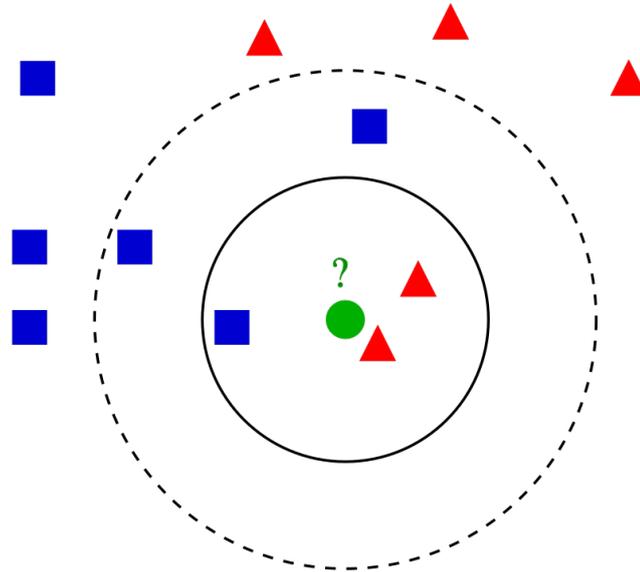


Gambar 2.1. Citra dengan resolusi 256×256

2.3 *K-nearest neighbor* (KNN)

K-nearest neighbor (KNN) adalah salah satu algoritma *machine learning* yang paling sederhana. KNN digunakan sebagai teknik klasifikasi di mana sebuah database dicari untuk menemukan elemen-elemen yang paling mirip dengan elemen *query* yang diberikan, dengan kesamaan yang didefinisikan oleh fungsi jarak. Setiap elemen dalam database memiliki label (*class*) yang terkait, dan tujuan utama

dari algoritma ini adalah untuk menentukan *kelas* dari kasus baru berdasarkan *class* dari elemen 'k' database yang paling mirip [15]. Gambar 2.2 merupakan ilustrasi proses klasifikasi metode KNN.



Gambar 2.2. Klasifikasi KNN

Terdapat beberapa perhitungan jarak yang dapat digunakan untuk menentukan nilai 'k', sebagai berikut :

1. *Euclidean Distance*

Dengan menggunakan perhitungan *Euclidean Distance* kita dapat menentukan jarak antara data citra latih dan citra uji pada suatu sistem dengan menggunakan *Euclidean Space*.

$$d_{(i,j)} = \sqrt{\sum_{k=1}^n (i_k - j_k)^2} \quad (2.2)$$

2. *Manhattan Distance*

Manhattan Distance dapat digunakan untuk mencari jarak setiap data citra latih dan citra uji dengan menggunakan jumlah dari nilai absolut pada perbedaan koordinat *cartesian* di dua titik.

$$d_{(i,j)} = \sum_{k=1}^n |i_k - j_k| \quad (2.3)$$

3. Minkowski Distance

Minkowski Distance, perhitungan untuk mencari jarak tetangga terdekat dengan matriks ruang vektor dimana dapat didefinisikan sebagai penyamarataan *Euclidean Distance* dan *Manhattan Distance*.

$$d_{(i,j)} = \left(\sum_{k=1}^n (i_k - j_k)^r \right)^{1/r} \quad (2.4)$$

4. Cosine Distance

Cosine Distance adalah perhitungan untuk mencari jarak dari nilai 0 hingga 2. Jika data latih dan data uji dapat dikatakan mirip diantara setiap data dan menghasilkan sudut 0° maka nilai yang dihasilkan adalah satu. Jika setiap data yang dipilih memiliki sudut 90° maka nilai yang dihasilkan adalah nol.

$$D_c(i, j) = 1 - S_c(i, j) \quad (2.5)$$

$$S_c(i, j) = \frac{\sum_{k=1}^n i_k j_k}{\sqrt{\sum_{k=1}^n (i_k)^2} \sqrt{\sum_{k=1}^n (j_k)^2}} \quad (2.6)$$

5. Correlation

Koefisien korelasi adalah perhitungan untuk mencari jumlah dari dua variabel yang berkorelasi. Jika hasil korelasi yang didapatkan (1.0), maka kedua variabel memiliki korelasi yang sempurna. Sedangkan untuk hasil korelasi (0), maka kedua variabel tidak memiliki kaitan satu sama lain [16].

