

## DAFTAR ISTILAH

Piksel	: Titik terkecil dari sebuah gambar.
Intensitas	: Nilai di dalam sebuah piksel.
Tepi	: Sebuah titik pemisah antar objek pada suatu gambar.
Segmentasi	: Pemisahan suatu objek, dengan objek yang lainnya, pada suatu gambar.
K-means	: Salah satu metode <i>clustering</i> .
<i>Centroid</i>	: Titik pusat kelompok, dalam algoritma K-means.
<i>Threshold</i>	: Batas sebuah nilai.
CHT	: Circular Hough Transform, metode untuk mencari lingkaran dalam sebuah gambar.
<i>rMin</i>	: Parameter radius minimal untuk metode CHT.
<i>rMax</i>	: Parameter radius maksimal untuk metode CHT.
CNN	: Convolution Neural Network, salah satu metode yang dapat digunakan untuk klasifikasi.

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Iris biometrik merupakan pengidentifikasian pola unik dari sebuah iris individu [7], yang nantinya akan digunakan sebagai ciri unik untuk sistem verifikasi atau identifikasi. Secara umum tahapan identifikasi iris ada empat bagian yaitu *image acquisition*, segmentasi, ekstraksi fitur dan klasifikasi [8]. Dalam aplikasinya iris biometrik memiliki beberapa kendala, beberapa diantaranya yaitu adanya sebuah derau atau *noise* pada data yang dapat menyulitkan pemrosesan [5] dan kompleksitas dari perhitungan yang harus dipertimbangkan. Dengan adanya derau dalam suatu data, dapat membuat proses iris biometrik menjadi tidak stabil sehingga perlu diantisipasi dari awal pemrosesan data yaitu pada tahap segmentasi, yang dimana proses tersebut akan memisahkan iris dari objek lainnya.

Metode pendeteksian tepi yang biasa diimplementasikan untuk segmentasi masih memiliki beberapa kekurangan, yaitu tidak mampu melakukan segmentasi data yang memiliki derau [5]. Dengan adanya masalah tersebut, sebelum data diproses pendeteksi tepi, data harus diolah menggunakan metode yang dapat menangani derau. Metode yang akan digunakan untuk mengatasi derau tersebut adalah metode *clustering* K-means. Metode K-means dapat mengelompokkan setiap intensitas piksel pada gambar menjadi k kelompok dan nilai dari setiap kelompok akan dinormalisasi. Sehingga hasil dari metode K-means akan menghasilkan k objek dengan nilai intensitas didalam objek nya sama, yang berarti menghilangkan derau yang ada didalamnya. Dengan ditambahkannya metode K-means pada segmentasi, terbukti dapat meningkatkan akurasi segmentasi iris hingga mencapai 98% seperti pada penelitian yang dilakukan Rajeev *et al* [13] dan H.Procena *et al* [5].

Kemudian untuk metode klasifikasi, masalah yang ada adalah informasi mengenai klasifikasi iris ini sangat kekurangan sumber yang menggunakan metode berbasis *neural network*, khususnya dalam metode pengembangan dari *neural network* itu sendiri salah satunya adalah Convolutional Neural Network (CNN) yang memang telah dilakukan pengembangan dari metode *neural network* untuk klasifikasi berbasis *image* [9]. CNN sudah terbukti dapat mengidentifikasi dan melakukan klasifikasi suatu gambar dengan akurat, salah satunya dibuktikan pada penelitian identifikasi wajah manusia oleh Li, Haoxiang, et al [4].

Adapun informasi mengenai aplikasi CNN dalam masalah klasifikasi biometrik namun tidak menjelaskan secara detail seperti apakah aplikasinya dalam klasifikasi iris, seperti arsitektur yang dipakai dan seberapa besar akurasi [9].

### 1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah disampaikan, didapatkan beberapa perumusan masalah yang akan dikaji.

1. Bagaimana cara mensegmentasi pupil dari gambar mata.
2. Bagaimana membuat sistem klasifikasi iris.
3. Arsitektur seperti apakah yang dapat mengklasifikasikan data iris dengan akurat.

### 1.3 Tujuan

Berikut tujuan dari penelitian yang dilakukan.

1. Dapat mensegmentasi data iris dengan tepat, dengan menggunakan metode segmentasi, K-Means, Canny Edge Detection dan Circular Hough Transform.
2. Membuat sistem klasifikasi yang mampu menklasifikasi iris dengan tepat, menggunakan Convolutional Neural Network (CNN).
3. Menghasilkan dan menganalisis sistem jaringan CNN yang dapat menklasifikasikan iris dengan akurat.

### 1.4 Batasan Masalah

Dibawah ini merupakan batasan masalah untuk pengerjaan yang akan dilakukan.

