

BAB III

LANDASAN TEORI

3.1 Pengolahan Citra

Penglihatan adalah indra yang paling banyak memberikan informasi yang ada di sekitar. Citra menjadi sebuah media penting bagi penglihatan karena melalui citra, informasi dapat disampaikan kepada manusia. Hampir semua media sekarang didasarkan pada citra. Kemajuan pesat di bidang teknologi penginderaan memungkinkan manusia untuk mengambil sebuah citra dengan resolusi tinggi dan menghasilkan pengolahan citra dan video serta rekonstruksi sintesis berbasis citra (Barmoutis et al., 2016).

Era pengolahan citra diawali ketika industri surat kabar mulai menggunakan teknologi komputer untuk memaksimalkan citra yang diperoleh untuk mendampingi berita. Pengolahan citra menjadi sebuah fitur penting untuk memaksimalkan penyampaian informasi dan sesuai untuk berbagai profesi seperti industri film, militer, medis, pendidikan, seni grafis, dan percetakan. Pengolahan citra merupakan sebuah teknik meningkatkan kualitas citra mentah yang didapat dari sensor maupun kamera. Langkah umum dalam pengolahan citra dimulai dengan pemindaian citra, peningkatan kualitas citra, interpretasi, dan penyimpanan. Secara umum pengolahan citra dapat dikelompokkan menjadi dua jenis, yaitu analog dan digital. Pengolahan citra dengan cara analog melibatkan perubahan citra dengan bantuan variasi teknologi elektrik sedangkan pengolahan citra digital melibatkan representasi numerik untuk memanipulasi citra (Bhat, 2014).

3.2 Pengenalan Pola

Menurut penelitian yang dilakukan (Dutt et al., 2012) pengenalan pola merupakan disiplin ilmu yang bertujuan mengklasifikasikan sesuatu ke dalam beberapa kategori atau kelas. Pengenalan pola juga merupakan bagian integral di

banyak mesin kecerdasan yang dibangun untuk keperluan pengambilan keputusan (Dutt et al., 2012). Tujuan utama dari pengenalan pola ialah untuk menentukan kategori dari sebuah pola yang diberikan. Pengenalan sebuah pola dapat menghasilkan dua kategori, yaitu *supervised classification* apabila pola yang dimasukkan teridentifikasi sebagai anggota kelas yang telah ditentukan (*member of predefined class*) dan *unsupervised classification* apabila masukannya adalah pola yang belum terdaftar kelasnya pada sistem (*member of unknown class*) (Jain et al., 2000).

Pengenalan pola telah berkembang selama bertahun-tahun semenjak kemunculannya pada tahun 1960. Ada banyak metode yang mendorong pengembangan pengenalan pola di berbagai bidang pengaplikasian, diantaranya pengenalan pola statistik (*Statistical Pattern Recognition*) yang bekerja berdasarkan pada penyebaran vektor yang didapat dari probabilitas dan model statistik, pengelompokan data (*data clustering*) yang mencari beberapa kelompok serupa dalam sekumpulan data, jaringan syaraf tiruan yang menerapkan konsep biologis manusia kedalam mesin untuk mengenali pola. Ada lima tahapan pengenalan pola. Tahapan-tahapan yang dilakukan agar dari sebuah pola bisa didapatkan keputusannya ialah mengakuisisi pola, ekstraksi fitur, *pre-processing*, pengklasifikasian /regresi/deskripsi, dan *post-processing* (Dutt et al., 2012).

3.3 Jaringan Syaraf Tiruan *Backpropagation*

Jaringan syaraf tiruan merupakan sarana penting bagi penambahan data yang berfungsi untuk klasifikasi dan pengelompokan. Jaringan syaraf tiruan mengambil keputusan pengelompokan berdasarkan contoh yang diberikan. Ketika diberikan cukup contoh data, jaringan syaraf tiruan dapat membuat pengelompokan atau bahkan menemukan sebuah pola baru pada basis pengetahuan. Jaringan syaraf tiruan memiliki dua tipe, yaitu yang hasil keluarannya sudah diketahui dan tercatat dalam basis pengetahuan (*Backpropagation Algorithm*), dan yang hasil keluarannya belum diketahui (Amardeep, 2017).

