

## BAB II TINJAUAN PUSTAKA

### 2.1 Tanaman Hortikultura

Tanaman hortikultura merupakan jenis tanaman yang ditanam di area pekarangan atau lahan tertentu. Tanaman hortikultura mempunyai sifat pembusukan yang cepat oleh karena itu, tanaman harus cepat dipanen untuk menghindari pembusukan, mengandung unsur gizi yang dibutuhkan manusia, serta mudah dalam pengolahan dan penanganannya. Selain untuk memenuhi kebutuhan pangan dan gizi masyarakat, tanaman hortikultura yang ditanam di kawasan pekarangan dapat meningkatkan pendapatan masyarakat (Rahayu et al., 2022). Ilustrasi dari tanaman hortikultura dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Tanaman Holtikultura  
Sumber : (Nur'aini, 2019)

Menurut Badan Pusat Statistik Indonesia pada tahun 2021, produksi cabai besar mencapai 1,36 juta ton, meningkat sebesar 7,62% (96,38 ribu ton) dibandingkan tahun 2020. Konsumsi cabai besar oleh sektor rumah tangga pada tahun 2021 mencapai 596,14 ribu ton, naik sebesar 8,49% (46,67 ribu ton) dari tahun sebelumnya. Konsumsi dari sektor rumah tangga menyumbang 72,94% dari total konsumsi cabai besar. Produksi cabai besar tertinggi pada tahun 2021 terjadi pada bulan November, mencapai 124,72 ribu ton dengan luas panen 28,12 ribu hektar. Provinsi dengan produksi cabai besar terbesar adalah Jawa Barat, Sumatera Utara, dan Jawa Tengah. Jawa Barat menyumbang 25,21% dari produksi nasional dengan total produksi 343,07 ribu ton dan luas panen 23,12 ribu hektar. Sumatera

Utara menyumbang 15,45% dengan produksi 210,22 ribu ton dan luas panen 17,22 ribu hektar. Sementara itu, Jawa Tengah berkontribusi 12,44% dengan produksi mencapai 169,28 ribu ton dan luas panen 22,68 ribu hektar.

## 2.2 Cabai Merah Besar

Cabai merah besar (*Capsicum annuum*) adalah salah satu jenis tanaman hortikultura yang dikenal karena rasa pedas dan panasnya, disebabkan oleh kandungan kapsaisin dalam getahnya. Berdasarkan Peraturan Presiden Nomor 48 Tahun 2016 tentang Penegasan kepada Perum Bulog mengenai Persediaan Pangan Nasional, cabai merah termasuk dalam daftar 12 bahan pangan pokok yang memerlukan perhatian khusus terkait ketersediaan dan kestabilannya. Hal ini karena cabai merupakan komoditas hortikultura strategis dengan nilai ekonomi tinggi, yang penting untuk memenuhi kebutuhan domestik, industri, dan ekspor (Nuha et al., 2023). Adapun contoh cabai merah besar dapat dilihat pada Gambar 2.2



**Gambar 2.2** Cabai Merah Besar  
Sumber : (Lubis, 2021)

Cabai merah besar adalah salah satu jenis sayuran yang memiliki nilai ekonomi yang relatif rendah. Keinginan akan rumah industri cabai merah skala besar yang memenuhi; Namun, keinginan terhadap cabai merah dalam skala besar seringkali tidak sesuai dengan harapan sehingga sering terjadi fluktuasi harga dari tahun ke tahun berikutnya (Suparyana, 2023).

## 2.3 Penyakit pada daun cabai

### 2.3.1 Virus Mosaik Cabai

Penyakit virus mosaik pada cabai disebabkan oleh *Chili Mosaic Virus* (CMV). Gejala penyakit ini meliputi perubahan warna daun yang belang antara hijau tua dan hijau muda, serta perubahan bentuk daun yang bisa menjadi cekung, keriting, atau memanjang. Penyakit ini ditularkan oleh kutu daun, dan salah satu cara untuk mencegahnya adalah dengan menggunakan insektisida untuk mengontrol populasi kutu daun. Contoh penyakit virus mosaik pada cabai dapat dilihat pada Gambar 2.3.



**Gambar 2.3** Penyakit Virus Mosaik Cabai  
Sumber : (Marianah, 2020)

### 2.3.2 Layu Fusarium

Penyakit layu fusarium disebabkan oleh patogen *Fusarium oxysporum* atau *Ralstonia solanacearum*. Gejala penyakit ini meliputi daun yang menguning dan layu mulai dari bagian atas tanaman. Secara bertahap, gejala ini menyebar ke seluruh tanaman, mengakibatkan kelayuan permanen meskipun daun tetap menempel. Salah satu strategi pengendalian yang banyak diterapkan adalah pengendalian biologis, yang memanfaatkan potensi mikroorganisme sebagai agen antagonis terhadap penyebab penyakit layu fusarium pada tanaman cabai merah (Susanna et al., 2023). Contoh penyakit layu fusarium dapat dilihat pada Gambar 2.4.



**Gambar 2.4** Penyakit Layu Fusarium  
Sumber : (Susanna et al., 2023)

### 2.3.3 Virus Kuning

Virus kuning adalah penyakit yang disebabkan oleh virus gemini. Gejala awalnya meliputi pemutihan pada tulang daun, diikuti dengan perubahan warna daun menjadi kuning, penebalan tulang daun, dan pengkerutan daun ke arah atas. Penyakit ini ditularkan oleh kutu kebul yang sebelumnya berada pada tanaman yang terinfeksi, sehingga tingginya populasi kutu kebul dapat mempengaruhi penyebaran penyakit ini. Selain itu, penyakit ini juga mungkin disebabkan oleh bibit yang sudah terinfeksi sebelumnya (Dzaky, 2021). Contoh penyakit virus kuning dapat dilihat pada Gambar 2.5.



**Gambar 2.5** Penyakit Virus Kuning  
Sumber : (Multazam et al., 2023)

UNIVERSITAS ISLAM NEGERI

### 2.3.4 Bercak Daun

Penyakit bercak daun pada tanaman cabai ditandai dengan gejala awal berupa bercak bulat kecil yang diikuti oleh klorosis, yang kemudian berkembang menjadi nekrosis dan terbentuknya lubang pada daun. Penyakit ini pertama kali menyerang daun tua dan kemudian menyebar ke daun muda yang terletak di pucuk batang atau tangkai bagian atas. Contoh penyakit bercak daun dapat dilihat pada Gambar 2.6.



**Gambar 2.6** Penyakit Bercak Daun  
Sumber : (Lestari & Aini, 2021)

SUMATERA UTARA MEDAN

### 2.3.5 Gejala Penyakit Pada Daun Cabai

Gejala awal dari serangan penyakit pada tanaman cabai merah sering kali tidak menunjukkan perbedaan yang jelas di antara tanaman. Tanaman cabai merah yang ditanam dalam polibag dan dirawat di rumah jaring cenderung mengalami pertumbuhan yang kurang optimal akibat serangan hama seperti aphids dan kutu putih, yang ditemukan hampir di setiap helai daun. Kehadiran aphids pada daun cabai mengindikasikan adanya infeksi oleh lebih dari satu patogen virus. Deskripsi gejala penyakit pada daun cabai dapat dilihat pada Tabel 2.1 di bawah ini.

