



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Time-of-Flight Camera

*Time-of-flight camera* adalah kamera yang mengukur jarak berdasarkan kecepatan cahaya, serta mengukur waktu perjalanan cahaya antara kamera dengan subjek untuk setiap titik pada gambar. Kamera ToF mendapatkan hasil gambar dengan setiap pancaran laser atau cahaya sedangkan sistem LIDAR (*light radar*) menangkap seluruh gambar dengan pancaran laser atau cahaya pada setiap titik yang dipindai (Iddan dan Yahav, 2001).

Sebuah kamera ToF memiliki banyak kelebihan dari teknologi pemindaian 3 dimensi alternatif lainnya, karena dapat mengukur kedalaman objek yang cocok diimplementasikan menjadi pemindai objek yang cepat (Cui dkk., 2010). Kamera ToF juga diterapkan dalam bidang otomotif, digunakan untuk pengamananan pejalan kaki, dan deteksi perubahan posisi yang tidak sesuai (Elkhalili dkk., 2006). Dalam *industry game*, kamera ToF digunakan oleh Microsoft pada perangkat konsol *game* Xbox One yang bernama Kinect. Pada generasi kedua Kinect memungkinkan *natural use interfaces* pada pemain dengan penggunaan *computer vision* dan teknik pengenalan gestur tubuh (Rubin, 2013). Kamera ToF dipilih karena pengambilan data 3 dimensi dari suatu objek membutuhkan kamera khusus yang berbeda dari kamera biasanya dan kamera ToF mudah digunakan dan kamera ToF memiliki harga yang relatif lebih murah (Zhang dan Lu, 2013).

Menurut Zhang dan Lu pada skripsi Daud Julio (2016) sebuah kamera ToF menghasilkan 3 jenis data untuk setiap *pixel* yang dihasilkan yaitu *distance*, *amplitude*, dan *intensity*. *Distance* adalah jarak antara kamera dengan objek. *Amplitude* adalah tingkat

keandalan dari nilai jarak yang didapatkan. Semakin reflektif permukaan dari objek, semakin tinggi juga nilai dari *amplitude*. Bila jarak objek di luar dari jarak pengamatan, maka nilai dari *amplitude* akan mendekati nol. *Intensity* adalah tingkat kecerahan dari suatu objek. Semakin banyak cahaya yang masuk ke kamera maka nilai dari *intensity* akan semakin tinggi.

## 2.2 Kinect Xbox One

Kinect Xbox One adalah perangkat interaksi yang diproduksi oleh Microsoft untuk menambakan interaksi kepada pemain seperti gestur tubuh dan berbicara (Tashev, 2013). Kinect Xbox One merupakan perkembangan lanjutan dari Kinect Xbox 360 dengan menambahkan teknologi *time-of-flight* (ToF) *camera* untuk mendapatkan data kedalaman yang lebih akurat. Cahaya yang digunakan untuk melakukan perhitungan jarak adalah cahaya inframerah sehingga dapat dilakukan perhitungan dengan tingkat cahaya yang rendah atau tanpa cahaya (Meisner, 2013).

## 2.3 Neural Network

*Neural Networks* atau jaringan syaraf tiruan adalah metode pemrosesan informasi yang terinspirasi dari cara kerja sistem syaraf pada makhluk hidup (Aleksander dan Morton, 1990). *Neural Networks* terdiri dari atas banyak elemen yang saling terkoneksi satu dengan lainnya. Menurut Daud Julio (2016), jaringan syaraf tiruan (JST) digunakan untuk tujuan tertentu seperti mendeteksi pola pikir dan klarifikasi data melalui sebuah proses pelatihan. Proses pelatihan tersebut dilakukan dengan mengubah hubungan antar *neuron* (Jonathan, 2017). Dalam implementasinya ke sistem komputer sebuah metode pemrosesan informasi dapat disebut sebagai *neural network* bila metode tersebut memenuhi kedua karakteristik berikut (Russel, 2012).

1. Memiliki kumpulan *weight* atau beban yang dapat berubah dan diubah dengan menggunakan sebuah algoritma pelatihan.
2. Memiliki kemampuan untuk mengira-ngira fungsi *non-linear* dari *input*.