Algoritma *Backpropagation* merupakan algoritma pembelajaran yang paling terkenal diantara algoritma jaringan syaraf tiruan lainnya. Pada algoritma *Backpropagation* bekerja dengan tiga buah lapisan. Lapisan pertama ialah *Input Layer*, lapisan ini menampung masukan untuk jaringan. Lapisan kedua, *Hidden Layer* berada diantara lapisan *input* dan *output* dan berfungsi sebagai titik aproaksi yang mengirim data dari lapisan sebelumnya ke lapisan selanjutnya. *Output Layer* adalah lapisan ketiga yang tugasnya menampung nilai keluaran yang menjadi pengenalan untuk masukan (Noor *et al.*, 2017).

3.4 Transformasi *Wavelet*

Wavelet merupakan suatu fungsi matematika yang membagi data menjadi beberapa komponen dengan frekuensi berbeda. Prinsip mendasar mengenai cara kerja *wavelet* ialah menganalisis sinyal pada skala atau resolusi berbeda, yang seringkali disebut multiresolusi. Salah satu fitur penting dari transformasi *wavelet* ialah dekomposisi multiresolusi. Transformasi *wavelet* memungkinkan citra yang sudah didekomposisi dapat direkonstruksi ulang dengan resolusi yang diinginkan (Sukanya and Preethi, 2013).

Wavelet dapat juga diartikan sebagai sebuah sinyal yang berada pada suatu skala dan waktu tertentu yang umumnya memiliki bentuk tak beraturan. Menggunakan *wavelet* memiliki banyak keuntungan. Kemampuannya untuk memisahkan detail halus dalam sebuah sinyal memungkinkan *wavelet* untuk digunakan pada sinyal berfrekuensi rendah dan meredam sinyal berfrekuensi tinggi. *Wavelet* yang digunakan pada frekuensi tinggi juga dapat dimanfaatkan untuk identifikasi detail kasar (Chowdhury and Khatun, 2012).

3.5 Peternakan Babi

Peternakan telah menjadi peran penting dalam usaha petani kecil dan penduduk lokal sejak lama sekali. Walaupun dibesarkan dengan masukan rendah namun peternakan masih dapat menghasilkan produknya dan memenuhi kebutuhan rumah tangga. Terkait dengan kenakekaragaman hayati, ternak merupakan *reservoir* gen yang bisa dijadikan aset untuk keperluan di masa mendatang. Peternakan mulai berkembang dari yang hanya sekedar memelihara di halaman belakang rumah hingga menjadi industri peternakan (Charoensook *et al.*, 2013).

Peternakan babi di Indonesia berkontribusi sebagai produksi daging nasional terbesar ketiga. Ditahun 2015 daging babi berkontribusi sebanyak 10,59% berada dibelakang daging sapi 17,38% dan ayam pedaging 23,97%. Peternakan babi masih memiliki potensi untuk dikembangkan lagi karena Indonesia memiliki beberapa faktor pendukung untuk mewujudkannya. Faktor pertama ialah dari segi sumber daya manusianya. Kebanyakan masyarakat Indonesia merupakan petani dan berpotensi mengembangkan peternakan babi. Potensi kedua ialah sumber daya alam yang mencukupi. Indonesia Timur memiliki banyak lahan yang dapat digunakan untuk mengembangkan bisnis peternakan babi, terlebih harga tanah yang masih murah untuk dijual atau disewakan. Faktor keempat adalah potensi dari penggunaan teknologi yang efisien seperti inseminasi buatan untuk meningkatkan produksi babi secara signifikan (Budaarsa, 2017).