1. Data yang digunakan merupakan *dataset* iris hasil uji coba iris scanner pada pegawai bandara dan *dataset* iris Casia iris lamp, berupa gambar iris 8 bit *grayscale* berukuran 640 x 480.
2. Data yang digunakan untuk *testing* merupakan gambar dari *dataset* yang sama dengan data *training*, namun gambar yang berbeda dengan *training*.
3. Gambar iris yang digunakan harus gambar yang di ambil dari arah depan seorang individu, tidak dari samping.
4. Pada tahap segmentasi, hanya lingkaran pupil yang dicari, sedangkan lingkaran iris, hanya akan diambil 36 *pixel* dari lingkaran pupil.
5. Pupil yang akan dideteksi diasumsikan berbentuk lingkaran.
6. Maksimum *epoch* pada pengujian arsitektur CNN hanya 2000 *epoch*.
7. Pengambilan 25 class pada tahap klasifikasi adalah 25 class dengan gambar iris yang berhasil disegmentasi seluruhnya.

### 1.5 Metodologi Penyelesaian

Metodologi penyelesaian dalam tugas akhir ini yaitu sebagai berikut:

1. Studi literatur  
Mencari dan memahami dari sumber-sumber yang berkaitan mengenai iris biometrik, pengolahan citra dan teknik klasifikasi. Adapun sumber tersebut didapat dari berbagai artikel, jurnal dan penelitian yang berkaitan dengan materi yang dituliskan.
2. Pengumpulan Data  
Data untuk iris biometrik yang akan digunakan dalam tugas akhir ini merupakan data yang diambil menggunakan iris *scanner* yang diuji coba pada pegawai bandara dan *dataset* Iris Casia Lamp.
3. Perancangan Sistem  
Merancang skenario yang akan dilakukan, akan dijelaskan alur rencana pengerjaan secara umum diawali dengan melakukan metode *clustering* K-means. Kemudian dilanjutkan dengan deteksi Canny dan ditransformasi menggunakan Circular Hough Transform sehingga menghasilkan lingkaran iris yang akan dipetakan ke data asli kemudian dinormalisasi dan akan menjadi *input* bagi algoritma CNN dengan jumlah *convolutional layer*, *pooling layer* dan *Fully Connected layer* akan divariasikan untuk mencari arsitektur dengan akurasi terbaik.
4. Implementasi

Proses pembuatan program sistem iris biometrik dengan skenario yang telah ditentukan.

5. Pengujian dan Analisis

Pengujian, pemaparan dan analisis hasil dari keseluruhan sistem.

6. Penulisan Laporan Tugas Akhir

Penulisan laporan secara lengkap yang terdiri dari pendahuluan, kajian pustaka, rancangan, serta analisis dan hasil.

## **BAB 2**

### **KAJIAN PUSTAKA**

#### **2.1 *Related Work***

Penelitian untuk sebuah klasifikasi iris sekarang ini sudah banyak dilakukan dan memiliki tujuan dan masalah yang berbeda-beda, beberapa diantaranya pada penelitian yang dilakukan Kevin *et al*, [8] yang menjelaskan dan mengenalkan proses klasifikasi iris. Jurnal tersebut menjelaskan bahwa iris biometrik memiliki keuntungan lebih dari segi akurasi untuk menklasifikasikan seorang individu, namun dalam proses klasifikasi tersebut, banyak sekali faktor yang harus dipertimbangkan, salah satunya adalah kesulitan dalam membentuk sistem klasifikasi iris yang ideal, dikarenakan banyaknya pemilihan metode dalam setiap tahap pemrosesan klasifikasi iris. Ditambah lagi adanya masalah derau pada gambar iris, sehingga dapat menurunkan akurasi pendeteksian lingkaran iris.

Masalah mengenai derau tersebut, dikaji kembali oleh Rajeev *et al* [13] dan H.Procena *et al*, [5] dari kedua peneliti tersebut disampaikan bahwa permasalahan derau pada gambar iris dapat diatasi dengan menggunakan metode tambahan pada tahap segmentasi, yaitu dengan menggunakan metode clustering k-means, yang dibuktikan dengan tingkat ketepatan segmentasi setelah digunakannya metode k-means tersebut, akurasi segmentasinya mencapai 98%.

Kemudian dari penelitian yang dilakukan oleh Maria *et al* [9] memiliki pandangan yang berbeda, jurnal tersebut lebih membahas pada tahapan fitur ekstraksi dan klasifikasi, menggunakan metode Artificial Neural Network (ANN) dan metode dari Deep Learning, yang masih jarang digunakan untuk klasifikasi iris. ANN sangat berpotensi untuk melakukan klasifikasi pada sebuah iris karena kemampuannya untuk membangun suatu jaringan terlatih yang dapat membedakan suatu ciri yang satu dengan lainnya, bahkan dengan digunakannya ANN, dapat diaplikasikan tanpa menggunakan metode fitur ekstraksi yang terpisah dengan metode klasifikasi. Terlebih lagi untuk beberapa algoritma pengembangan dari ANN, seperti algoritma Deep Learning, Convolutional Neural Network (CNN) dapat dibandingkan performanya dengan metode Support Vector Machine (SVM).

