

BAB II

DASAR TEORI

2.1 Tanaman Bayam

Bayam (*Amaranthus spp.*) merupakan sayuran yang banyak mengandung vitamin dan mineral, dapat tumbuh sepanjang tahun pada ketinggian sampai dengan 1000 mdpl dengan pengairan secukupnya [1]. Tanaman bayam merupakan tanaman yang cukup familiar dan mudah ditemui. Bahkan tanaman ini juga sering di tanam oleh para petani sayur di Indonesia.

Budidaya tanaman bayam. biasanya sudah tumbuh cukup besar dan siap untuk dipanen setelah 20 hari, sejak benih ditebar. Terlebih untuk budidaya bayam organik, mungkin bisa lebih lama dari biasanya. Namun dengan beberapa perlakuan tertentu, ternyata umur panen tanaman bayam bisa lebih cepat [2]. Tanaman bayam yang dijadikan objek penelitian dapat dilihat pada gambar 2.1 dibawah ini.



Gambar 2.1 Bayam

2.2 Penyakit Karat (*Puccinia spp.*)

Penyakit karat (*Puccinia spp.*) merupakan penyakit pada tanaman yang biasa menyerang jambu air, jambu biji, *beach cherry*, bayam, gandum, kacang tanah. Penyebab munculnya penyakit ini ialah karena organisme pengganggu yakni cendawan *Puccinia psidii*, *Puccinia aristidae*, *Puccinia triticina*, *Puccinia arrachidis*. Sementara gejala-gejala yang akan timbul bila tanaman terserang penyakit ini diantaranya muncul bercak-bercak cokelat di daun yang lama-kelamaan menyebar ke semua bagian tanaman. Sementara itu, penyebaran penyakit karat dibantu melalui angin, udara lembab dan percikan air hujan. Selain itu, benih dan bibit yang terkontaminasi juga menjadi sumber penyebaran ke tanaman sehat. Upaya pengendalian dapat dilakukan dengan mencabut dan memusnahkan tanaman

yang terserang sebelum menular, menggunakan fungisida berbahan aktif klorotalonil, tebukozanol, triadimefon, seperti 325 SC, Score 250EC atau dapat juga dengan mengatur jarak agar semua tanaman tersinari matahari dan memangkasnya secara teratur agar tidak ada bagian yang ternaungi. Berikut merupakan citra daun bayam yang terkena penyakit karat ditunjukkan pada gambar 2.2 [3].



Gambar 2.2 Citra Daun Bayam Terjangkit Karat

2.3 Pengolahan Citra

Pengolahan citra atau *image processing* adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan citra dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision* [4]. Adapun beberapa operasi dalam pengolahan citra diantaranya dijelaskan seperti berikut.

a. Mengubah ukuran atau dimensi citra (*resizing*)

Ukuran citra didefinisikan dengan format lebar (*width*) $\times$ tinggi (*height*) dengan satuan piksel [px]. Apabila lebar dikalikan dengan tinggi citra, maka jumlah hasil perkalian merupakan banyak piksel yang terdapat pada citra tersebut [5]. Proses mengubah ukuran citra ini bertujuan untuk mengubah dimensi citra asli ke citra dengan dimensi baru.

b. Konversi citra RGB ke abu-abu (*grayscale*)

Citra asli yang berwarna RGB dirubah menjadi citra (*grayscale*). Citra *grayscale* adalah citra yang hanya memiliki satu buah kanal sehingga yang ditampilkan hanyalah nilai intensitas atau dikenal juga dengan istilah derajat

keabuan. Secara teori ada beberapa cara dalam mengonversi citra berwarna RGB ke dalam citra *grayscale* [5].

c. Konversi citra *grayscale* ke citra biner (*thresholding*)

Mengubah citra berwarna abu-abu menjadi citra biner. Citra biner atau citra hitam putih adalah citra yang hanya memiliki 2 kemungkinan nilai untuk setiap pikselnya, yaitu 0 dan 1. Nilai 0 akan tampil sebagai warna hitam sedangkan nilai 1 akan tampil sebagai warna putih. Maka dari itu, jenis citra ini hanya membutuhkan 1-bit untuk menyimpan nilai pada setiap pikselnya. Jenis citra ini sering digunakan untuk proses *masking* ataupun proses segmentasi citra. Untuk mendapatkan citra biner maka dibutuhkan citra *grayscale* yang dilakukan *thresholding* terhadapnya berdasarkan nilai ambang batas (*threshold*) yang ditentukan. Jika nilai piksel pada citra *grayscale* melebihi atau menyamai nilai *threshold*, maka nilai piksel tersebut dikonversi menjadi 1. Namun jika nilai piksel kurang dari nilai *threshold*, maka nilai piksel tersebut dikonversi menjadi 0 [5].

d. Operasi Morfologi

Dalam konteks pengolahan citra digital, morfologi merupakan studi mengenai topologi dan struktur objek dari sebuah citra. Proses dalam morfologi mengacu pada operasi tertentu di mana sebuah objek dalam citra akan dikenali dengan elemen penstruktur dan dengan demikian akan mengurangi atau menambah bentuk suatu objek pada citra [5].

e. Ekstraksi Fitur

Ekstraksi fitur adalah salah satu cara yang dapat digunakan untuk mengenali suatu objek berdasarkan pada histogram khusus yang dimiliki objek tersebut. Ekstraksi fitur bertujuan untuk melakukan perhitungan serta perbandingan yang dapat dipakai untuk klasifikasi suatu citra berdasarkan ciri-ciri histogram yang dimiliki [6].