3.6 Microsoft Visual Studio

Microsoft Visual Studio merupakan *Integrated Development Environment* (IDE) yang dibuat oleh Microsoft. IDE ini dapat digunakan untuk membuat atau mengembangkan berbagai aplikasi antarmuka pengguna grafis. Beberapa proyek yang dapat dibangun menggunakan Microsoft Visual Studio antara lain Windows *Forms*, situs web, aplikasi web, dan layanan web (*web services*) (Sharma, 2013). Microsoft Visual Studio mendukung editor kode IntelliSense serta usaha mengubah suatu struktur kode pada program agar kode program tersebut dapat lebih mudah dibaca dan dimodifikasi. *Plugin* dapat dimasukkan pada program untuk

meningkatkan fungsi di hampir seluruh *level*, termasuk dukungan untuk sistem *source-control* seperti *subversion* dan menambah *toolset* baru seperti editor dan desainer visual untuk bahasa tertentu (Heng *et al.*, 2015).

3.7 *Framework .NET*

Framework .NET merupakan komponen Windows yang terintegrasi yang diciptakan agar dapat membantu pengembangan berbagai jenis aplikasi termasuk aplikasi *Web Services XML* (Indarto, 2017). *Framework .Net* di desain untuk memenuhi beberapa tujuan:

1. Menyediakan lingkungan kerja yang konsisten bagi bahasa pemrograman yang sifatnya berorientasi objek.
2. Menyediakan lingkungan kerja ketika menjalankan kode dengan meminimalisi proses *software deployment* serta mengurangi kemungkinan konflik antara versi *software*.
3. Menyediakan lingkungan kerja yang aman.
4. Menyediakan lingkungan kerja yang mengurangi permasalahan performa.
5. Mempermudah pihak pengembang memvariasikan aplikasinya.
6. Membuat sebuah standar industri yang memastikan semua aplikasi berbasis *framework .NET* dapat berintegrasi.

3.8 *C#*

Bahasa pemrograman ciptaan Microsoft dalam masa kepemimpinan Anders Hejlsberg. Hejlsberg sebenarnya seorang pengembang yang membuat berbagai macam bahasa pemrograman seperti Borland Turbo, C++, dan Borland Delphi. Bahasa pemrograman C# telah mendapat standarisasi secara internasional oleh ECMA.

C# dapat digunakan untuk membangun berbagai jenis aplikasi, khususnya aplikasi berbasis Windows pada perangkat komputer dan aplikasi berbasis web dan *web services*. Penggunaan bahasa pemrograman C# pada *framework .NET* dinilai

tepat karena berbagai kelebihanannya dan karena konsepnya yang sudah menggunakan paradigma berorientasi objek (Indarto, 2017).

3.9 Android Studio

Android studio adalah lingkungan pengembangan terpadu (*Integrated Development Enviroment*) resmi untuk pengembangan aplikasi berbasis android. Berdasarkan perangkat lunak *JetBrains'intelliJ*, Android Studio diimplementasikan secara khusus untuk pengembangan android dan tersedia untuk sistem operasi Windows, Mac OS X, dan Linux (Rojatkar *et al.*, 2016). Aplikasi android biasanya ditulis dalam bahasa pemrograman *Java*, begitu pula Android Studio. Selama pengembangan, pengembang membuat dokumen yang berisi konfigurasi android dan menulis algoritma aplikasi dalam bahasa pemrograman *Java*. Android Studio mengubah dokumen aplikasi yang berbasis *Java* ini menjadi aplikasi android (Kala, 2014).

Android Studio memiliki kelebihan di desain atarmuka pengguna yang merespons transisi dengan cepat. Editor tata letak memungkinkan pengguna memasukkan komponen antarmuka dengan cara seret dan letak. Ada juga pilihan untuk pratinjau tata letak yang telah dibuat pada beberapa konfigurasi layar. Selain dari sisi antarmukanya, *Android Studio* juga mempermudah penggunaanya dalam memakai *library* yang dapat diintegrasikan pada *SDK* (Rojatkar *et al.*, 2016).