**Tabel 2.1** Gejala Penyakit Pada Daun Cabai

NO	Nama Penyakit	Gejala	Keterangan
1	Virus Mosaik Cabai	Daun menjadi cekung, keriting, atau memanjang (Febria et al., 2023).	Daun menunjukkan perubahan warna yang belang antara hijau tua dan hijau muda. Kadang-kadang, bentuk daun juga mengalami perubahan, seperti menjadi cekung, keriting, atau memanjang.
2	Layu Fusarium	Daun mulai menguning dan layu dari bagian atas tanaman (Syarifudin, 2020)	Gejala secara bertahap akan menyebar ke seluruh tanaman, menyebabkan kelayuan permanen meskipun daun tetap menempel.
3	Virus Kuning	Daun akan mengalami pемutihan pada tulang daun, diikuti dengan perubahan warna menjadi kuning. Selain itu,	Penyakit ini ditularkan oleh kutu kebul yang berasal dari tanaman yang sudah terinfeksi, sehingga populasi kutu kebul yang tinggi dapat



NO	Nama Penyakit	Gejala	Keterangan
		tulang daun akan menebal, dan daun akan mengeriting ke arah atas (Alamsyah & Nonci, 2023)	mempengaruhi penyebaran penyakit ini.
4	Bercak Daun	Bercak kecil berbentuk bulat disertai klorosis, yang berkembang menjadi nekrosis hingga membentuk lubang (Devianto et al., 2023)	Penyakit ini pertama kali menyerang daun-daun tua sebelum menyebar ke daun-daun muda yang berada di bagian pucuk atau ujung batang atau tangkai.

### 2.3.6 Pengendalian Penyakit Pada Daun Cabai

#### 1. Penyakit Virus Mosaik Cabai.

Untuk pengendalian penyakit virus mosaik pada daun cabai, pestisida yang dapat digunakan adalah insektisida untuk mengendalikan vektor (kutu daun) yang menyebarkan virus. Berikut adalah beberapa contoh insektisida yang dapat digunakan:

- a. Insektisida Sistemik : *Imidakloprid*, *thiamethoxam*, dan *acetamiprid*
- b. Insektisida Kontak : *Profenofos*, *deltametrin*, dan *sipermetrin*
- c. Insektisida Botani : Ekstrak bawang putih, ekstrak mimba (*Azadirachta indica*), dan piretrin (dari *chrysanthemum*).

Penggunaan insektisida harus dilakukan dengan tepat waktu, dosis, dan sasaran. Beberapa hal yang perlu diperhatikan meliputi:

- a. Penggunaan insektisida sebaiknya dilakukan ketika jumlah vektor (kutu daun) masih sedikit, agar dapat mencegah penyebaran virus.
- b. Rotasi penggunaan insektisida dengan mekanisme kerja yang berbeda dapat menghindari resistensi.

- c. Insektisida dengan kategori aman bagi musuh alami (*predator* dan *parasitoid*) lebih direkomendasikan untuk menunjang pengendalian hayati.
- d. Selain insektisida, dapat juga dilakukan pengendalian lain seperti penggunaan varietas tahan, sanitasi lahan, dan pengendalian vektor secara mekanis.

## 2. Penyakit Layu Fusarium.

Untuk mengatasi penyakit layu fusarium pada tanaman cabai, berbagai jenis pestisida dapat digunakan, terutama fungisida (Susanna et al., 2023). Berikut adalah beberapa contoh fungisida yang efektif dalam mengendalikan penyakit layu fusarium pada daun cabai.:

- a. Fungisida Sistemik: *Carbendazim, thiophanate-methyl*, dan *propiconazole*.
- b. Fungisida Kontak: *Mancozeb*, klorotalonil, dan tembaga oksiklorida.
- c. Fungisida Biologis: *Trichoderma spp*, *Bacillus subtilis*, *Pseudomonas fluorescens*.

Selain penggunaan fungisida, pengendalian penyakit layu fusarium pada cabai juga dapat dilakukan dengan:

- a. Menggunakan benih/bibit cabai yang bebas dari patogen penyebab layu fusarium.
- b. Memperbaiki drainase lahan dan menghindari genangan air.
- c. Menerapkan rotasi tanaman dengan tanaman non-inang Fusarium.
- d. Meningkatkan kesuburan tanah dengan pemberian bahan organik.
- e. Menggunakan varietas cabai yang toleran atau tahan terhadap Fusarium.
- f. Mengendalikan vektor serangga pembawa patogen, seperti thrips.
- g. Menerapkan sanitasi kebun dengan membersihkan sisa-sisa tanaman terinfeksi.

## 3. Penyakit Virus Kuning

Untuk pengendalian penyakit virus kuning pada daun cabai, pestisida yang dapat digunakan adalah Insektisida untuk Mengendalikan Vektor (Kutu Kebul/*Whitefly*)

yang menyebarkan virus (Pramono, 2019). Berikut adalah beberapa contoh insektisida yang dapat digunakan:

- a. Insektisida sistemik: *Imidakloprid, Thiamethoxam, Dinotefuran*.
- b. Insektisida kontak: *Profenofos, Deltametrin, Sipermetrin*
- c. Insektisida botani: Ekstrak bawang putih, Ekstrak mimba, Piretrin

#### 4. Penyakit Bercak Daun

Untuk mengatasi penyakit bercak pada daun cabai, dapat digunakan berbagai jenis pestisida. (Lestari & Aini, 2021) sebagai berikut:

- a. Fungisida Kontak: Tembaga oksida, Mankozeb, Klorotalonil, Propineb, Mancozeb.
- b. Fungisida Sistemik: Difenokonazol, Propineb, Azoksistrobin, Tebukonazol, Kresoksim-metil.
- c. Fungisida Biologis: *Bacillus subtilis, Trichoderma harzianum, Pseudomonas fluorescens*.

## 2.4 Pengolahan Citra

Pengolahan Citra Digital (*Digital Image Processing*) adalah cabang ilmu yang fokus pada teknik-teknik untuk memproses gambar, baik itu gambar statis seperti foto maupun gambar bergerak seperti video. Konsep "digital" mengacu pada penggunaan komputer untuk mengolah gambar secara digital.

RGB, singkatan dari *Red-Green-Blue*, adalah tiga warna dasar yang umumnya digunakan sebagai referensi untuk menghasilkan warna lain. Dari model RGB ini, warna dapat dikonversi menjadi kode numerik yang memastikan tampilan warna yang konsisten secara universal. Komputer menyusun informasi warna dalam model warna yang seragam, sehingga memudahkan pengolahan warna RGB. (Ratna, 2020).

### 2.4.1 Citra

Citra adalah gambaran visual dari objek atau situasi yang bisa dilihat oleh mata manusia atau alat penginderaan. Citra dapat berupa gambar dua dimensi atau data multidimensi yang merepresentasikan atribut-atribut tertentu dari objek atau



adegan tersebut. Dalam konteks umum, citra sering kali merujuk pada gambar dua dimensi yang terdiri dari piksel-piksel. Setiap piksel memiliki nilai numerik yang menggambarkan tingkat cahaya atau warna di posisinya. Citra bisa berupa citra hitam-putih (*grayscale*) yang menunjukkan intensitas cahaya dalam berbagai nuansa abu-abu, atau citra berwarna di mana setiap piksel memiliki komponen warna seperti merah, hijau, dan biru (RGB).

Selain itu, citra juga dapat berbentuk citra tiga dimensi (3D) yang mencakup informasi kedalaman (*depth*) atau citra multispektral yang merepresentasikan intensitas cahaya pada berbagai panjang gelombang. Citra multispektral sering digunakan dalam penginderaan jauh atau bidang pengolahan citra satelit untuk analisis dan pemetaan pemanfaatan lahan, vegetasi, dan banyak lagi.

Citra dapat berasal dari berbagai sumber, seperti kamera digital, pemindai (*scanner*), sensor citra, atau hasil dari proses pengolahan sebelumnya. Citra dapat dimanfaatkan dalam berbagai bidang, seperti fotografi, pengolahan gambar medis, pengenalan pola, visi komputer, grafika komputer, pemetaan, dan banyak aplikasi lainnya. Dalam pengolahan citra, algoritma dan teknik komputasi digunakan untuk memanipulasi, menganalisis, dan memahami citra, serta mengekstrak informasi yang berguna dari citra tersebut (Orisa & Hidayat, 2019).