#### **2.2 *Iris Biometrik***

Iris biometrik adalah sebuah metode yang dapat mengenali pola karakteristik unik dari iris mata manusia, yang dimana setiap manusia memiliki pola unik atau ciri khusus yang berbeda-beda [8], dengan kemampuannya tersebut iris biometrik sekarang ini menjadi pusat perhatian untuk sistem identifikasi. Dalam aplikasinya iris biometrik dapat menjadi sebuah terobosan terbaru untuk menutupi kekurangan sistem verifikasi. Sistem verifikasi dengan iris biometrik memiliki kelebihan tersendiri, beberapa diantaranya:

1. Tidak akan hilang maupun lupa.
2. Memiliki kemungkinan kecil mengalami perubahan.
3. Dibutuhkan keahlian khusus untuk diretas dengan kata lain sulit untuk diretas.
4. Tingkat ketelitian melakukan identifikasi sangat tinggi.

Dalam prosesnya secara umum iris biometrik memiliki empat tahapan yaitu *image acquisition*, segmentasi, fitur ekstraksi dan klasifikasi [8]. *Image acquisition* adalah tahap pengambilan gambar mata seorang individu menggunakan sebuah alat

khusus dan yang biasanya digunakan adalah kamera inframerah. Setelah didapatkan sebuah gambar dilanjutkan dengan tahapan segmentasi, yang dimana proses tersebut adalah proses pemisahan antara iris mata dengan pupil dan sclera sehingga didapatkan hanya gambar iris mata saja, yang akan dinormalisasi. Selanjutnya gambar tadi akan di ambil pola atau ciri uniknya yang dapat mewakili kepemilikan gambar iris tersebut, yang pada akhirnya akan diklasifikasi untuk melakukan identifikasi, kepemilikan gambar tersebut.

Namun dalam implementasinya ke-empat tahapan diatas tidak semudah seperti yang dituliskan, setiap tahapan yang telah dituliskan memiliki metode yang berbeda-beda dan setiap kombinasi metode yang digunakan akan memiliki performansi yang berbeda juga [8]. Permasalahan yang banyak terjadi adalah adanya derau pada gambar yang didapatkan dari *image acquisition*, yang menyebabkan kekacauan pada tahapan-tahapan selanjutnya dan dapat dipastikan akan berdampak pada penurunan akurasi identifikasi iris tersebut.

Dari empat tahapan umum dan masalah yang telah dijelaskan sebelumnya, berikut beberapa metode yang akan digunakan. Pada tahapan segmentasi, serangkaian metode yang digunakan adalah metode K-means, deteksi tepi Canny Circular Hough Transform dan untuk klasifikasi, digunakan algoritma Convolutional Neural Network (CNN) yang akan dijelaskan lebih spesifik pada sub bab selanjutnya.

## 2.3 Segmentasi Iris

Segmentasi iris merupakan salah satu tahapan pada klasifikasi iris, yang di dalamnya memiliki serangkaian metode untuk memisahkan lingkaran iris dengan lingkaran pupil dan juga sclera, sampai didapatkan gambar hasil normalisasi dari lingkaran iris. Berikut metode-metode yang akan digunakan pada tahap segmentasi iris.

### 2.3.1 K-Means

Algoritma K-means adalah algoritma *unsupervised learning* yang dapat mengelompokkan data menjadi beberapa kelompok, dengan cara menghitung jarak sebuah data dengan titik pusat setiap kelompok, data yang memiliki jarak terpendek menuju titik pusat kelompok, akan menjadi anggota kelompok tersebut, implementasi pada pengolahan citra data yang akan dikelompokkan adalah nilai intensitas setiap piksel, pada gambar yang akan diproses, tujuan digunakannya algoritma K-means pada gambar adalah untuk membagi daerah pada gambar yang menunjukkan sebuah objek dengan nilai intensitas yang sama berikut tahapan pada algoritma K-means [13].

1. Menghitung distribusi dari intensitas setiap *piksel* pada gambar.
2. Memilih titik pusat secara acak, sejumlah  $k$  kelompok, berdasarkan intensitas *piksel*, untuk titik pusat awal  $\mu_1, \mu_2, \dots, \mu_k$ .
3. Mengelempokan intensitas berdasarkan jarak terdekat dengan titik pusat kelompok.

$$c^{(i)} := \arg \min_j \|x^{(i)} - \mu_j\|^2 \quad (2.1)$$

Penjelasan persamaan tersebut adalah dimana  $x^{(i)}$  adalah nilai intensitas poin  $c^{(i)}$  dan  $c^{(i)}$  tersebut merupakan poin ke i yang akan diisi dengan nilai j yang meminimumkan  $\|x^{(i)} - \mu_j\|^2$ .

4. Menghitung titik pusat baru untuk setiap kelompok.

$$\mu_i := \frac{\sum_{i=1}^m 1\{c_{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c_{(i)} = j\}} \quad (2.2)$$

Dimana  $\mu_i$  adalah centroid kelompok i.

5. Ulangi langkah ke tiga dan ke empat sampai nilai titik pusat kelompok konvergen.

Dengan catatan i diiterasi sebanyak intensitasnya, sedangkan j diiterasi sebanyak kelompoknya [13].