$$r(i, j) = \frac{N \sum_{k=1}^n i_k j_k - \sum_{k=1}^n i_k \sum_{k=1}^n j_k}{\sqrt{N \sum_{k=1}^n (i_k)^2 - (\sum_{k=1}^n i_k)^2} \sqrt{N \sum_{k=1}^n (j_k)^2 - (\sum_{k=1}^n j_k)^2}} \quad (2.7)$$

2.4 K-Means Clustering

K-means Clustering adalah algoritma yang digunakan untuk mengelompokkan data ke dalam sejumlah klaster yang ditentukan sebelumnya berdasarkan kesamaan fitur-fitur dalam data tersebut. Namun, algoritma K-means memiliki beberapa keterbatasan, seperti sensitivitas terhadap inisialisasi awal dan keharusan untuk menentukan jumlah klaster sebelumnya [17].

Algoritma ini bekerja dengan cara menginisialisasi titik-titik pusat klaster (*centroid*) secara acak, kemudian mengalokasikan setiap titik data ke klaster terdekat

berdasarkan jarak Euclidean, dan mengupdate titik pusat kluster berdasarkan rata-rata titik-titik data yang telah dialokasikan ke dalam kluster tersebut. Proses ini diulangi hingga tidak ada perubahan signifikan dalam alokasi kluster atau titik pusat kluster, atau hingga mencapai jumlah iterasi yang ditentukan [18]. Berikut merupakan perhitungan untuk algoritma K-Means :

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2 \quad (2.8)$$

J = fungsi objektif

k = jumlah kluster

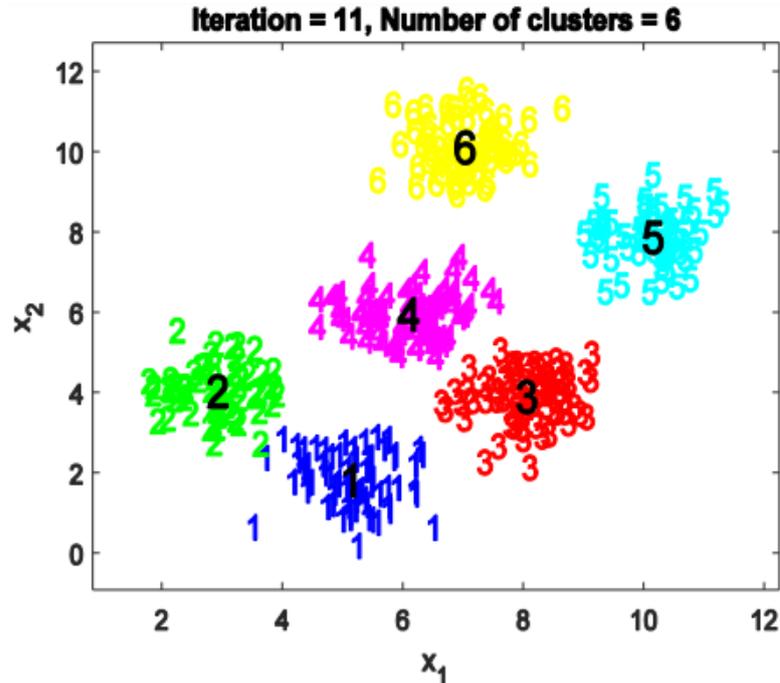
n = jumlah total data

x_i = indeks i

c_j = centroid untuk kluster

$\left\| x_i^{(j)} - c_j \right\|^2$ = fungsi jarak

Gambar 2.3 merupakan hasil klustering setelah dilakukan perulangan sebanyak sebelas kali dan menghasilkan jumlah kluster sebanyak enam, yang didapatkan dari delapan dataset [17].



Gambar 2.3. Klustering K-Means

2.5 Dataset Wajah Yang Digunakan

Dataset wajah memiliki peran penting dalam pengembangan dan evaluasi sistem pengenalan wajah. Dalam penelitian ini, percobaan dan pengujian berfokus pada penggunaan tiga dataset wajah yang terkenal, yaitu dataset wajah AT&T, Yale Face, dan Georgia. Ketiga dataset ini telah lama menjadi pilihan utama dalam penelitian pengenalan wajah karena kualitasnya yang baik, keberagaman subjek, serta ketersediaan metadatanya yang kaya. Dataset AT&T terdiri dari sejumlah besar citra wajah resolusi rendah dari sejumlah subjek, sementara Yale Face dataset menyediakan beragam pose, pencahayaan, dan ekspresi wajah. Sementara itu, dataset Georgia menampilkan wajah dengan berbagai kondisi lingkungan dan perbedaan kualitas gambar.