#### 2.4.2 Komponen Citra

Komponen citra merujuk pada elemen-elemen yang membentuk sebuah citra. Dalam konteks pemrosesan citra dan grafika komputer, terdapat beberapa komponen utama yang sering dibahas. Berikut adalah beberapa komponen citra yang umum:

1. **Piksel (*Pixel*):** Piksel merupakan unit terkecil dalam sebuah citra. Setiap piksel memiliki nilai intensitas yang menunjukkan tingkat kecerahan atau warna di posisi tertentu dalam citra. Piksel-piksel ini tersusun dalam pola grid yang membentuk keseluruhan citra.
2. **Resolusi (*Resolution*):** Resolusi mengacu pada jumlah piksel yang terdapat dalam citra. Resolusi dapat diukur dalam piksel per inci (PPI)

untuk citra cetak atau piksel per unit panjang untuk citra digital. Semakin tinggi resolusi, semakin besar detail dan kejelasan citra.

3. Kedalaman Warna (*Color Depth*): Kedalaman warna merujuk pada jumlah bit yang digunakan untuk menyimpan informasi warna setiap piksel dalam citra. Kedalaman warna menentukan jumlah warna yang dapat direproduksi dalam citra. Contohnya, citra dengan kedalaman warna 8 bit per piksel memiliki 256 warna yang mungkin, sedangkan citra dengan kedalaman warna 24 bit per piksel (biasanya disebut citra Truecolor) dapat merepresentasikan hingga 16,7 juta warna.
4. Ruang Warna (*Color Space*): Ruang warna menggambarkan model matematis yang digunakan untuk merepresentasikan warna dalam citra. Beberapa ruang warna yang sering digunakan termasuk RGB (merah, hijau, biru), CMYK (sian, magenta, kuning, hitam), dan HSV (hue, saturation, value). Setiap ruang warna memiliki cara yang berbeda untuk menggambarkan dan mengatur warna.
5. Histogram: Histogram adalah gambaran grafis dari distribusi intensitas piksel dalam sebuah citra. Histogram membantu dalam menganalisis dan memahami karakteristik cahaya dan kecerahan dalam citra. Dengan menganalisis histogram, informasi seperti kontras, kecerahan, dan distribusi nilai piksel dapat diekstraksi.
6. Filter: Filter digunakan dalam pemrosesan citra untuk menerapkan transformasi tertentu pada piksel-piksel citra. Filter dapat digunakan untuk mengurangi kebisingan, memperbaiki tajamnya citra, mengubah tingkat kecerahan, atau menerapkan efek artistik. Contoh filter yang umum digunakan termasuk filter Gaussian, filter sharpening, dan filter emboss.

### 2.4.3 Citra Analog

Citra analog adalah citra yang dihasilkan dari sinyal analog yang bersifat kontinu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit. Citra analog diperoleh dari perangkat komputasi analog, seperti

kamera analog dan mata manusia. Contoh citra analog termasuk gambar yang dilihat oleh mata manusia dan foto atau video yang diambil dengan kamera analog. Sebaliknya, citra digital merupakan representasi intensitas cahaya dalam format grid dua dimensi. Citra ini dibentuk oleh elemen gambar (piksel) yang memiliki koordinat  $(x,y)$  dan amplitudo  $f(x,y)$ . Di sini, amplitudo  $f(x,y)$  menentukan intensitas suatu citra, sementara koordinat  $(x,y)$  menunjukkan posisi elemen dalam citra tersebut (Ratna, 2020).

#### 2.4.4 Citra Digital

Citra digital adalah gambar atau representasi dari suatu objek yang dapat diproses oleh komputer, sedangkan citra analog tidak dapat langsung diolah oleh komputer karena tidak dapat direpresentasikan dalam format digital. Agar citra analog dapat diproses oleh computer, maka perlu dikonversi menjadi citra digital terlebih dahulu. Citra digital adalah citra yang telah diproses oleh komputer, dan juga mencakup citra yang dihasilkan dari perangkat digital (Furqan et al., 2022). Citra digital mengacu pada representasi digital dari gambar atau visual. Ini dapat merujuk pada citra yang dibuat menggunakan perangkat lunak grafis atau hasil dari pemrosesan digital dari citra yang dihasilkan oleh perangkat keras seperti kamera digital atau pemindai.

Citra digital terdiri dari kumpulan piksel, di mana setiap piksel memiliki nilai numerik yang mewakili warna atau intensitas cahaya pada lokasi tertentu di gambar. Piksel-piksel ini dapat diatur dan dimodifikasi dengan menggunakan perangkat lunak pengolahan citra untuk menciptakan efek, mengubah warna, atau memperbaiki gambar. Citra digital memiliki beberapa keuntungan dibandingkan dengan citra analog. Mereka dapat dengan mudah disimpan, disalin, dan dibagikan dalam format digital. Selain itu, citra digital dapat diubah dengan lebih mudah dan cepat menggunakan perangkat lunak pengolahan citra untuk memperbaiki cacat atau meningkatkan kualitas gambar.

Perbedaan utama antara citra analog dan citra digital terletak pada sifat kontinu dan diskritnya. Citra analog bersifat kontinu, seperti gambar pada layar televisi, foto sinar-X, foto yang dicetak di kertas foto, lukisan pemandangan, hasil

*CT scan*, dan gambar-gambar yang terekam pada pita kaset. Citra analog tidak bisa langsung direpresentasikan di komputer dan memerlukan proses konversi dari analog ke digital agar dapat diproses oleh komputer.

Sebaliknya, citra digital adalah citra yang dapat diolah oleh komputer dan terdiri dari sinyal yang terdiskritkan, yakni titik-titik warna yang membentuk gambar. Hasil perekaman citra digital dapat disimpan pada media penyimpanan dan diproses langsung di komputer. Secara umum, citra digital dibagi menjadi tiga jenis berdasarkan kombinasi warna pada piksel: citra RGB, citra grayscale, dan citra biner.

### 1) Citra RGB

Citra RGB, atau citra warna, adalah model warna aditif yang terdiri dari tiga kanal warna: merah, hijau, dan biru. Setiap kanal warna memiliki nilai intensitas piksel dengan kedalaman bit 8-bit, yang memungkinkan variasi warna sebanyak 256 tingkat (0 - 255). Seperti yang ditunjukkan pada Gambar 2.7 di bawah ini.



Gambar 2.7 Representasi citra RGB dan kanal warna penyusunan  
Sumber : (Marleny, 2021)

### 2) Citra *Grayscale* (Skala Keabuan)

Citra *grayscale*, atau *graylevel*, terdiri dari warna-warna piksel dalam rentang gradasi antara hitam dan putih. Citra ini juga dikenal sebagai derajat keabuan karena meliputi berbagai nuansa abu-abu antara warna hitam (minimum) dan putih (maksimum). Dalam citra *grayscale* 8-bit, rentang dari hitam hingga putih dibagi menjadi 256 tingkat abu-abu, di mana warna hitam sempurna diwakili oleh nilai 0 dan putih sempurna oleh nilai 255. Seperti yang terlihat pada Gambar 2.8.



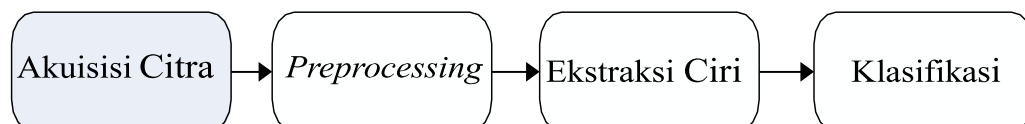
Gambar 2.8 Citra hasil konversi RGB menjadi grayscale  
Sumber : (Marleny, 2021)

Proses konversi ke *grayscale* merupakan langkah awal dalam pengolahan citra karena menyederhanakan model citra, sehingga mempermudah perhitungan intensitas abu-abu. Ada dua metode untuk mengubah citra berwarna menjadi citra *grayscale*: pertama, dengan menggunakan pembobotan nilai warna merah (R), hijau (G), dan biru (B), dan kedua, dengan menghitung nilai rata-rata dari ketiga nilai warna R, G, dan B (Yulina, 2021).