### 2.3.2 Deteksi Tepi Canny

Pendeteksian tepi pada gambar, secara konsepnya adalah mencari titik dimana adanya perubahan nilai intensitas piksel yang besar, dengan jarak perubahannya sangat kecil [3]. Hasil dari pendeteksian tepi tersebut akan membentuk suatu titik pemisah atau batas dalam sebuah gambar, sehingga dalam suatu gambar dapat dipisahkan setiap bagian objeknya.

Deteksi tepi Canny di dalamnya memiliki beberapa tahapan pengerjaan, berikut tahapan deteksi Canny [3].

1. Penghalusan atau *filtering*

Pada tahap ini sebuah gambar yang akan diproses, dihaluskan terlebih dahulu, bertujuan untuk menghilangkan suatu derau sekaligus menguatkan sebuah tepi, dengan diaplikasikan sebuah Gaussian filter, dengan standar deviasi 1.4 ( $\sqrt{2}$ ) dan ukuran konvolusi kernel 5 x 5 berikut konvolusi kernelnya,

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (2.3)$$

konvolusi kernel tersebut akan dilakukan operasi konvolusi ke pada gambar yang akan dideteksi.

2. Menghitung *gradient*

Menghitung *gradient* setiap piksel dari sebuah gambar, tujuan menghitung *gradient* tersebut adalah untuk mencari kekuatan tepi dan arah tepi, untuk menghitung gradiennya diaplikasikan kernel sobel,

$$K_{Gx} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$K_{Gy} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.5)$$

hasil perhitungan menggunakan kernel tersebut dilanjutkan dengan menghitung kekuatan tepinya dengan cara,

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

kemudian dihitung arahnya.

$$\theta = \arctan \frac{G_y}{G_x} \quad (2.7)$$

### 3. *Non-maximum suppression*

*Non-maximum suppression* adalah tahap yang memproses setiap nilai tepian dan arahnya dari hasil tahap ke-2, agar pada hasil akhir hanya ada nilai tepian yang menunjukkan suatu tepi yang sebenarnya, berikut prosedur dalam *Non-maximum suppression*.

1. Putar arah *gradient*  $\theta$  ke arah koordinat terdekat, berdasarkan dari delapan *piksel* yang terhubung dengan *piksel gradient* tersebut.
2. Bandingkan kekuatan tepi dari *piksel* tersebut, dengan *piksel* yang di arah positif dan negatif arah gradiennya.
3. Apabila kekuatan tepi *piksel* tersebut memiliki nilai terbesar, maka simpan nilainya, namun apabila nilai *piksel* tersebut lebih kecil dari yang dibandingkan, maka nilainya akan dihapus.

### 4. *Double thresholding*

*Double thresholding* adalah langkah yang melakukan penkategorian, apakah piksel yang ada termasuk piksel yang memiliki kekuatan tepi yang kuat atau kekuatan tepi yang lemah, dengan cara memberikan dua batas kategori atau biasa disebut *threshold*, berupa batas untuk kategori kekuatan tepi kuat dan batas kekuatan tepi lemah.

### 5. *Edge tracking by hysteresis*

Tahap ini merupakan tahap pemrosesan hasil kategori dari *Double thresholding*, yang dimana kategori yang memiliki kekuatan tepi yang kuat akan dipetakan pada gambar hasil akhir dari pendeteksian tepi, sedangkan untuk tepi yang berkategori lemah, akan ikut dipetakan hanya apabila tepi yang lemah tersebut tersambung dengan tepi kategori kuat, kemudian piksel dari tepi tersebut dibagi ke dalam blok (*binary large object*) yang terhubung secara 8 ketetanggaan piksel, dalam ketetanggaan tersebut setidaknya harus ada 1 piksel yang memiliki tepi berkategori kuat untuk disimpan, apabila dalam 8 ketetanggaan tersebut tidak ada yang memiliki tepian kuat maka nilai tepi akan dihapus.

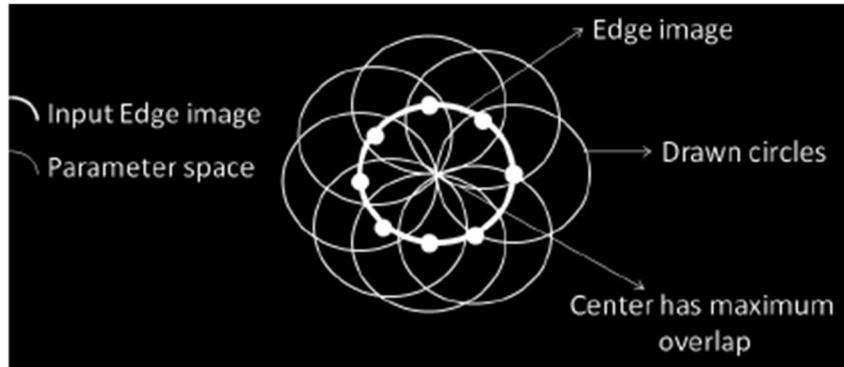
## 2.3.3 *Circular Hough Transform*

*Circular hough transform* adalah algoritma pendeteksian objek pada gambar yang dapat mendeteksi bentuk lingkaran, dengan cara memproyeksikan sebuah lingkaran dengan radius  $r$  pada gambar hasil dari pendeteksian tepi. Lingkaran tersebut diproyeksikan sepanjang tepi pada objek, sampai akhirnya dapat ditemukan *overlap* terbanyak pada hasil proyeksi lingkaran, dengan begitu ditemukanlah titik pusat dan radius dari objek lingkaran yang dicari [14]. Berikut persamaan yang digunakan *circular hough transform*.