2.5.1 Dataset AT&T

Data wajah AT&T diambil pada bulan April 1992 dan 1994. Database ini digunakan dalam konteks proyek pengenalan wajah yang dilakukan dengan kolaborasi dengan kelompok *Speech, Vision, and Robotics* dari Departemen Teknik Cambridge University. Dataset ini berisi sepuluh gambar berbeda dari masing-masing 40 subjek yang berbeda. Untuk beberapa subjek, gambar-gambar tersebut diambil pada waktu yang berbeda, dengan variasi pencahayaan, ekspresi wajah. Semua gambar diambil dengan latar belakang yang gelap dengan subjek terlihat dari depan dan dalam posisi tegak [19]. Gambar 2.4 adalah contoh citra wajah dari dataset AT&T.

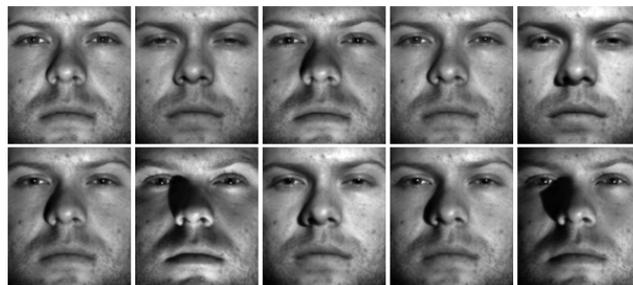


Gambar 2.4. Dataset AT&T

Ukuran setiap gambar adalah 92×112 piksel, dengan 256 tingkat abu-abu per piksel. Gambar-gambar ini disusun dalam 40 direktori (satu untuk setiap subjek).

2.5.2 Dataset *Yale Face*

Dataset Yale Face terdiri dari 2.414 gambar dari 38 subjek dengan masing-masing subjek memiliki 64 kondisi pencahayaan yang bervariasi [20]. Semua data gambar yang akan digunakan dalam penelitian telah disejajarkan secara manual, dipotong, lalu diubah ukurannya menjadi gambar 168×192 [21]. Pada penelitian ini banyak citra yang digunakan adalah 840 citra, gambar dari 28 subjek dengan masing-masing subjek memiliki 30 citra dengan kondisi pencahayaan yang baik. Citra yang digunakan merupakan citra yang dipilih dengan kondisi pencahayaan yang tidak intensif. Gambar 2.5 merupakan contoh citra dari dataset Yale Face.



Gambar 2.5. Dataset *Yale Face*

2.5.3 Dataset *Georgia Tech Face*

Database wajah Georgia Tech berisi gambar-gambar dari 50 orang yang diambil dalam dua sampai tiga sesi pada tahun 1999 di Center for Signal and Image Processing, Institut Teknologi Georgia. Semua orang dalam database ini direpresentasikan oleh 15 gambar JPEG berwarna dengan latar belakang yang berantakan diambil dengan resolusi 640×480 piksel. Ukuran rata-rata wajah dalam gambar-gambar ini adalah 150×150 piksel [22]. Gambar-gambar ini menampilkan wajah-wajah frontal dan miring dengan ekspresi wajah, kondisi pencahayaan, dan skala yang berbeda. Setiap gambar dilabeli secara manual untuk menentukan posisi wajah dalam gambar tersebut. Gambar 2.6 merupakan contoh citra wajah dari dataset *Georgia Tech Face*.



Gambar 2.6. Dataset *Georgia Tech Face*

2.6 Python

Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido van Rossum untuk pertama kalinya di Centrum Wiskunde & Informatica (CWI) di Belanda pada awal tahun 1990-an. Python dapat dijalankan pada berbagai sistem operasi dan memiliki struktur program yang sederhana, sehingga mudah dipelajari. Python mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural, pemrograman berorientasi objek, dan pemrograman fungsional [23].

Python memiliki kemampuan untuk memperluas bahasa dengan kode yang dikumpulkan, yang dapat secara signifikan mempercepat eksekusi. Selain itu, Python dilengkapi dengan perpustakaan standar yang komprehensif dan kemampuan untuk dengan mudah menggabungkan modul tambahan, memungkinkan pengembangan program dengan cepat. Python juga bersifat *open source* dan dapat dijalankan pada berbagai platform, yang menambah fleksibilitas [24].