### 3) Citra Biner

Citra *grayscale* dapat diubah menjadi citra biner melalui proses *thresholding*. Dalam proses ini, diperlukan nilai *threshold* sebagai batas konversi. Pixel dengan intensitas yang lebih besar dari atau sama dengan nilai *threshold* akan dikonversi menjadi 1, sedangkan pixel dengan intensitas di bawah nilai *threshold* akan dikonversi menjadi 0.

Pengolahan citra digital melibatkan beberapa tahapan. Secara diagramatik, proses pengolahan citra digital dimulai dari Akuisisi Citra, dilanjutkan dengan *Preprocessing* (perbaikan kualitas citra), dan diakhiri dengan representasi citra, seperti yang ditunjukkan pada Gambar 2.9:



Gambar 2.9 Proses pengolahan citra secara diagram  
Sumber : (Melati, 2020)



### 1) Akuisisi

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari suatu citra yang tampak (foto, gambar, lukisan, patung, pemandangan) menjadi citra digital (Hasugian & Zufria, 2019). Alat yang digunakan untuk pencitraan adalah :

- a. Video
- b. Kamera Digital
- c. Kamera Konvensional dan Konverter analog to digital
- d. Scanner
- e. Photo Sinar-X / Sinar Inframerah

### 2) *Preprocessing*

*Preprocessing* citra adalah proses pengolahan data citra sebelum dianalisis lebih lanjut. Tahapan ini mencakup pembersihan *noise* dari citra, perubahan format warna, serta deteksi tepi dan sudut dalam citra. Beberapa proses dalam *preprocessing* termasuk komposit, pemotongan (*cropping*), dan penyusunan mozaik citra. Proses *preprocessing* meliputi akuisisi data, penyesuaian histogram citra, pemotongan dan pengubahan ukuran (*resize*), serta segmentasi ROI (*Region of Interest*). Untuk ekstraksi fitur, metode *Gray Level Co-occurrence Matrix* (GLCM) digunakan.

Tahap *preprocessing* data dilakukan setelah pengumpulan dan analisis data yang diperlukan. Untuk mendapatkan model yang sangat akurat, proses *preprocessing* data sangat penting. Dataset dari berbagai sumber sering mengandung kesalahan, nilai yang kosong, dan ketidak konsistenan, sehingga diperlukan *preprocessing* untuk mengubah data tersebut menjadi data yang dapat diolah dengan data *mining* (Sabda & Suhardi, 2023).

### 3) Ekstraksi Ciri

Ekstraksi ciri adalah proses mengambil informasi dari pola untuk mempermudah klasifikasi. Tujuan ekstraksi ciri adalah untuk memperoleh

karakteristik yang membedakan dan mewakili sifat utama, sambil menghilangkan fitur yang tidak relevan dalam proses klasifikasi. Ekstraksi ciri dilakukan dengan menghitung jumlah titik atau piksel yang ditemukan dalam setiap pemeriksaan, yang dilakukan dalam berbagai arah koordinat kartesian pada citra digital yang dianalisis, termasuk vertikal, horizontal, diagonal kanan, dan diagonal kiri (Zebua, 2020).

## 2.5 Klasifikasi Citra

Klasifikasi adalah proses mengelompokkan atau menempatkan objek atau data ke dalam kategori atau kelas yang telah ditentukan sebelumnya berdasarkan atribut-atribut yang dimilikinya. Tujuan klasifikasi adalah untuk mengidentifikasi pola atau karakteristik yang membedakan objek atau data antara kelas-kelas yang berbeda. Ada dua tugas utama dalam klasifikasi: (1) membangun model sebagai prototipe untuk disimpan sebagai memori, dan (2) menggunakan model tersebut untuk menganalisis, mengklasifikasikan, dan memprediksi objek data lainnya sehingga dapat dipahami dalam konteks model objek yang telah ada sebelumnya (Putro et al., 2020).

UNIVERSITAS ISLAM NEGERI  
SUMATERA UTARA MEDAN

### 2.5.1 Ekstraksi Tekstur

Pada tahap ini, citra yang telah diubah menjadi data Latih dan data Uji akan melalui proses ekstraksi ciri menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM), yang menghasilkan empat fitur utama. Fitur-fitur ini meliputi kontras, korelasi, energi, dan homogenitas, dengan penggunaan parameter offset [0 1]. Penggunaan parameter offset dipilih karena citra cabai merah dilihat dari sudut daun, yang mempermudah analisis (Wirayudhana, 2021).

*Gray Level Co-Occurrence Matrix* (GLCM) adalah metode statistik yang digunakan untuk mendeskripsikan karakteristik tekstur dalam citra digital. Metode ini menganalisis distribusi spasial nilai keabuan (*gray level*) dalam citra (Novianto & Erawan, 2020). GLCM adalah *contrast*, *correlation*, *energy* dan *homogeneity*.

Prinsip kerja GLCM adalah sebagai berikut:

1. Citra dikonversi menjadi citra keabuan (*grayscale*) dengan rentang nilai piksel tertentu, misalnya 0-255.
2. Matriks *co-occurrence* (kebersamaan) dihitung untuk suatu arah dan jarak spasial tertentu. Matriks ini menunjukkan seberapa sering suatu nilai keabuan tetangga muncul dalam citra.
3. Berbagai fitur statistik dihitung dari matriks *co-occurrence*, seperti kontras, energi, entropi, homogenitas, dll. Fitur-fitur ini kemudian digunakan untuk mendeskripsikan tekstur citra.

*Contrast* : mengukur intensitas kontras antara piksel dan tetangganya di atas seluruh gambar, mengevaluasi variasi lokal.

$$\text{Contrast} = \sum \sum (i - j)^2 p_{ij} \quad (2.1)$$

Ini berkisar dari 0 (untuk gambar konstan) ke akar kuadrat rata-rata ukuran GLCM -1.

*Correlation* : mengukur bagaimana sebuah piksel berkorelasi dengan tetangganya seluruh gambar, mengevaluasi kemungkinan terjadinya gabungan dari pasangan piksel yang ditentukan.

$$\text{Correlation} = \sum \sum (i - \mu_i)(j - \mu_j) \sqrt{(\sigma_i^2)(\sigma_j^2)} p_{ij} \quad (2.2)$$

Di mana  $\mu$  dan  $\sigma$  adalah nilai rata-rata dan standar deviasi. Ini berkisar antara -1 dan 1, yang berdiri dengan sempurna citra negatif atau berkorelasi positif. *Energy*, juga dikenal sebagai keseragaman atau momen kedua angular: itu mengembalikan jumlah elemen kuadrat di GLCM.

$$\text{Energy} = \sum \sum p_{ij}^2 \quad (2.3)$$

Ini berkisar dari 0 hingga 1, menjadi 1 nilai untuk gambar konstan.

*Homogeneity* : mengukur kedekatan distribusi elemen dalam GLCM ke GLCM diagonal.

$$\text{Homogeneity} = \sum \sum \frac{1}{1 + (i - j)^2} p_{ij} \quad (2.4)$$

Ini berkisar dari 0 hingga 1. Homogenitas adalah 1 untuk GLCM diagonal

Contoh perhitungan GLCM :

Misalkan memiliki citra input 4x4 seperti berikut :

0 0 1 1

0 0 1 1

0 2 2 2

2 2 3 3

Dengan parameter: 4 level keabuan (0-3), arah  $0^\circ$  (horizontal ke kanan), jarak 1 piksel.

1. Buat GLCM Kosong :

0 0 0 0

0 0 0 0

0 0 0 0

0 0 0 0



2. Hitung Frekuensi Pasangan :

a. (0,0) muncul 2 kali

b. (0,1) muncul 2 kali

c. (1,1) muncul 2 kali

d. (2,2) muncul 3 kali

e. (2,3) muncul 1 kali

f. (3,3) muncul 1 kali

3. Isi GLCM :

2 2 0 0

0 2 0 0

0 0 3 1

0 0 0 1

UNIVERSITAS ISLAM NEGERI

SUMATERA UTARA MEDAN

4. Normalisasi GLCM :

Jumlah total pasangan = 11

2/11 2/11 0 0

0 2/11 0 0

0 0 3/11 1/11

0 0 0 1/11

5. Hitung Fitur Tekstur :

Sebagai contoh, untuk menghitung *contrast* dapat dilihat pada persamaan dibawah ini :

$$Contrast = \sum_{i,j} |i-j|^2 \cdot p(i,j)$$

Di mana  $p(i,j)$  adalah elemen GLCM yang dinormalisasi.

$$\begin{aligned}
 \text{Kontras} &= (0-0)^2(2/11) + (0-1)^2(2/11) + (1-1)^2(2/11) + (2-2)^2(3/11) + (2-3)^2(1/11) + \\
 &(3-3)^2(1/11) \\
 &= 0 + 2/11 + 0 + 0 + 1/11 + 0 \\
 &= 3/11 \approx 0.2727
 \end{aligned}$$

Jadi, hasil dari citra input 4x4 yaitu  $3/11 \approx 0.2727$  (Nabella et al., 2019).

## 2.6 Identifikasi Penyakit Pada Daun Cabai

Dalam melakukan identifikasi terhadap penyakit antraknosa pada cabai merah, ada beberapa Langkah yang harus diperhatikan seperti berikut ini :

1. Observasi Gejala : Pertama – tama, dilakukan observasi visual terhadap tanaman cabai merah yang diduga terinfeksi penyakit antraknosa, kemudian perhatikan gejala yang muncul pada buah cabai merah.
2. Pengambilan Sampel : Jika terdapat gejala yang mencurigakan, maka ambil sampel buah cabai merah yang terinfeksi. Sampel ini akan digunakan untuk menganalisis lebih lanjut di laboratorium atau oleh ahli penyakit tanaman.
3. Analisis Laboratorium : Sampel yang diambil kemudian dianalisis di laboratorium untuk mengidentifikasi penyebab penyakit.
4. Identifikasi Patogen : Berdasarkan hasil analisis laboratorium, patogen penyebab penyakit antraknosa dapat diidentifikasi.
5. Evaluasi Kerusakan dan Penyebaran : Selain mengidentifikasi patogen, penting juga untuk mengevaluasi tingkat kerusakan yang disebabkan oleh penyakit antraknosa dan memahami cara penyebarannya. Hal ini dapat membantu dalam mengambil 20indaka-langkah pengendalian yang tepat.
6. Pengendalian Penyakit Antraknosa : Setelah identifikasi penyakit antraknosa, langkah-langkah pengendalian yang sesuai dapat diambil, seperti penggunaan pestisida, pengolahan lingkungan, penerapan praktik sanitasi, pemilihan varitas tanaman yang tahan terhadap penyakit, atau penggunaan metode biologi atau budaya seperti penggunaan rotasi tanaman.



## 2.7 Algoritma

Algoritma adalah metode yang digunakan untuk menyelesaikan masalah tertentu. Algoritma merupakan metode untuk mengevaluasi program yang diungkapkan dalam bahasa yang dapat dimengerti (Alda & Rifki, 2022). Algoritma juga termasuk jenis bahasa yang ketat dan eksplisit yang digunakan untuk memecahkan masalah. Lebih lanjut, algoritma pemrograman terdiri dari seperangkat aturan yang ditulis secara jelas dan ringkas untuk menyelesaikan suatu masalah tertentu. Namun jika dipahami dalam konteks teknologi informasi, maka program adalah sebuah aplikasi yang digunakan untuk melakukan pembelajaran atau permasalahan, seperti aplikasi memo, desain grafis, perhitungan, dan aplikasi lainnya (Pemrograman, 2020).

## 2.8 *K-Nearest Neighbor* (K-NN)

Algoritma *K-Nearest Neighbor* (K-NN) adalah metode klasifikasi yang mengelompokkan objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut, sesuai dengan jumlah  $k$  yang ditentukan, dan mengklasifikasikan ke dalam kelas baru (Furqan et al., 2022). K-NN termasuk dalam algoritma supervised learning, di mana hasil dari instansi query yang baru diklasifikasikan berdasarkan mayoritas kategori dalam K-NN. Kelas yang paling sering muncul akan menjadi hasil klasifikasi. Tujuan utama dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan sampel pelatihan.

*K-Nearest Neighbor* (KNN) merupakan suatu pendekatan untuk menghitung kedekatan antara kasus baru dengan kasus lama berdasarkan pencocokan bobot sejumlah fitur yang ada (Putri et al., 2024). K-NN melakukan klasifikasi dengan memproyeksikan data pembelajaran ke ruang berdimensi tinggi, yang dibagi menjadi bagian-bagian yang mewakili kriteria data pembelajaran. Setiap data pembelajaran direpresentasikan sebagai titik dalam ruang dimensi tersebut.

Algoritma *K-Nearest Neighbor* atau biasa disebut KNN merupakan salah satu metode klasifikasi data yang cara kerjanya relatif lebih sederhana dibandingkan dengan metode klasifikasi data lainnya. Algoritma ini mencoba mengklasifikasikan

data baru yang belum diketahui kelasnya dengan cara memilih sejumlah  $K$  data yang paling dekat dengan data baru tersebut. Kelas terbesar dari  $K$  data terdekat tersebut dipilih sebagai kelas prediksi untuk data baru tersebut.  $K$  pada umumnya ditentukan oleh banyaknya jarak yang sama dalam proses klasifikasi. Algoritma *K-Nearest Neighbor* (KNN) merupakan metode klasifikasi yang *robust* terhadap data latih yang berisik dan efektif ketika data berukuran besar (Safitri & Fakhriza, 2024).

Algoritma *K-Nearest Neighbor* (K-NN) melakukan klasifikasi dengan memproyeksikan data pembelajaran ke dalam ruang berdimensi tinggi, di mana setiap dimensi mewakili fitur dari data tersebut. K-NN adalah metode *supervised learning*, yang berbeda dengan *unsupervised learning*, di mana *supervised learning* bertujuan menemukan pola baru dengan menghubungkan pola data yang sudah ada dengan data baru, sementara *unsupervised learning* bertujuan menemukan pola dalam data yang belum memiliki pola yang jelas.

Sebagai metode *supervised learning*, K-NN mengklasifikasikan sampel uji baru berdasarkan mayoritas kategori dari K-NN. Ketepatan K-NN dipengaruhi oleh keberadaan data yang tidak relevan atau jika bobot fitur tidak sesuai dengan relevansi terhadap klasifikasi. Selain digunakan untuk analisis klasifikasi, K-NN juga telah diterapkan dalam prediksi dalam beberapa dekade terakhir. K-NN termasuk dalam kelompok instance-based learning dan merupakan salah satu teknik lazy learning. (Admojo & Ahsanawati, 2020).

### 2.8.1 Cara Kerja Algoritma *K-Nearest Neighbor* (K-NN)

Tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel pelatihan. Pengklasifikasian tidak melibatkan penggunaan model tertentu untuk pencocokan, melainkan hanya bergantung pada memori. Ketika diberikan titik uji, algoritma akan mencari sejumlah  $k$  objek (titik pelatihan) yang paling dekat dengan titik uji tersebut. Klasifikasi dilakukan dengan cara melakukan voting terbanyak di antara kelas-kelas dari  $k$  objek tersebut. Algoritma K-NN menggunakan ketetanggaan untuk menentukan nilai prediksi dari sampel uji yang baru. Jarak antara tetangga biasanya dihitung menggunakan jarak Euclidean.

Berdasarkan penelitian yang dilakukan oleh (Ardianti, 2023) K-NN adalah metode yang dapat memberikan informasi lebih akurat dan mengurangi kesalahan yang mungkin terjadi jika bergantung pada indra manusia untuk menentukan kualitas terbaik. Algoritma K-NN sangat sederhana dan bekerja dengan cara mengukur jarak terpendek antara sampel uji dan sampel pelatihan untuk menentukan tetangga terdekat (K-NN). Setelah mengidentifikasi K-NN, mayoritas kelas dari K-NN tersebut diambil untuk membuat prediksi mengenai sampel uji. Data untuk algoritma K-NN terdiri dari berbagai atribut *multivariat*  $X_i$  yang akan digunakan untuk mengklasifikasikan  $Y$ . Data dalam K-NN bisa berada dalam berbagai skala, mulai dari ordinal hingga nominal.

Algoritma *K-Nearest Neighbor* menggunakan prinsip klasifikasi ketetanggaan untuk memprediksi nilai dari instansi *query* yang baru. Algoritma ini bekerja dengan menentukan jarak terpendek dari instansi *query* ke sampel pelatihan untuk mengidentifikasi tetangga terdekatnya. Langkah-langkah dalam metode *K-Nearest Neighbor* meliputi:

- a. Menentukan parameter  $k$
- b. Menghitung jarak antara data yang akan dievaluasi dan semua pelatihan
- c. Mengurutkan jarak yang telah dihitung
- d. Menentukan jarak terdekat hingga urutan  $k$
- e. Mengidentifikasi kelas yang bersesuaian untuk jarak tersebut
- f. Menghitung jumlah kelas dari tetangga terdekat dan menetapkan kelas tersebut sebagai kelas untuk data yang akan dievaluasi.

### 2.8.2 Perhitungan Jarak (*dissimilarity*)

Perhitungan *dissimilarity*, atau yang juga dikenal sebagai jarak antar data ( $d$ ), merupakan aspek penting dalam algoritma *k-Nearest Neighbor* (K-NN). Terdapat berbagai metode untuk menghitung *dissimilarity*, salah satunya adalah:

- a. Jarak Manhattan

Dinamakan Manhattan karena didasarkan pada tata kota Manhattan yang tersusun dalam blok-blok persegi, sehingga sering disebut juga sebagai *City*

*Block Distance, Absolute Value Distance, atau Boxcar Distance.* Rumus untuk menghitung jarak Manhattan adalah sebagai berikut:

$$nd(x, y) = \sum |x_i - y_i| \quad i = 1 \quad (2.5)$$

Persamaan jarak Manhattan hanya melibatkan penjumlahan selisih absolut  $x_i$  dan  $y_i$ .

b. Jarak Euclidean

Metode paling umum digunakan dalam algoritma K-NN untuk menghitung jarak adalah jarak Euclidean. Rumusnya adalah sebagai berikut:

$$n(x, y) = \sqrt{\sum (x_i - y_i)^2} \quad i = 1 \quad (2.6)$$

dimana :

$x_i$  : data sampel

$y_i$  : data uji atau data *testing*

$i$  : variabel data

$d(x,y)$  : *dissimilarity*/jarak

$n$  : dimensi data

c. Jarak Minkowski

Persamaan ini adalah bentuk umum dari jarak Euclidean, di mana operasi kuadrat dapat diganti dengan operasi nilai mutlak dan pangkat non-negatif lainnya.

Rumusnya adalah

$$n(x, y) = (\sum |x_i - y_i|^r)^{\frac{1}{r}} \quad i = 1 \quad (2.3)$$

dimana :

$x_i$  : data sampel

$y_i$  : data uji atau data *testing*

$i$  : variabel data

$d(x,y)$  : *dissimilarity*/jarak

$n$  : dimensi data

$r$  : parameter

Jarak Euclidean adalah kasus khusus dari persamaan ini dengan  $r = 2$ .

d. Jarak Chebychev

$$d(x, y) = \max^n |x_i - y_i| \quad (2.4)$$

Persamaan ini mencari jarak terbesar antara  $x_i$  dan  $y_i$ .

Contoh perhitungan manual rumus k-nn dapat dilihat pada Tabel 2.2 dibawah:

**Tabel 2.2** Contoh Perhitungan Manual K-NN.

Sumber : (Al-Khowarizmi & Muhammad Arifin, 2021).

Suhu (°C)	Batuk (1-5)	Nyeri Otot (1-5)	Penyakit
38.0	2	3	Flu
39.0	3	4	Covid-19
37.5	1	2	Flu
40.0	4	5	Covid-19
38.5	2	4	Demam Berdarah
39.5	3	5	Demam Berdarah

Data yang akan diklasifikasikan :

- a. Suhu: 38.5°C
- b. Batuk: 3
- c. Nyeri Otot: 3

Langkah – langkah Perhitungan Manual KNN:

1. Hitung jarak Euclidean: kita akan menghitung jarak Euclidean antara gejala pasien baru dengan pasien terdahulu.

Rumus jarak Euclidean untuk tiga fitur adalah:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

Hitung Jarak dengan Semua Data:

- a. Jarak ke Pasien 1 (Flu) :

$$d = \sqrt{(38.5 - 38.0)^2 + (3 - 2)^2 + (3 - 3)^2}$$

$$d = \sqrt{(0.5)^2 + (1)^2 + (0)^2} = \sqrt{0.25 + 1 + 0} = \sqrt{1.25} \approx 1.12$$

- b. Jarak ke Pasien 2 (Covid-19):

$$d = \sqrt{(38.5 - 39.0)^2 + (3 - 3)^2 + (3 - 4)^2}$$

$$d = \sqrt{(-0.5)^2 + (0)^2 + (-1)^2} = \sqrt{0.25 + 0 + 1} = \sqrt{1.25} \approx 1.12$$

- c. Jarak ke Pasien 3 (Flu):

$$d = \sqrt{(38.5 - 37.5)^2 + (3 - 1)^2 + (3 - 2)^2}$$



$$d = \sqrt{(1)^2 + (2)^2 + (2)^2} = \sqrt{1 + 4 + 1} = \sqrt{6} \approx 2.45$$

- d. Jarak ke Pasien 4 (Covid-19):

$$d = \sqrt{(38.5 - 40.0)^2 + (3 - 4)^2 + (3 - 5)^2}$$

$$d = \sqrt{(-1.5)^2 + (-1)^2 + (-2)^2} = \sqrt{2.25 + 1 + 4} = \sqrt{7.25} \approx 2.6$$

- e. Jarak ke Pasien 5 (Demam Berdarah):

$$d = \sqrt{(38.5 - 38.5)^2 + (3 - 2)^2 + (3 - 4)^2}$$

$$d = \sqrt{(0)^2 + (1)^2 + (-1)^2} = \sqrt{0 + 1 + 1} = \sqrt{2} \approx 1.41$$

- f. Jarak ke Pasien 6 (Demam Berdarah):

$$d = \sqrt{(38.5 - 39.5)^2 + (3 - 3)^2 + (3 - 5)^2}$$

$$d = \sqrt{(-1)^2 + (0)^2 + (-2)^2} = \sqrt{1 + 0 + 4} = \sqrt{5} \approx 2.24$$

2. Pilih  $k$  Tetangga Terdekat: misalkan kita memilih  $k = 3$ . Dari perhitungan jarak, tiga tetangga terdekat adalah:

- a. Pasien 1 (Flu) dengan jarak 1.12
- b. Pasien 2 (Covid-19) dengan jarak 1.12
- c. Pasien 5 (Demam Berdarah) dengan jarak 1.41

3. Voring Mayorias: UNIVERSITAS ISLAM NEGERI

- a. Dari tiga tetangga terdekat, terdapat 1 pasien dengan Flu, 1 pasien dengan Covid-19, dan 1 pasien dengan Demam Berdarah.
- b. Dalam kasus ini, hasil klasifikasi bisa ambigu karena ketiga penyakit memiliki jumlah tetangga yang sama. Namun, jika kita melihat jarak yang paling kecil, pasien baru lebih mungkin mengalami Flu atau Covid-19 (karena jarak terdekat adalah 1.12 untuk Flu dan Covid-19).