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (2.8)$$

Dengan  $r$  adalah radius,  $x_0$  dan  $y_0$  adalah titik pusat lingkaran pada sumbu  $x$  dan  $y$  berikut parameter  $x$  dan  $y$  tersebut.

$$x = x_0 + r \cos(\theta) \quad y = y_0 + r \sin(\theta) \quad (2.9)$$



**Gambar 2.7** Proses circular hough transform [14].

### 2.3.4 Rubber Sheet Model

Normalisasi iris adalah pengkonversian iris dari dimensi kartesian  $(x, y)$  menjadi gambar dalam dimensi polar  $(r, \theta)$ , dengan interval  $r [0,1]$  dan sudut  $\theta (0, 2\pi)$  konversi dimensi kartesian menjadi polar, tidak dipengaruhi oleh ukuran gambar dan dilatasi dari pupil. Sehingga proses normalisasi merupakan tahapan wajib sebelum dilakukanya klasifikasi untuk memperoleh ukuran gambar yang sama. Berikut persamaan normalisasi iris daugman (2.10) [6].

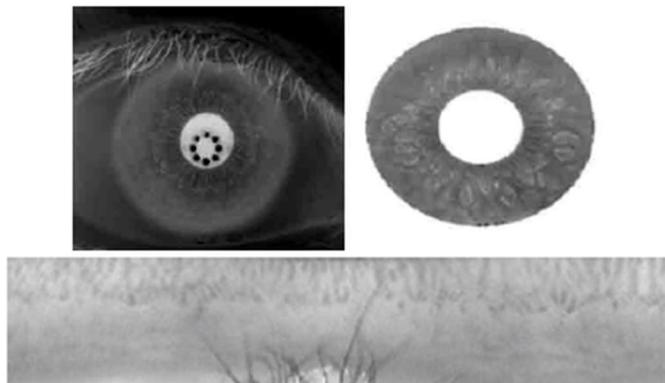
$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \quad (2.10)$$

Dengan  $I(x, y)$  adalah iris di dimensi kartesian  $(x, y)$ , sedangkan  $x(r, \theta)$  dan  $y(r, \theta)$  adalah kombinasi linier dari lingkaran iris dan lingkaran pupil.

$$x(r, \theta) = (1 - r)x_p(\theta) + rx_s(\theta) \quad (2.11)$$

$$y(r, \theta) = (1 - r)y_p(\theta) + ry_s(\theta) \quad (2.12)$$

Sebuah gambar yang memiliki ukuran yang berbeda-beda tidak akan bisa diproses pada tahapan selanjutnya, banyak faktor yang dapat membuat sebuah gambar iris memiliki ukuran yang berbeda-beda, sekalipun dari orang yang sama [8], berikut contoh normalisasi iris.



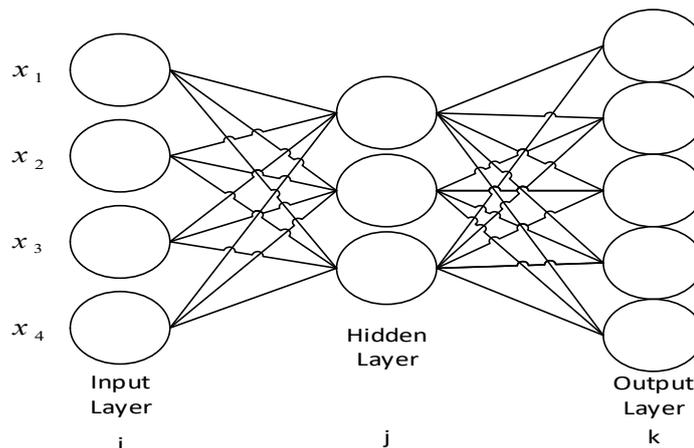
**Gambar 2.8** Normalisasi iris [14].

## 2.4 Klasifikasi

Tahapan klasifikasi adalah tahap, memisahkan suatu *class* dengan *class* yang lain, berdasarkan ciri unik dari *class* tersebut, pada kasus ini adalah iris. Metode yang akan digunakan dalam klasifikasi ini adalah Convolutional Neural Network. Berikut akan dijelaskan metode yang berkaitan dengan CNN dan metode CNN itu sendiri.