Rumus dari *neural networks* dapat ditulis sebagai berikut.

$$f(x) = K(\sum w_i g_i(x)) \quad \dots(2.1)$$

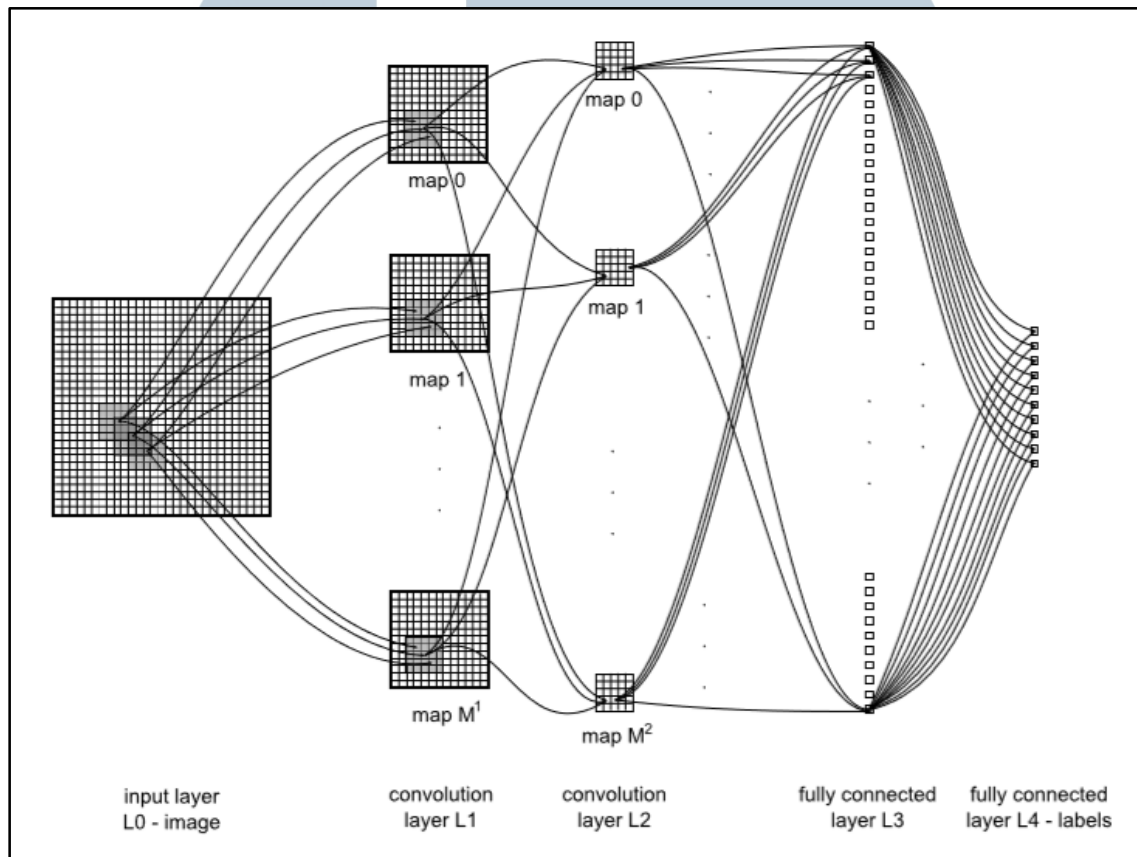
1.  $f(x)$  adalah hasil *output* dari pemrosesan informasi menggunakan *neural networks*.
2.  $x$  adalah *input* atau masukan dari *neural networks* yang akan diproses.
3.  $K$  adalah fungsi yang digunakan untuk menggabungkan berbagai *input* dari suatu *node* menjadi *output* dari *node* tersebut.
4.  $w_i$  adalah beban dari sinaps atau hubungan antar *node*.
5.  $g_i(x)$  adalah hasil dari pemrosesan *node*  $g$  terhadap *input*  $x$ .
6.  $w_i g_i(x)$  adalah perkalian beban terhadap *input* yang terjadi ketika *input* diteruskan melalui sinaps.
7.  $\sum w_i g_i(x)$  adalah gabungan dari setiap hasil perkalian yang diterima oleh *node*.

Hasil dari perhitungan tersebut tidak dapat menghasilkan *output* yang sesuai karena nilai beban yang ditentukan secara acak. Perolehan *output* yang sesuai dapat dilakukan dengan mengubah dan menyesuaikan dengan nilai-nilai beban tersebut dengan melakukan proses pelatihan terhadap *neural networks*.

#### 2.4 Convolutional Neural Network

*Convolutional Neural Networks* (CNN) merupakan bagian dari *deep learning* yang sering digunakan untuk menganalisis citra visual. CNN merupakan jenis syaraf tiruan *multi-layer* (LeCun, 2013). CNN menggunakan *pre-processing* yang relatif sedikit dibandingkan

dengan algoritma sejenis (Oord dkk, 2013). CNN telah diterapkan dengan baik untuk mendeteksi, segmentasi, dan pengenalan objek yang digunakan untuk mendeteksi muka, teks, dan tubuh manusia dalam citra yang natural (LeCun dkk, 2015).