### 2.8.3 Konsep K-NN

*K-Nearest Neighbor* (KNN) adalah salah satu dari 10 metode data mining yang paling populer. KNN tergolong dalam klasifikasi *lazy learning* karena menunda proses pelatihan (atau bahkan tidak melakukan pelatihan) hingga ada data uji yang perlu ditentukan label kelasnya. Pada saat itu, algoritma K-NN akan dijalankan. Klasifikasi K-NN dilakukan berdasarkan kemiripan data baru dengan data yang telah ada (Mabayoje et al., 2019).

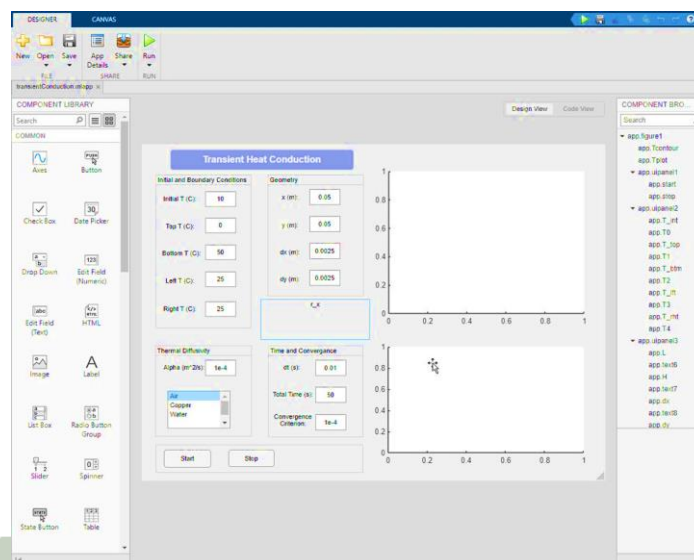
#### 2.8.4 Tujuan Algoritma K-NN

Tujuan dari algoritma K-NN adalah mengklasifikasikan objek baru berdasarkan atribut dan data latih. Algoritma ini tidak menggunakan model tertentu untuk dicocokkan, melainkan hanya mengandalkan memori. Ketika diberikan titik *query*, algoritma akan mencari sejumlah  $K$  objek atau titik latih yang paling dekat dengan *query* tersebut. Klasifikasi dilakukan dengan metode voting dari mayoritas kelas pada  $K$  objek yang terdekat. Algoritma K-NN bekerja secara sederhana dengan mengukur jarak terpendek antara *query instance* dan data latih untuk menentukan K-NN.

Pemilihan nilai  $K$  dalam algoritma ini bergantung pada data. Secara umum, semakin tinggi nilai  $K$ , semakin kecil pengaruh noise pada klasifikasi, tetapi batasan antar kelas menjadi lebih kabur. Nilai  $K$  yang optimal dapat ditentukan melalui optimasi parameter, seperti menggunakan cross-validation. Jika klasifikasi didasarkan hanya pada data latih yang paling dekat ( $K=1$ ), algoritma ini disebut sebagai *Nearest Neighbor* (D. A. Lestari & Mahdiana, 2021).

#### 2.9 Matlab GUI

MATLAB adalah perangkat lunak yang digunakan untuk pemrograman, analisis, serta komputasi teknis dan matematis yang berbasis matriks. Nama MATLAB merupakan singkatan dari *Matrix Laboratory*, karena program ini mampu menangani perhitungan dalam bentuk matriks. Awalnya, MATLAB dirancang untuk menyelesaikan persoalan-persoalan yang berkaitan dengan persamaan aljabar linear. Namun, seiring waktu, fungsionalitas dan kinerja komputasinya terus berkembang. (Putra & Rosiyanti, 2021).



**Gambar 2.10** Matlab GUI  
Sumber : (Fatwa et al., 2022)

MATLAB dapat dijalankan pada berbagai sistem operasi, seperti Windows, Linux, dan macOS. Selain itu, MATLAB juga dapat diintegrasikan dengan aplikasi atau bahasa pemrograman eksternal lainnya, seperti C, Java, .NET, dan Microsoft Excel. MATLAB juga dilengkapi dengan *toolbox* yang dapat digunakan untuk aplikasi khusus, seperti pengolahan sinyal, sistem kontrol, logika fuzzy, jaringan saraf tiruan, optimasi, pengolahan citra digital, bioinformatika, simulasi, dan berbagai teknologi lainnya.

GUI (*Graphical User Interface*) adalah salah satu komponen MATLAB yang memungkinkan pembuatan antarmuka grafis untuk menyelesaikan masalah matematika dengan cara yang lebih efisien dan menarik. Berbeda dengan m-file yang hanya dapat dijalankan di *Command Window*, GUI memungkinkan pembuatan *form* (lembar kerja) untuk setiap aplikasi dengan atribut-atribut yang disediakan oleh MATLAB (Murniasih et al., 2021).

## 2.10 Google Colaboratory (Colab)

Google Colab adalah layanan cloud computing yang disediakan oleh Google untuk mendukung pengembangan dan penelitian ilmiah. *Colaboratory*, atau “Colab” adalah produk dari Google *Research*. Colab memungkinkan untuk menulis dan mengeksekusi kode python arbitrer melalui *browser*, dan sangat cocok untuk

*machine learning*, analisis data, dan Pendidikan. Selain itu, Google colab bisa digunakan secara bersama-sama oleh pengembang aplikasi, sehingga sangat mendukung kebutuhan kolaborasi antar anggota tim. Google colab memang masih sangat awam bagi orang biasa karena hanya digunakan oleh orang yang membutuhkan *coding environment*, seperti para *developer* atau *programmer* (Febby Wilyani et al., 2024).

## 2.11 Python

Python merupakan bahasa pemrograman fungsional, procedural dan berorientasi objek yang dibuat oleh Guido Van Rossum pada akhir 1980-an. Saat ini, bahasa pemrograman ini banyak digunakan untuk aplikasi pada domain beragam, misalnya pengembangan perangkat lunak, pengembangan *web* dan *Guide User Interface* (GUI), dan pengembangan aplikasi desktop pendidikan dan aplikasi ilmiah. Python merupakan bahasa pemrograman yang banyak diminati karena sederhana, kecil, dan fleksibel. Bahasa pemrograman python terkenal karena kesederhanaannya dan ketangguhannya.







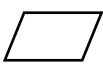



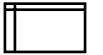





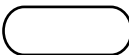

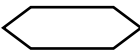

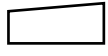



Beberapa keunggulan dari bahasa pemrograman python adalah serba guna, gratis dan sintaksisnya mudah. Sintaksis python yang mudah membuat proses belajar dan pemahaman mudah. Bahasa pemrograman ini tersedia di banyak *platform* termasuk Windows, Linux dan Mac OS. Bahasa pemrograman python digunakan untuk mengembangkan dan mendemostrasikan aplikasi dengan cepat karena bahasa pemrograman python mudah dipelajari. Kode program pada python ringkas dan efektif, karenanya dapat dimengerti dan dikelola.

Selain itu, bahasa pemrograman python menyediakan *library* bawaan yang banyak, sehingga bahasa pemrograman python dapat mempercepat pengembangan perangkat lunak. *Python Software Foundation* (PSF) mempercepat pertumbuhan bahasa pemrograman python. Bahasa serbaguna seperti python dapat digunakan tidak hanya untuk menulis *script* sederhana untuk menangani operasi file tetapi juga untuk mengembangkan perdagangan besar-besaran situs *web* Perusahaan (Anam et al., n.d., 2019).





## 2.12 Flowchart

*Flowchart* merupakan diagram atau gambar yang menggambarkan alur proses dan keterkaitan dalam suatu program. *Flowchart* diperlukan untuk menjelaskan urutan program secara visual, sehingga memudahkan orang lain dalam memahami alur program yang telah dibuat, seperti yang ditunjukkan pada Tabel 2.3 di bawah ini.

**Tabel 2.3** Simbol-simbol *Flowchart*  
Sumber: (Rosaly & Prasetyo, 2020)

Gambar	Fungsi	Gambar	Fungsi
	Proses		Card
	Proses Pilihan		Punched tape
	Keputusan		<i>Summing Junction</i>
	<i>Input</i> data dan <i>Output</i> informasi		Or
	Predefine Proses		Collate
	Internar Storage		Sort
	Dokumen		<i>Extract</i>
	Mutidokumen		Merge
	Terminator (mulai dan akhir)		Storage Data
	Preparasi		Delay
	Manual input		Sequential Access Storage
	Manual operasi		Magnetik Disk



Gambar	Fungsi	Gambar	Fungsi
	Penghubung		Direct Access Storage
	Off Page Penghubung		Display

### 2.13 Penelitian Terkait

Berbagai penelitian sebelumnya telah membahas secara umum tentang klasifikasi jenis cabai merah menggunakan algoritma *K-Nearest Neighbor* (K-NN). Contoh dan implementasi dari penelitian tersebut dapat dilihat pada Tabel 2.4 di bawah ini:

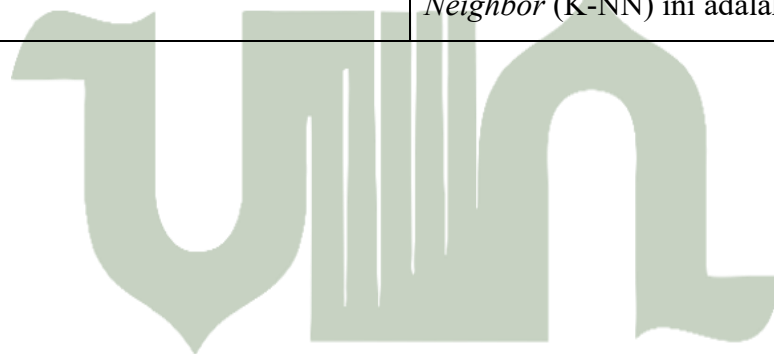
**Tabel 2.4** Penelitian Terkait

NO	Judul dan Tahun	Ringkasan
1	Klasifikasi Jenis Pisang Berdasarkan Fitur Warna, Tekstur, Bentuk Citra Menggunakan SVM dan KNN (Yana & Nafi'iyah, 2021)	Dari citra Pisang diambil fitur warna nilai rata-rata RGB, standar deviasi RGB, skewness RGB, entropy RGB. Fitur tekstur nilai rata-rata citra grayscale, standar deviasi grayscale, dan gray level cooccurance matrix (kontras, energi, korelasi, homogeneity). Serta fitur bentuk dari citra biner nilai area, perimeter, metric, major axis, minor axis, eccentricity. Hasil ujicoba menunjukkan algoritma SVM nilai akurasi mengklasifikasi jenis Pisang secara berturut-turut dari fitur warna, tekstur, bentuk adalah 41,67%, 33,3%, 8,3%. Dan hasil klasifikasi jenis Pisang dengan algoritma KNN, nilai K terbaik adalah

NO	Judul dan Tahun	Ringkasan
		2 pada fitur warna 55,95%, fitur tekstur 58,33%, dan fitur bentuk 45,24%
2	<p>Diagnosis Penyakit Cabai Menggunakan Metode Fuzzy <i>K-Nearest Neighbor</i>(FKNN) (Akbari et al., 2019)</p>	<p>FK-NN dapat diterapkan untuk diagnosis penyakit pada tanaman cabai melalui beberapa tahapan, yaitu menghitung jarak antara data latih dan data uji, memilih jarak terkecil dari data latih ke data uji sebanyak K, kemudian melakukan proses fuzzifikasi dan defuzzifikasi. Kelas dengan nilai defuzzifikasi tertinggi digunakan sebagai hasil klasifikasi. Nilai K mempengaruhi tingkat akurasi sistem, di mana semakin tinggi nilai K, akurasi cenderung menurun. Akurasi tertinggi yang diperoleh dari pengujian adalah ketika <math>K = 5</math>, dengan nilai mencapai 92%.</p>
3	<p>Klasifikasi Mutu Buah Jambu Biji Getas Merah Berdasarkan Tekstur Menggunakan <i>Grey Level Co-Occurence Matrix</i> (GLCM) dengan Klasifikasi KNN (Wirayudhana, 2021s)</p>	<p>Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa K-NN mampu mengidentifikasi dan mengklasifikasikan mutu buah jambu biji getas merah dengan menggunakan empat fitur, yaitu <i>Energy</i>, <i>Homogeneity</i>, <i>Contrast</i>, dan <i>Correlation</i> pada sudut 0 derajat. Persentase tertinggi dalam klasifikasi mutu buah jambu biji terjadi pada <math>K=9</math> dengan nilai 45,8%. Akurasi yang</p>

NO	Judul dan Tahun	Ringkasan
		<p>rendah ini kemungkinan disebabkan oleh tahap akuisisi data yang tidak konsisten dengan masing-masing kelas, yaitu kelas Super, kelas A, dan kelas B, sehingga menghasilkan akurasi yang rendah.</p>
4	<p>Ekstraksi Fitur Warna dan GLCM Pada Algoritma KNN Untuk Klasifikasi Kematangan Rambutan (Hadi &amp; Rachmawanto, 2022)</p>	<p>Berdasarkan penelitian yang dilakukan dengan menggunakan dataset citra rambutan untuk klasifikasi tingkat kematangan buah, akurasi tertinggi dicapai dengan <math>K=1</math>, menghasilkan nilai akurasi sebesar 98,75%. Sebaliknya, akurasi terendah diperoleh pada <math>K=7</math> dan <math>K=9</math>, dengan nilai akurasi sebesar 92,5%. Hasil ini menunjukkan bahwa semakin tinggi nilai <math>K</math> yang digunakan, semakin menurun akurasi klasifikasi. Penurunan akurasi ini disebabkan oleh jarak yang semakin jauh antara tetangga yang digunakan dalam proses klasifikasi.</p>
5	<p>Implementasi Metode <i>K-Nearest Neighbors</i> (KNN) Guna Mengetahui Klasifikasi Kematangan Stroberi (Widodo et al., 2023).</p>	<p>Berdasarkan analisis yang telah dilakukan, dapat disimpulkan sebagai berikut: 1) Aplikasi klasifikasi kematangan buah stroberi menggunakan metode <i>K-Nearest Neighbor</i> (K-NN) berhasil dikembangkan untuk membantu</p>

NO	Judul dan Tahun	Ringkasan
		<p>masyarakat dalam menentukan kematangan buah stroberi. 2) Dari pengujian terhadap 125 dataset, yang terdiri dari 71 data pelatihan dan 54 data pengujian, terdapat 13 kesalahan dalam klasifikasi. Dengan demikian, tingkat keberhasilan aplikasi yang menggunakan metode <i>K-Nearest Neighbor</i> (K-NN) ini adalah 76%.</p>



UNIVERSITAS ISLAM NEGERI  
SUMATERA UTARA MEDAN