### 2.4.1 Multi Layer Perceptron (MLP)

Multi Layer Perceptron adalah sebuah sistem yang meniru cara bekerja jaringan neuron manusia, yang dimana sistem tersebut dapat menerima sebuah *input*, kemudian memproses dan mempelajari *input* dan merespon melalui sebuah *output*, sistem tersebut biasa disebut jaringan syaraf tiruan [14]. Di dalam sebuah jaringan syaraf tiruan secara umumnya ada tiga komponen didalamnya yaitu *input layer hidden layer* dan *output layer*. Didalam setiap *layer* tersebut terdapat komponen yang lebih kecil, yaitu neuron, neuron pada setiap *layer* bisa bervariasi banyaknya, fungsi utama neuron adalah untuk menerima suatu *input* dan memberikan *output* berupa respon. Berikut gambaran sederhana dari sistem Multi Layer Perceptron.



**Gambar 2.9** Multi Layer Perceptron [1].

Cara kerja MLP tersebut sebagai berikut. Pertama sebuah data akan melalui *input layer* dengan jumlah neuron *input* disesuaikan dengan jumlah data, kemudian setiap data dari *input* neuron diteruskan ke *hidden layer*, sebelum nilai-nilai dari *input* tersebut direspon oleh neuron di *hidden layer*, dilakukan operasi dengan sebuah bobot terlebih dahulu, kemudian setelah itu dimasukkan ke sebuah fungsi aktivasi dan akan menjadi nilai *output* untuk neuron pada *hidden layer*, proses tersebut sama untuk setiap neuron selanjutnya, dengan *input* untuk neuron selanjutnya adalah *output* hasil dari neuron *hidden layer*, proses tersebut dilakukan juga pada *output layer*, namun biasanya pada *output layer* memiliki neuron yang lebih sedikit untuk menghasilkan kesimpulan keseluruhan *output*. Berikut persamaan umum untuk pembobotan pada neuron,

$$y_j = out_j = f\left(\sum_{i=1}^n w_{ji}x_i + \theta_j\right) \quad (2.13)$$

yang dimana  $y_j$  adalah nilai hasil untuk neuron ke  $j$  dan  $w_{ji}$  adalah bobot untuk neuron *input* ke  $i$ , menuju neuron *output* ke  $j$ , sedangkan  $x_i$  adalah data *input* ke  $i$  dan  $\theta_j$  adalah bias untuk neuron ke  $j$ .

Pada persamaan (2.13) terdapat  $f(\sum_{i=1}^n w_{ji}x_i + \theta_j)$  atau  $f(Net_j)$ , dimana fungsi tersebut merupakan sebuah fungsi aktivasi, yang bertujuan untuk memberikan interval atau batasan nilai untuk *output* dari neuron secara tidak linier, berikut contoh fungsi aktivasi sigmoid,

$$y_j = out_j = \frac{1}{1 + e^{-x}} \quad (2.14)$$

di dalam algoritma MLP memiliki kemampuan untuk memperbaiki kesalahan pada jaringan, untuk memperbaiki kesalahan tersebut, di dalam MLP digunakan algoritma *backpropagation* yang di dalamnya terdapat *gradien descent*, dengan parameter yang diminimumkan adalah *error*, persamaan (2.15) merupakan persamaan perhitungan *error* untuk masing-masing *node* ke  $j$  sedangkan persamaan (2.16) adalah *error* SSE.

$$e_j = (t_j - o_j) \quad (2.15)$$

$$E = \frac{1}{2} \sum_j (t_j - o_j)^2 \quad (2.16)$$

Dengan  $t_j$  merupakan target *output* yang ingin dicapai dikurangi  $o_j$ , yang merupakan *output* aktualnya, mengacu dari persamaan (2.16) yang merupakan persamaan *error*, dapat dicari persamaan bobot yang dapat menurunkan *error*, pada persamaan (2.17),

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}, w'_{ji} = w_{ji} + \Delta w_{ji} \quad (2.17)$$

yang dimana  $\Delta w_{ji}$  adalah vektor bobot baru yang menghubungkan *layer index*  $i$  dan  $j$ , dengan  $\eta$  adalah merupakan konstanta *learning rate* dan  $w'_{ji}$  adalah bobot baru untuk *layer*  $i$  dan  $j$ , dari persamaan (2.17) lebih spesifiknya pada persamaan (2.18),

$$w'_{kj} = w_{kj} + \eta \delta_k o_j \quad (2.18)$$

$$w'_{ji} = w_{ji} + \eta \delta_j o_i \quad (2.19)$$

dengan  $w'_{kj}$  merupakan bobot baru pada *layer*  $k$  sedangkan  $w'_{ji}$  bobot baru untuk *layer*  $j$  dan  $\delta$  merupakan gradien lokal dari fungsi *error* [1].

#### 2.4.2 Convolutional

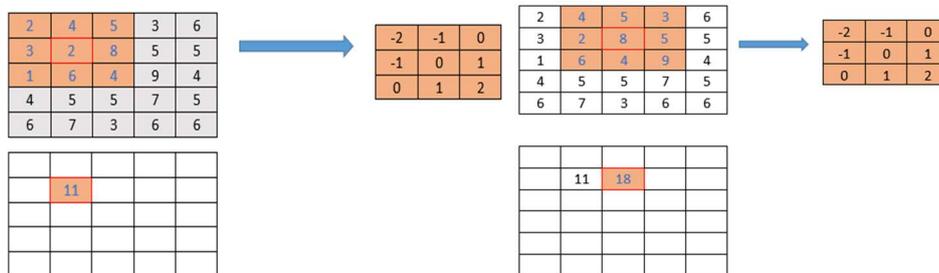
Konvolusi dalam pengolahan citra, adalah proses operasi sebuah gambar dengan sebuah matriks konvolusi sehingga didapat sebuah gambar baru dari hasil operasi tersebut. Matriks tersebut dapat berfungsi sebagai filter atau kernel, berikut fungsi diskrit untuk operasi konvolusi [12],

$$f(x, y)g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x - a, y - b) \quad (2.20)$$

dengan  $g(x)$  merupakan filter konvolusi, berikut notasi lain,

$$h(x, y) = f(x, y) \otimes g(x, y) \quad (2.21)$$

$h(x, y)$  merupakan hasil konvolusi matriks gambar dengan matriks konvolusinya dan simbol  $\otimes$  adalah simbol operasi konvolusi, berikut ilustrasi dari operasi konvolusi.



**Gambar 2.10** Ilustrasi operasi konvolusi

Sebuah matriks berukuran 5 x 5 yang dikonvolusi oleh matriks konvolusi berukuran 3 x 3, pada operasinya matriks convolusi *difilp* secara horizontal dan vertikal terlebih dahulu, kemudian dikali dan ditambahkan, pada contoh gambar operasinya seperti di bawah ini.

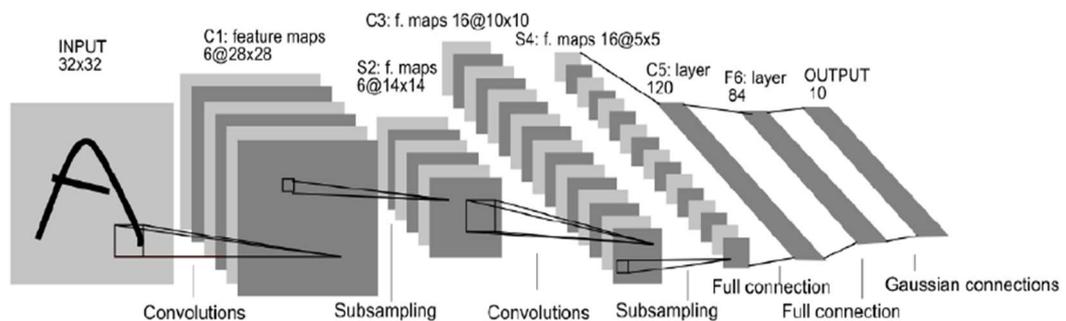
$$(2 \times -2) + (4 \times -1) + (5 \times 0) + (3 \times -1) + (2 \times 0) + (8 \times 1) + (1 \times 0) + (6 \times 1) + (4 \times 2) = 11$$

Hasil tersebut di simpan pada matriks baru, selanjutnya matriks konvolusi akan digeser *piksel* per *piksel* untuk menghitung nilai *piksel* yang lain, sampai semua nilai pada gambar tersebut lengkap, sehingga terbentuk matriks baru yang telah dikonvolusi.

### 2.4.3 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah algoritma yang berdasar pada algoritma Artificial Neural network (ANN) yang dasar cara kerjanya terinspirasi dari cara kerja otak manusia, yang terdiri dari kumpulan neuron [12]. CNN banyak digunakan diidentifikasi dan klasifikasi pada sebuah pengolahan citra, konsep utama CNN adalah menggunakan sebuah matriks konvolusi yang digabungkan dengan algoritma ANN [10].

Sebuah aritektur sederhana dari CNN secara umumnya seperti gambar di bawah ini.



**Gambar 2.11** Gambaran arsitektur CNN [11].

Pada gambar tersebut CNN memiliki setidaknya enam *layer* utama yaitu

1. *Input layer*

2. *Convolutional layer*
3. *Feature layer*
4. *Max pool layer (sub sampling)*
5. *Fully connected layer*
6. *Output layer*

*Input layer*, merupakan *layer* yang memberikan *input* untuk jaringan CNN, *input* tersebut berupa sebuah gambar, misalkan gambar yang digunakan adalah gambar *grayscale* berukuran  $32 \times 32$  maka *input layer* nya juga akan berukuran  $32 \times 32 \times 1$ , angka satu tersebut menunjukkan jumlah *channel* pada *input*, sedangkan apabila gambarnya berupa RGB maka akan berukuran  $32 \times 32 \times 3$ , yang berarti tiga *channel* karena memiliki tiga komponen warna.