Fitur yang dapat dikenali menggunakan histogram seperti mean, standar deviasi, dan entropi [5].

2.4 SVM (*Support Vector Machine*)

SVM merupakan algoritma klasifikasi dengan tujuan untuk menemukan fungsi pemisah yang dapat memisahkan 2 set data dari 2 kelas yang berbeda [7].

SVM merupakan sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi – fungsi linier dalam sebuah fitur yang berdimensi tinggi dan dilatih dengan menggunakan algoritma pembelajaran yang didasarkan pada teori optimasi. SVM pertama kali diperkenalkan pada tahun 1992 oleh Vapnik sebagai rangkaian dari beberapa konsep – konsep unggulan dalam bidang pattern recognition.

Tingkat akurasi pada model yang akan dihasilkan oleh proses peralihan dengan SVM sangat bergantung terhadap fungsi kernel dan parameter yang digunakan [8]

2.5 Android Studio

Android Studio adalah IDE pemrograman Android resmi dari Google yang dikembangkan dari IntelliJ. Sebelum ada Android Studio, *programmer* Android telah menggunakan Eclipse. Eclipse adalah IDE pemrograman Android sebelum munculnya Android Studio. Bisa dibilang Google telah berpaling dari Eclipse dan menjadikan Android Studio sebagai IDE resminya.

Dikarenakan sudah meresmikan Android Studio pada tanggal 16 Mei 2013, Google menghentikan *support* ADT ke Eclipse tak lama kemudian dan ADT resmi hanya didapatkan oleh Android Studio. Dikembangkan di atas IntelliJ IDEA besutan JetBrains, Android Studio dirancang khusus untuk pengembangan Android. IDE Ini tersedia untuk digunakan pada sistem operasi Windows, Mac OS X dan Linux [9].

Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. Android studio juga terintegrasi dengan Android Software Development Kit (SDK) untuk *deploy* ke perangkat android [9].

2.6 OpenCV

OpenCV adalah sebuah *library computer vision* yang bersifat *open source* dengan kemampuan sebuah mesin dapat melihat hingga mampu mengekstrak informasi dari sebuah gambar. OpenCV dapat berjalan di berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan juga mendukung diberbagai *platform* seperti Windows, Linux, Mac OS, iOS dan Android [10].

Intel meluncurkan versi pertama dari OpenCV pada 1999, dan awalnya memerlukan library dari Intel *Image Processing Library*. Kemudian *dependency* tersebut akhirnya dihilangkan sehingga terciptalah OpenCV seperti yang sekarang sebagai *standalone library* [10].

2.7 Matlab

MATLAB merupakan bahasa pemrograman yang digunakan pada teknik-teknik komputasi, seperti penyelesaian matematik dan metode numerik, MATLAB menyediakan fasilitas-fasilitas untuk komputasi, visualisasi, pemrograman dan pengolahan database.

Selain itu MATLAB memiliki fitur yang dikelompokkan berdasarkan aplikasi tertentu yang dikenal dengan nama *tool box*. Dengan *tool box* ini pengguna diharapkan dapat mempelajari dan mengaplikasikan teknologi pada bidang kajian tertentu [11].

2.8 Bahasa Pemrograman Java

Java dikembangkan oleh Sun Microsystems pada tahun 1995. Java merupakan salah satu bahasa pemrograman yang berorientasi objek. Dengan bahasa pemrograman berorientasi objek maka seseorang dapat mengembangkan program dengan lebih mudah dari pada bahasa pemrograman terstruktur. Objek yang dibuat dapat diturunkan fungsinya pada kelas-kelas tertentu sesuai kebutuhan. Salah satu kelebihan java ialah *multiplatform*, dimana program yang dibuat dapat digunakan pada berbagai sistem operasi komputer seperti Windows, Linux, Mac Os dan Sun Solaris.

Beberapa keunggulan java yaitu java merupakan bahasa yang sederhana. Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa Java namun harus waspada karena mungkin Java mengambil arah (semantiks) yang berbeda dibanding C++ [12].

2.9 Pemrograman Berorientasi Objek

Pendekatan berorientasi objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur, setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar.

Lebih jauh lagi, pendukung OOP (*Object-Oriented Programming*) mengklaim bahwa OOP lebih mudah dipelajari bagi pemula. Dibandingkan dengan pendekatan sebelumnya, pendekatan OOP lebih mudah dikembangkan dan dirawat. Bahasa pemrograman yang menggunakan pendekatan OOP adalah Java, C#.NET, Objective-C, dan lain-lain [13].

2.10 UML

Unified Modeling Language atau yang disingkat UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak berorientasi objek. UML memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema basis data, dan komponen-komponen yang diperlukan dalam sistem perangkat lunak.

UML merupakan sebuah bahasa standar untuk pengembangan sebuah perangkat lunak yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan perangkat lunak. UML terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram tersebut menggambarkan permasalahan maupun solusi dari suatu model. UML mempunyai 9 diagram, yaitu *use-case*, *class*, *object*, *sequence*, *collaboration*, *activity*, *component* dan *deployment diagram* [13].