3.10 Java

Menurut Wong(2002) dalam (Bringula *et al.*, 2012), java merupakan salah satu bahasa pemrograman paling sukses di pasaran. Sun Microsystems(2008) dalam (Bringula *et al.*, 2012) menyatakan bahwa *Java* dianggap sebagai bahasa pemrograman tujuan umum yang mengurangi siklus kompilasi dan memungkinkan untuk dijalankan pada berbagai sistem operasi. Karena kian populer, pengembangan aplikasinya semakin diminati dan begitu juga para pengembang dan

pemrogram (Bringula *et al.*, 2012). Java merupakan bahasa pemrograman yang kuat dan banyak digunakan di berbagai peron (Gondchawar, 2015).

Bahasa pemrograman Java juga digunakan pada aplikasi android. Perbedaan *Java* pada aplikasi android ialah program tidak dieksekusi menggunakan *Java Virtual Machine*(JVM) standar. Sebagai pengganti JVM, Google menciptakan mesin virtual khusus bernama Dalvik yang berfungsi untuk mengeksekusi (Holla and Mahima, 2012)dan mengubah kode *byte* Java. Proses pengubahan kode dapat dilakukan secara otomatis maupun manual. Dalvik mengambil kelas java yang dihasilkan dan kemudian menggabungkannya menjadi satu atau lebih dokumen Dalvik *Executable* sehingga mengurangi kebutuhan ruang hingga setengahnya tanpa melakukan pengompresan (Holla and Mahima, 2012).

3.11 EmguCV

EmguCV merupakan sebuah *library* terbuka antar *platform* yang dapat memanggil fungsi *image processing* pada OpenCV. EmguCV kompatibel dengan bahasa pemrograman C# dan dapat digunakan pada Microsoft Visual Studio. Beberapa keuntungan dari menggunakan EmguCV yaitu dapat menggunakan fitur-fitur OpenCV pada platform lain yang tidak dapat menggunakan OpenCV, kompatibel dengan banyak bahasa pemrograman seperti C++, C#, VB, dan Phytion, serta kompatibel untuk dijalankan pada beragam sistem operasi seperti Windows, Linux, dan MacOS (Herusutopo *et al.*, 2012).

Pada proyek tugas akhir ini, EmguCV digunakan penulis untuk menggunakan fungsi OpenCV yang dapat memanipulasi gambar. EmguCV dipilih sebagai *wrapper* dari OpenCV agar dapat digunakan pada bahasa pemrograman C#. Bansode(2015) dalam (Bansode *et al.*, 2015) menyatakan bahwa *image processing* menggunakan EmguCV memiliki akurasi yang tinggi dan dapat diandalkan bahkan untuk penggunaan *real time* sekalipun.

3.12 OpenCV

Aplikasi yang memanfaatkan fitur kamera biasanya melibatkan metode pengolahan citra seperti *Gaussian*, *Median*, *Mean Laplacian*, *Sobel filter*, dan lain sebagainya. Pengembang yang memiliki pengetahuan dasar mengenai pengolahan citra dapat membuat kodenya sendiri dan menerapkannya pada aplikasi buatannya, namun sebagai pengembang yang tidak memiliki dasar pengetahuan pengolahan citra tentu akan disulitkan dalam proses pembuatannya. Untuk mengatasi masalah tersebut, pengembang melakukan impor *library* ke dalam pekerjaan mereka. Dibidang pengolahan citra terdapat sebuah *library* yang terbuka untuk digunakan bernama OpenCV (*Open Computer Vision*) (Anuar *et al.*, 2000).

OpenCV merupakan *library* yang dikembangkan *Intel* yang bertujuan untuk sistem penglihatan komputer real time. *Library* ini independen terhadap platform manapun. OpenCV ditulis dalam bahasa pemrograman C dengan antarmukanya yang tersedia dalam C. Selain C, OpenCV juga menyediakan antarmuka lengkap yang disediakan bagi bahasa pemrograman lain seperti Python, Java, Matlab, dan Octave. OpenCv bekerja dengan struktur data yang cepat dan efisien sehingga memungkinkan untuk dipakai pada aplikasi pengolahan Citra yang berjalan secara *Real Time* (Dhawan and Honrao, 2013).