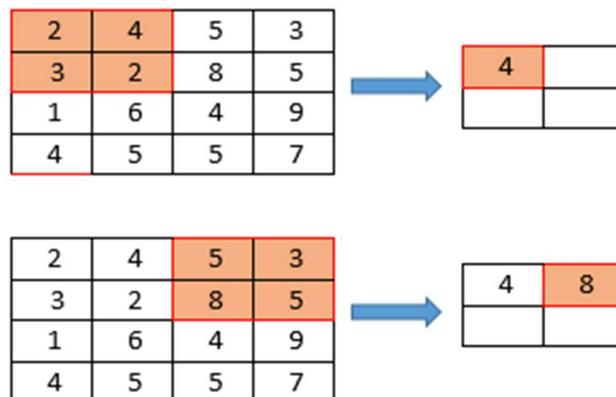
*Convolutional layer* merupakan *layer* konvolusi, seperti konvolusi yang dijelaskan di sub bab 2.4.2 sebelumnya, *layer* konvolusi berfungsi sebagai filter yang akan memanipulasi gambar dari *input layer*, banyaknya *channel* pada *layer* konvolusi harus sama dengan banyaknya *channel* pada *input layer*. Pada *layer* konvolusi jumlah *layer* yang akan digunakan bisa beberapa kali dengan nilai matriks konvolusi yang berbeda, sehingga dapat menghasilkan banyak *output* dari satu *input*, yang berarti dari satu *input* dapat diperoleh beberapa fitur, yang akan menjadi hasil di fitur *layer*. Pada **Gambar 2.4**, hasil konvolusi pada fitur *layer* c1 menghasilkan ukuran fitur *layer* sebesar  $28 \times 28 \times 6$  yang berarti gambar *input* tersebut dikonvolusi dengan nilai yang berbeda sebanyak enam kali.

Fitur *layer* merupakan *layer* yang menyimpan fitur-fitur hasil dari konvolusi, pada fitur *layer* biasanya memiliki fungsi aktivasi, fungsi aktivasi pada fitur *layer* yang biasanya digunakan adalah Rectified Linier Unit (ReLU), ReLU secara matematis memiliki persamaan,

$$f(x) = \max(0, x) \quad (2.22)$$

fungsi utama dari ReLU adalah menghilangkan nilai negatif pada *piksel*, kemudian merubahnya menjadi nilai nol, sedangkan apabila nilai *piksel* lebih dari nol nilai nya tetap.

*Max pool layer*, merupakan *layer* yang mereduksi sekaligus mengambil fitur terkuat dari fitur *layer*, dengan cara mengambil *piksel* terbesar dari fitur *layer* dan dipetakan ke sebuah matriks baru dengan ukuran yang lebih kecil, diilustrasikan oleh **Gambar 2.6**, jangkauan matriks *max pool* nya sebesar  $2 \times 2$ , dengan *stride* (2,2), *stride* adalah langkah pergerakan matriks *max pool*.



**Gambar 2.12** Max Pool Layer

Adapun tujuan utama dari *max pool layer* adalah untuk membuat sebuah fitur tahan terhadap perubahan posisi, sehingga meskipun objek yang nanti diklasifikasi sedikit berbeda dari posisi pada saat *training*, masih dapat diklasifikasi dengan baik.

*Fully Connected layer* pada dasarnya adalah sebuah Multi Layer Perceptron (MLP), MLP tersebut pada konsepnya adalah sama seperti yang telah dijelaskan pada sub bab 2.4.1, hanya saja pada algoritma CNN *input* dari MLP tersebut merupakan *multi channel matrix*. Fungsi dari MLP pada CNN adalah sebagai *classifier* untuk melakukan serangkaian proses yang akhirnya memberikan probabilitas untuk *multiclassification output*, adapun fungsi aktivasi yang digunakan pada MLP adalah fungsi aktivasi ReLU seperti pada persamaan (2.22).

*Output layer* merupakan *layer* terakhir yang akan memberikan nilai berupa vector berisi probabilitas untuk *multiclass classification*, dengan fungsi aktivasi yang dipakai merupakan fungsi aktivasi softmax, seperti pada persamaan (2.23).

$$\sigma(a_j) = \frac{\exp(a_j)}{\sum_k^m \exp(a_k)} \quad (2.23)$$

Dimana,

$\sigma(a_j)$  : nilai probabilitas untuk neuron j,

$a_j$  : nilai hasil *scoring* pada Fully connected neuron ke j.

Cara algoritma CNN mengubah *weight* dan biasnya, sama seperti algoritma MLP dengan menggunakan algoritma *Backpropagation*, hanya saja ada perbedaan sedikit untuk *Backpropagation* CNN. Pertama *Backpropagation* pada CNN ada perbedaan pada turunan fungsi aktifasinya, untuk perhitungan errornya menggunakan perhitungan *error cross entropy* pada persamaan (2.24).

$$D(\sigma, L) = -\sum_i L_i \log(\sigma_i) \quad (2.24)$$

$$E = \frac{1}{N} \sum_i D(\sigma, L) \quad (2.25)$$

Dimana,

$L$  : *desired* label dengan *one hot encoding*,

$\sigma$  : hasil probabilitas softmax,

$D(\sigma, L)$  : distance atau jarak probabilitas softmax terhadap  $L$ ,

$E$  : error untuk algoritma *backpropagation*.

Dari persamaan (2.24) dan (2.25) terlihat bahwa tahap untuk menemukan bobot baru, sama dengan pembaharuan bobot pada MLP persamaan (2.18), hanya yang akan membedakannya fungsi turunan dari fungsi aktifasinya.