Gambar 2.1 Arsitektur Convolutional Neural Network (Ciresan, 2013)

*Convolutional layer* memiliki parameter besar dan jumlah *pixel*, ukuran *kernel*, *skipping factor*, dan tabel koneksi antar *pixel* (Ciresan, 2013).

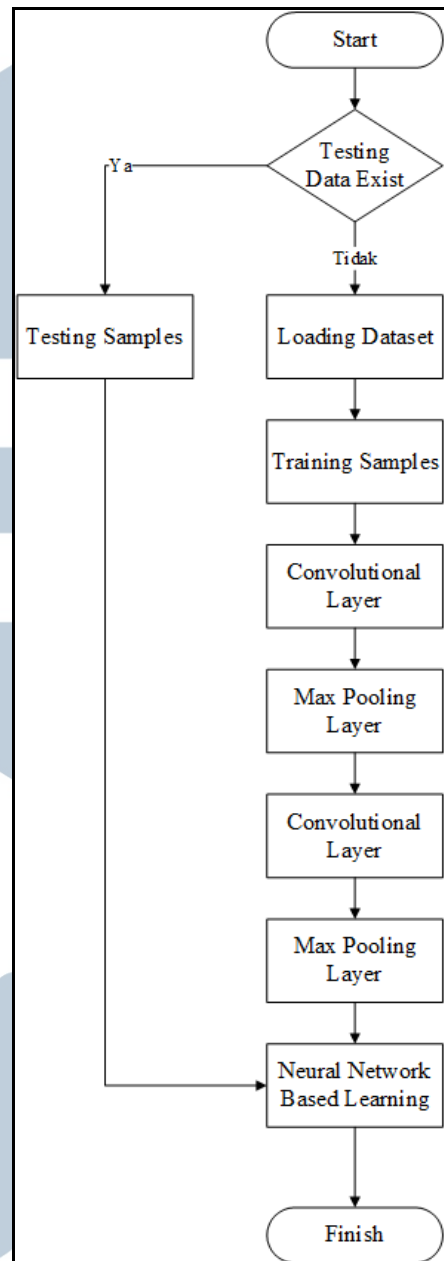
$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^{n+1}} + 1; M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^{n+1}} + 1 \quad \dots(2.2)$$

1.  $M$  setiap *pixel* dari layer yang memiliki ukuran yang sama.
2.  $K$  bagian yang akan dipetakan dari sebuah *input*.
3.  $S$  bagian dari *pixel* yang terpakai yang sudah dipetakan.

*Pooling layer* mengurangi dimensi dari sebuah data dengan mengkombinasikan *output* sebuah *neuron* pada satu *layer* menjadi sebuah *neuron* pada *layer* selanjutnya. *Pooling layer* dapat membantu mempercepat konvergensi dan meningkatkan generalisasi. Hasil dari *pooling layer* diberikan oleh daerah persegi panjang yang tidak tumpang tindih yang memungkinkan posisi dari *invariant* pada daerah yang lebih besar dan menurunkan sampel gambar pada setiap arah (Ciresan, 2013).

*Classification layer* adalah ukuran kernel dari *convolutional filter* dan persegi dari *pooling layer* serta factor yang dipilih sedemikian rupa sehingga hasil dari area terakhir pada *convolutional layer* dikecilkan ukurannya menjadi 1 piksel per area, atau lapisan yang terhubung sepenuhnya menggabungkan hasil dari *convolutional layer* teratas ke dalam vektor satu dimensi (Ciresan, 2013). Berikut adalah Gambar 2.1 yang merupakan *flowchart* dari algoritma CNN.





Gambar 2.2 Flowchart Algoritma CNN

## 2.5 Principal Component Analysis

*Principal Component Analysis* adalah teknik yang menggunakan prinsip matematika untuk mengubah sejumlah variabel yang mungkin berkorelasi menjadi variabel kecil yang disebut komponen utama. PCA digunakan untuk penyederhanaan, pengurangan data,

pemodelan, dan klasifikasi data. Berdasarkan penelitian Jonathan (2017), PCA dapat diimplementasikan untuk mengoptimasi jaringan saraf tiruan dalam pengenalan wajah 3 dimensi.

Data dari wajah yang sudah didapatkan, kemudian dihilangkan *noise* untuk meningkatkan akurasi pada perhitungan PCA menggunakan rumus.

$$\Psi = \frac{1}{N} \sum_{i=1}^N \tau_i \quad \dots(2.3)$$

Dimana:

$\Psi$  = vektor psi = *noise*

N = jumlah vektor = jumlah citra

$\tau_i$  = vektor tau ke-i

# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA