

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tulisan Tangan**

Tulisan tangan adalah hal unik yang dihasilkan manusia, setiap pribadi memiliki keunikan dalam menghasilkan tulisan sesuai dengan kepribadiannya (Joshi et al., 2015). Sebuah kata yang sama akan terlihat berbeda jika ditulis oleh orang yang berbeda. Di sisi lain, perbedaan bahasa juga mempengaruhi bentuk huruf atau bentuk aksara yang digunakan seperti huruf hijaiyyah untuk bahasa Arab, huruf kanji untuk bahasa Jepang, dan di Indonesia diantara aksara yang kita miliki yaitu aksara Jawa dan aksara Sunda. Saat ini, alfabet latin adalah aksara yang paling banyak digunakan pada beberapa bahasa, seperti bahasa Inggris dan bahasa Indonesia.

Berdasarkan bentuknya, terdapat dua jenis huruf latin yaitu huruf latin bersambung dan huruf latin cetak (Pratama, 2019). Huruf latin bersambung ditulis dengan huruf yang memiliki garis penghubung (ditulis miring), ada yang menulis secara miring dan ada pula yang tegak. Saat ini, penggunaan huruf latin cetak lebih banyak digunakan dalam kehidupan sehari-hari. Hal ini karena bentuk huruf cetak penulisannya lebih mudah daripada huruf bersambung. Namun, tak jarang pula yang menggabungkan kedua huruf tersebut dalam satu kata.

#### **2.2 Citra Digital**

Citra adalah representasi dari informasi yang terkandung di dalamnya sehingga mata manusia dapat menganalisis dan menginterpretasi informasi sesuai dengan

tujuan yang diharapkan. Citra digital merupakan suatu teknik pengolahan/manipulasi gambar atau citra yang dipresentasikan dalam bidang dua dimensi. Citra digital merupakan fungsi 2 dimensi  $f(x, y)$ .  $x$  dan  $y$  merupakan koordinat spasial dan  $f$  adalah nilai intensitas warna pada koordinat  $x$  dan  $y$ . Nilai  $f$ ,  $x$  dan  $y$  adalah nilai berhingga. Nilai dalam citra digital bersifat diskret. Nilai  $x$  menyatakan banyaknya pixel per baris dan nilai  $y$  menyatakan banyaknya pixel per kolom. Jumlah pixel keseluruhan adalah  $x * y$ . Nilai  $f(x, y)$  antara 0 dengan 255 yang merupakan nilai yang menyatakan derajat keabuan. Nilai 0 menyatakan pixel sangat gelap (hitam) dan 255 menyatakan pixel sangat terang (putih) (Wulandari, 2019).

### **2.3 Pengenalan Pola**

Pengenalan pola merupakan proses klasifikasi objek atau pola menjadi beberapa kelas atau kategori dengan tujuan untuk mengambil keputusan (Ervina, 2020). Terdapat tiga langkah utama dalam pengenalan pola, yaitu *preprocessing*, ekstraksi ciri, dan klasifikasi. *Preprocessing* merupakan proses untuk memperbaiki citra sehingga citra dapat digunakan untuk proses selanjutnya. Selanjutnya citra akan diekstraksi untuk memunculkan karakteristik pembeda dari setiap citra. Terakhir citra akan diklasifikasikan atau dikelompokkan menjadi kelas yang sesuai untuk setiap citra.

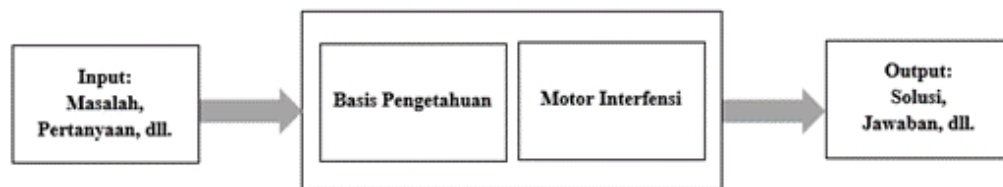
### **2.4 Artificial Intelligent (AI)**

*Artificial Intelligent (AI)* yang biasa diartikan kecerdasan buatan merupakan bagian dari ilmu komputer yang mempelajari bagaimana menjadikan mesin atau komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia,

atau bahkan bisa lebih baik lagi. Menurut John McCharty (1956) yang dikutip dari sebuah artikel, AI melibatkan mesin yang dapat melakukan tugas-tugas yang merupakan ciri khas kecerdasan manusia, seperti perencanaan, pemahaman bahasa, pembelajaran dan pemecahan masalah. Terdapat dua hal utama dalam pembuatan aplikasi kecerdasan buatan (Rakhmat Kurniawan. R, S.T., 2020), yaitu:

1. *Knowledge Base* (Basis Pengetahuan), bagian ini berisikan fakta, teori, serta pemikiran yang saling berhubungan satu dan yang lainnya.
2. *Inference Engine* (Motor Inferensi) yaitu kemampuan untuk menarik kesimpulan berdasarkan pengalaman.

Bagian utama *artificial intelligent* digambarkan dalam Gambar 2. 1



**Gambar 2. 1** Bagian Utama Artificial Intelligence (AI) (Rakhmat Kurniawan. R, S.T., 2020)

AI merupakan salah satu disiplin ilmu yang luas, beberapa lingkup utamanya antara lain Sistem Pakar (*Expert System*), Pengolahan Bahasa Alami (*Natural Language Processing*), Pengenalan Ucapan (*Speech Recognition*), *Computer Vision*, dan lainnya. Sistem pakar adalah sebuah proses menirukan seorang pakar dengan mentransfer kepakaran ke dalam komputer sehingga dapat digunakan oleh yang bukan pakar sekalipun. Melalui pengolahan bahasa alami, pengguna dapat melakukan komunikasi dengan komputer menggunakan bahasa sehari-hari. Pengenalan ucapan membuat manusia dapat melakukan komunikasi dengan

komputer menggunakan suara. Sedangkan *computer vision* berfungsi untuk menginterpretasikan objek atau gambar yang tampak melalui komputer.

Pada domain tertentu terdapat sistem yang memiliki keahlian manusia yang disebut *soft computing*. Hal tersebut merupakan inovasi dari sistem cerdas yang mampu beradaptasi dan bekerja lebih baik ketika terjadi perubahan lingkungan. *Soft computing* mengeksplorasi adanya toleransi terhadap ketidakpastian, ketidaktepatan, dan kebenaran parsial sehingga dapat diselesaikan dan dikendalikan dengan mudah agar sesuai dengan realita. Metodologi yang sering digunakan dalam *soft computing* salah satunya adalah Jaringan Syaraf Tiruan (*Artificial Neural Network*). Metodologi lain yang digunakan adalah *Fuzzy System* untuk mengakomodasi ketepatan, *Probabilistic Reasoning* untuk mengakomodasi ketidakpastian, dan *Evolutionary Computing* yang biasa digunakan untuk optimasi.

Secara garis besar AI merupakan pembelajaran tentang bagaimana membuat komputer melakukan hal-hal yang dapat dilakukan lebih baik oleh manusia. Komputer yang semula hanya difungsikan sebagai alat hitung saja diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

#### **2.4.1 *Machine Learning* (ML)**

Istilah *machine learning* pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. *Machine Learning* berfokus pada pembuatan mesin yang dapat belajar tanpa diprogram secara eksplisit.

*Machine learning* memiliki tiga teknik pembelajaran, yaitu *Supervised Learning*, *Unsupervised learning*, dan *Reinforcement Learning*. Dua metode pembelajaran mesin yang paling banyak diadopsi adalah *Supervised learning* (pembelajaran terawasi) dan *Unsupervised learning* (pembelajaran tanpa pengawasan).

*Supervised learning* merupakan pembelajaran yang terdiri dari data masukan dan data target keluaran untuk dipakai melatih jaringan sehingga dapat menghasilkan bobot yang diinginkan. *Supervised learning* umumnya digunakan dalam aplikasi yang menyimpan data historis untuk memprediksi kemungkinan peristiwa di masa depan. Contohnya, kartu kredit dapat mengantisipasi kemungkinan penipuan atau pelanggan asuransi mana yang akan mengajukan klaim.

*Unsupervised Learning* digunakan terhadap data yang tidak memiliki label historis. Sistem tidak diberi tahu “jawaban yang benar”, dalam kata lain algoritma harus mencari tahu apa yang akan ditampilkan. Tujuan pembelajaran ini adalah untuk mengeksplorasi data dan mengelompokkan unit-unit yang memiliki kemiripan ke dalam area tertentu. Salah satu contoh *unsupervised learning* yaitu pengelompokan individu berdasarkan demografi sosial.

#### **2.4.2 *Deep Learning***

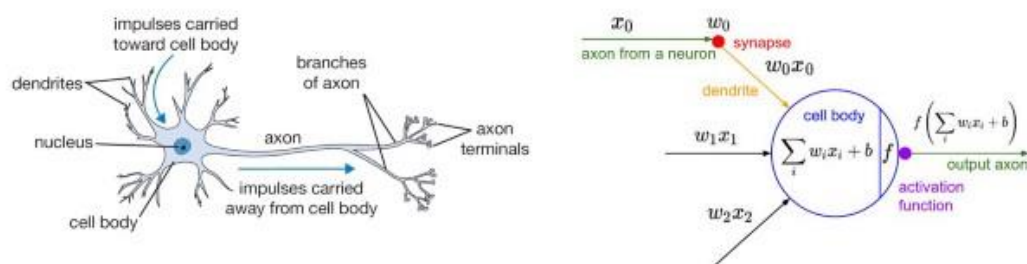
*Deep Learning* merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning* karena menambahkan lebih banyak lapisan, sehingga

model pembelajaran yang dibuat bisa mewakili data citra berlabel dengan lebih baik.

Aplikasi konsep *deep learning* dapat ditanggihkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer Vision*, *Speech recognition*, dan *Natural Language Processing*. Metode *Convolutional Neural Network* merupakan metode yang umum digunakan untuk proses *deep learning* dan mampu memberikan hasil yang cukup baik.

## 2.5 Neural Network

*Neural network* atau jaringan syaraf tiruan (Sena, 2017a) adalah model yang terinspirasi oleh bagaimana *neuron* dalam otak manusia bekerja. Tiap neuron pada otak manusia saling berhubungan dan informasi mengalir dari setiap neuron tersebut, seperti yang ditampilkan dalam Gambar 2. 2



**Gambar 2. 2** Ilustrasi neuron dan model matematisnya (Sena, 2017a)

Berdasarkan Gambar 2. 2, ada tiga input dalam *neuron* ( $x_0$ ,  $x_1$ ,  $x_2$ ), setiap input tersebut dikalikan terlebih dahulu dengan *variable* yang disebut sebagai ‘*weight*’ ( $w_0$ ,  $w_1$ ,  $w_2$ ), setelah itu ketiganya dijumlahkan. Setiap koneksi *neuron* memiliki

'*weight*' masing-masing dan nilainya akan berubah saat proses *learning* sampai model yang dihasilkan ANN mendekati target *output* yang diinginkan. Setelah itu bisa ditambahkan bias  $b$  ke dalam hasil penjumlahan di atas. Nilai bias ini tidak datang dari input layer. Bias seperti *intercept* dalam persamaan linear, ditambahkan untuk mengatur agar hasil perhitungan lebih akurat. Setelah semua penjumlahan di atas dilakukan, *neuron* akan dimasukkan ke sebuah fungsi yang disebut *Activation Function*. *Activation Function* mengatur apakah neuron tersebut harus aktif atau tidak (Yunus, 2020).

Lapisan-lapisan penyusun ANN dibagi menjadi tiga (Ervina, 2020), yaitu:

1. *Input Layer*

Node-node di dalam lapisan input disebut unit-unit input. Unit-unit input menerima input dari dunia luar. Input yang dimasukkan merupakan penggambaran suatu masalah.

2. *Hidden Layer*

Node-node di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Output dari lapisan ini tidak secara langsung dapat diamati.

3. *Output Layer*

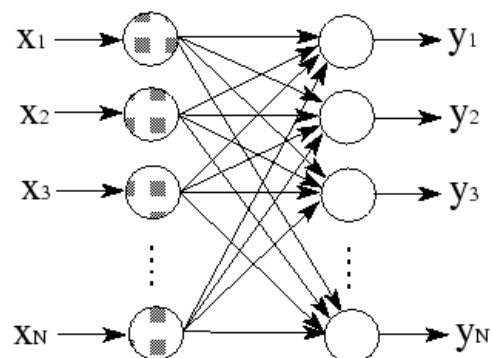
Node-node pada lapisan *output* disebut unit-unit *output*. Keluaran atau *output* dari lapisan ini merupakan output jaringan syaraf tiruan terhadap suatu permasalahan.

### 2.5.1 Arsitektur *Neural Network*

*Neural Network* memiliki beberapa arsitektur jaringan yaitu:

#### 1. *Single Layer Network*

*Neuron* pada *input layer* tidak memiliki *activation function*, sedangkan *neuron* pada *hidden layer* dan *output layer* memiliki *activation function* yang kadang berbeda tergantung daripada data atau masalah yang kita miliki. Gambar 2. 3 merupakan gambaran dari *Single Layer Network*

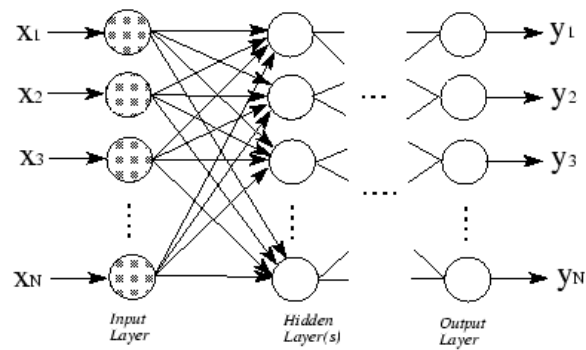


**Gambar 2. 3** Arsitektur *Single Layer Network*

#### 2. *Multi Layer Perceptron (MLP)*

*Multi Layer Perceptron* memiliki 3 lapisan, yaitu *input layer*, *hidden layer*, dan *output layer*. Setiap *neuron* pada MLP saling berhubungan yang ditandai dengan tanda panah pada gambar di atas. Tiap koneksi memiliki *weight* dengan nilai dari tiap *weight* yang berbeda. *Hidden layer* dan *output layer* memiliki tambahan “*input*” yang biasa disebut dengan bias. *Neuron* pada *input layer* tidak memiliki *activation function*, sedangkan *neuron* pada *hidden layer* dan *output layer* memiliki *activation function* yang kadang berbeda tergantung daripada data atau masalah yang kita miliki (Sena, 2017a). Arsitektur MLP dapat dilihat pada Gambar 2 .4





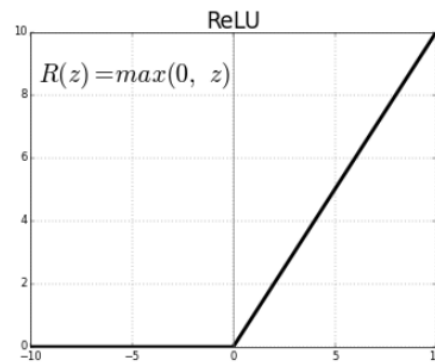
**Gambar 2. 4** Arsitektur Multi Layer Perceptron

### 2.5.2 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas internal (*summation function*) yang mungkin berbentuk *linear* ataupun *non-linear*. Fungsi ini bertujuan untuk menentukan apakah neuron diaktifkan atau tidak (Nurhikmat, 2018). Fungsi aktivasi yang akan digunakan pada penelitian ini yaitu fungsi aktivasi ReLU dan Tanh.

#### 1. Fungsi Aktivasi ReLU

Fungsi ReLU (*Rectified Linear Unit*) menjadi salah satu fungsi yang populer saat ini. Fungsi ini bekerja dengan dua cara yaitu jika masukan dari *neuron-neuron* berupa bilangan negatif maka fungsi ini akan menerjemahkan nilai tersebut kedalam nilai 0, dan jika masukan bernilai positif maka *output* dari *neuron* adalah nilai aktivasi itu sendiri. Grafik fungsi ReLU digambarkan pada Gambar 2. 5

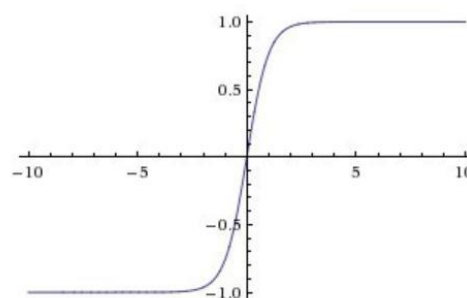


**Gambar 2. 5** Fungsi aktivasi ReLU

Fungsi ReLU memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan dengan fungsi sigmoid dan tanh.

## 2. Fungsi Aktivasi Tanh

Fungsi aktivasi Tanh merupakan fungsi nonlinear. Masukan untuk fungsi aktivasi ini berupa bilangan real dan keluaran dari fungsi tersebut memiliki *range* antara -1 sampai 1. Grafik fungsi aktivasi tanh dapat dilihat pada Gambar 2.6



**Gambar 2. 6** Fungsi aktivasi Tanh

Fungsi Tanh memiliki kekurangan yaitu dapat mematikan gradient, akan tetapi fungsi ini juga memiliki kelebihan yaitu output yang dimiliki fungsi Tanh merupakan *zero-centered*.

### 2.5.3 Pelatihan Neural Network

Pada umumnya terdapat dua tahap pembelajaran, yaitu *training* dan *evaluation*. Pada tahap *training* setiap *weight* dan bias pada tiap *neuron* akan diperbaharui terus menerus hingga *output* yang dihasilkan sesuai dengan harapan. Pada tiap iterasi akan dilakukan proses *evaluation* yang biasanya digunakan untuk menentukan kapan harus menghentikan proses *training* (Sena, 2017a). Proses *training* terdiri dari dua tahap, yaitu:

1. ***Forward Pass***.

*Forward pass* atau biasa juga disebut *forward propagation* adalah proses dimana kita membawa data pada input melewati tiap neuron pada *hidden layer* sampai kepada *output layer* yang nanti akan dihitung errornya.

2. ***Backward Pass***

Error yang kita dapat pada *forward pass* akan digunakan untuk memperbaharui setiap *weight* dan bias dengan *learning rate* tertentu.

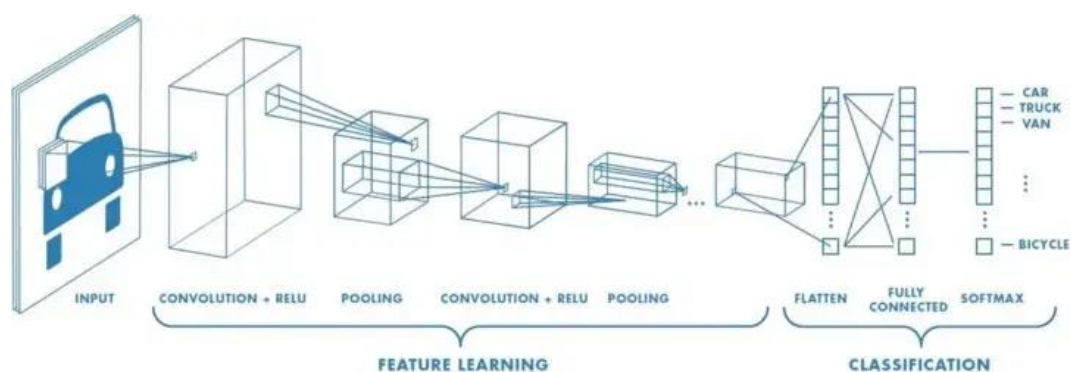
Kedua proses di atas akan dilakukan berulang-ulang sampai didapatkan nilai *weight* dan bias yang dapat memberikan nilai error sekecil mungkin pada *output layer*.

## 2.6 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*, tepatnya untuk mendeteksi dan mengenali objek pada sebuah *image*. Cara kerja CNN terinspirasi dari cara manusia dalam menghasilkan persepsi visual (Lina, 2019).

CNN adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning* khususnya CNN, berhasil mengungguli metode *Machine Learning* lainnya seperti SVM klasifikasi pada objek citra (Eka Putra, 2016).

*Convolutional Neural Network* terdiri dari *weight*, *bias*, dan *activation function* seperti pada *Neural Network*. Sedangkan untuk arsitekturnya CNN memiliki 2 bagian besar yaitu *Feature Extraction Layer* dan *Fully Connected Layer*, arsitektur tersebut dapat dilihat pada Gambar 2. 7



**Gambar 2. 7** Arsitektur CNN (Prihatiningsih et al., 2019)

### 2.6.1 Convolutional Layers

*Convolutional Layer* atau lapisan konvolusi terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*) (Sena, 2017c). Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada lapisan tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN (Eka Putra, 2016).

Operasi konvolusi bekerja dengan cara mengalikan matriks citra masukan dan citra keluaran, kemudian outputnya akan dihitung dengan *dot product*. Penentuan volume *output* juga dapat ditentukan dari masing-masing lapisan dengan *hyperparameters*. Persamaan di bawah ini digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali output.

$$(W - F + 2P) / (S + 1)$$

Keterangan:

W = Ukuran Citra

F = Ukuran Filter

P = Nilai *Padding*

S = Nilai *Stride*

Berdasarkan *persamaan* di atas, dapat dihitung ukuran spasial dari volume output dimana *hyperparameter* yang dipakai adalah ukuran volume (W), filter (F), *stride* yang diterapkan (S) dan jumlah *padding* yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser filter melalui input citra dan *Padding* adalah nilai untuk mendapatkan angka nol di sekitar pinggiran citra.

Sebagai contoh, Gambar 2. 8 memberikan gambaran dari proses konvolusi dengan matriks input berukuran 7x7 dan filter 5x5, dan pergeseran atau stride 1. Sehingga proses tersebut akan menghasilkan matriks berukuran 3x3.

13	4	0	0	0	0	0
6	0	0	0	0	0	0
16	1	0	0	0	0	0
15	1	0	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

 $*$ 

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

 $=$ 

250	14	0
204	10	0
102	1	0

**Gambar 2. 8** Contoh proses konvolusi, input 7x7 dan filter = 5

### 2.6.2 Pooling Layer

*Pooling* adalah proses mereduksi ukuran sebuah citra. Tujuannya untuk mengurangi dimensi dari *feature map* (*downsampling*), sehingga proses komputasi menjadi lebih cepat. Pada prinsipnya, *pooling layer* terdiri dari sebuah filter dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*.

Metode *pooling* yang umum digunakan adalah *Average pooling* dan *MaxPooling*. *Average pooling* mengambil nilai rata-rata dari setiap grid untuk menyusun matriks baru. Sedangkan *MaxPooling* mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi (Eka Putra, 2016). Kedua proses tersebut menggunakan filter 2x2 yang diaplikasikan dengan pergeseran sebanyak dua langkah dan beroperasi pada setiap irisan dari matriks inputnya. Kedua metode tersebut diilustrasikan melalui Gambar 2. 9 dan 2. 10

250	14	0
204	10	0
102	1	0

119.5	6
79.25	2.75

**Gambar 2. 9** Contoh Pooling menggunakan Average Pooling

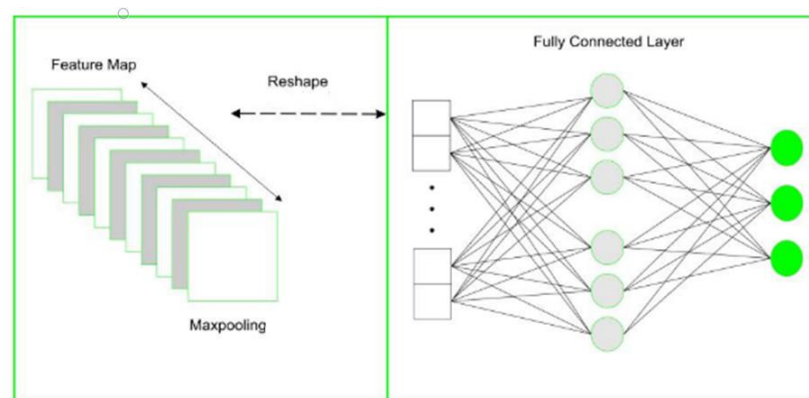
250	14	0
204	10	0
102	1	0

250	14
204	10

**Gambar 2. 10** Contoh Pooling menggunakan Max Pooling

### 2.6.3 Fully Connected Layer

*Feature map* yang dihasilkan dari *feature extraction layer* masih berbentuk *multidimensional array*, sehingga harus dilakukan “*flatten*” atau *reshape feature map* menjadi sebuah *vector* agar bisa digunakan sebagai *input* dari *fully-connected layer*, ilustrasinya tercantum pada Gambar 2. 11



**Gambar 2. 11** Contoh Fully Connected Layer (Wulandari, 2019)

### 2.6.4 Cross Entropy-Loss Function

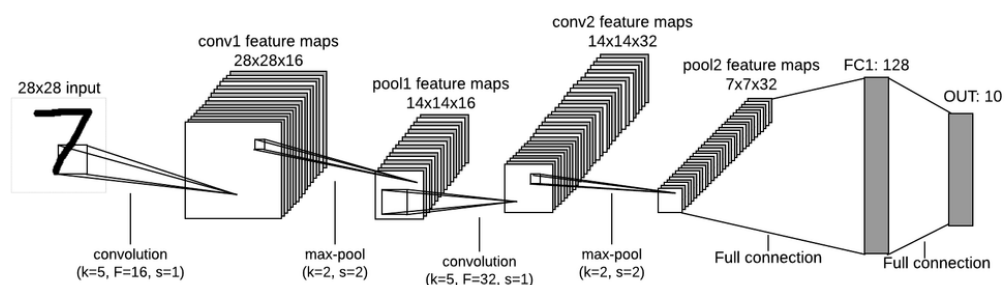
*Loss Function* merupakan fungsi yang menggambarkan kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh model. *Loss Function* yang baik adalah fungsi yang menghasilkan error paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebenaran yang asli untuk meminimalkan perbedaan tersebut. Fungsi yang cocok untuk model dengan kelas yang banyak salah satunya adalah *Cross Entropy*. Algoritma ini bekerja dengan

meminimalkan kemungkinan log negatif dari dataset, yang merupakan ukuran langsung dari performa prediksi model.

### 2.6.5 LeNet5

LeNet5 adalah suatu jaringan berlapis banyak berbasis CNN yang diperkenalkan oleh Yann LeCun. LeNet5 merupakan pengembangan dari LeNet yang sudah ada, keunggulannya yaitu memiliki jumlah parameter bebas atau jumlah lapisan yang lebih banyak dari LeNet sebelumnya. Pertambahan jumlah parameter ini disebabkan oleh makin majunya kecepatan atau teknologi dari komputer sehingga mampu untuk melakukan perhitungan matematis secara lebih cepat.

Gambar 2. 12 menunjukkan contoh arsitekur LeNet5 yang biasa digunakan,



**Gambar 2. 12** Arsitekur LeNet5 <https://www.easy-tensorflow.com/tutorials/convolutional-neural-nets-cnns>

## 2.7 Algoritma Optimasi

Optimasi adalah proses melatih model secara iteratif untuk menghasilkan evaluasi fungsi maksimum dan minimum (Czum, 2020). Tujuan optimasi tersebut adalah agar hasil yang didapatkan di setiap *epoch* mencapai hasil yang maksimal dengan tingkat kesalahan yang minimal. Ada berbagai cara yang dapat digunakan untuk mengoptimalkan model pembelajaran, diantaranya *Stochastic Gradient Descent* dan *Adam*.



### 2.7.1 *Stochastic Gradient Descent (SGD)*

*Gradient Descent* adalah salah satu algoritma paling populer dalam melakukan optimasi pada *Artificial Neural Network* (ANN). Algoritma ini digunakan untuk mengupdate sebuah parameter dalam hal ini adalah bobot (*weight*) dan bias. Algoritma ini cukup sederhana untuk dipahami. *Gradient Descent* bekerja dengan cara meminimalkan fungsi  $J(\theta)$  yang memiliki parameter  $\theta$  dengan memperbarui parameter ke suatu arah menurun (Czum, 2020). Tujuan pengoptimalan dari algoritma ini untuk menemukan parameter yang dapat meminimalkan *loss function*.

*Gradient Descent* memiliki *Learning Rate* ( $\eta$ ) yang digunakan untuk menentukan langkah-langkah yang kita ambil untuk mencapai titik minimum. Hal ini bisa digambarkan dimana suatu objek akan menuruni sebuah bukit dengan langkah tersebut hingga mencapai pada lembah (titik minimum). *Stochastic Gradient Descent* (SGD) adalah metode *gradient descent* yang melakukan update parameter untuk setiap data pelatihan  $x^{(i)}$  serta label  $y^{(i)}$ . Persamaannya adalah sebagai berikut,

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)})$$

SGD sering melakukan *update* dengan varians tinggi yang menyebabkan fungsi objektif meningkat secara tidak beraturan. Disisi lain, hal ini dapat membuat *loss function* akan melompat ke titik minimal yang baru dan berpotensi melompat ke minimum yang tidak pasti. Namun, hal ini dapat dicegah dengan cara mengurangi *learning rate*, dan SGD akan menurunkan nilai *loss function* ke titik minimum secara optimal.

### 2.7.2 Adam Optimizer

Adam adalah algoritma pengoptimalan yang dapat digunakan sebagai ganti dari prosedur *Stochastic Gradient Descent* klasik untuk memperbarui *weight network* secara iteratif berdasarkan data training (Czum, 2020). Adam pertama kali dipresentasikan oleh Diederik Kingma dari OpenAI dan Jimmy Ba dari University of Toronto dalam paper mereka di 2015 ICLR yang berjudul “Adam: A Method for *Stochastic Optimization*“. Metoda Adam menghitung *individual adaptive learning rates* untuk parameter yang berbeda dari perkiraan momen pertama dan kedua dari gradien. Adam banyak digunakan karena mudah untuk diterapkan, efisien secara komputasi, dan cocok untuk data/parameter yang besar.

### 2.8 EMNIST Dataset

*Extended Modified NIST* atau yang diketahui sebagai EMNIST Dataset berisikan bentuk teks tulisan tangan huruf dan angka, untuk bentuk hurufnya sendiri ditulis dalam bentuk kapital dan huruf kecil. Data yang tersedia pada EMNIST *Dataset* dapat diunduh dalam format *.csv*, yang terdiri dari:

- a. EMNIST *By\_Class* berisi karakter huruf dan angka, memiliki 62 kelas dan berisi 814.255 data.
- b. EMNIST *By\_Merge* berisi karakter huruf dan angka, memiliki 47 kelas dan berisi 814.255 data.
- c. EMNIST *Balanced* berisi karakter huruf dan angka, memiliki 47 kelas dan berisi 131.600 data.
- d. EMNIST *Digits* berisi karakter angka, memiliki 10 kelas dan berisi 280.000 data.

- e. EMNIST Letters berisi karakter huruf, memiliki 37 kelas dan berisi 103.600 data.
- f. MNIST berisi karakter angka, memiliki 10 kelas dan berisi 70.000 data.

## 2.9 TensorFlow

TensorFlow merupakan salah satu *framework* untuk *deep learning*, agar pengguna lebih mudah dalam menyelesaikan masalah terkait *deep learning*. TensorFlow dikembangkan oleh Google (Sena, 2017b).

## 2.10 Keras

Keras merupakan *wrapper* dari TensorFlow untuk lebih memudahkan dalam mencari optimizer yang paling cepat dan bagus, *tweaking hyperparameter*, dll (Sena, 2017b).

Keras mulai dikembangkan pada tahun 2015. Saat ini, Keras telah berkembang menjadi salah satu pustaka yang paling populer dan banyak digunakan. Keras dibangun di atas Theano dan TensorFlow. Salah satu fitur utamanya adalah API yang sangat intuitif dan ramah pengguna, hal tersebut memungkinkan pengguna menerapkan jaringan neural hanya dalam beberapa baris kode (Li, 2019).

Selain TensorFlow, Keras juga bisa menggunakan *backend* lain yaitu Theano dan CNTK (Sena, 2017b).

## 2.11 Google Colaboratory

*Google Colaboratory* atau *Google Colab* merupakan perangkat komputasi awan (*cloud computing*) yang dibuat oleh Google dengan tujuan untuk memudahkan kegiatan pembelajaran dan pengolahan data dengan mudah menggunakan antarmuka berbasis Jupyter Notebook atau iPython (*interactive*

*Python*). *Google Colab* menyediakan sebuah platform komputasi gratis berupa komputer virtual untuk setiap penggunanya yang dilengkapi dengan kemampuan pengolahan data yang memadai. Dengan memanfaatkan *Google Colab*, pengguna tidak perlu melakukan instalasi atau pengaturan yang rumit untuk keperluan pengolahan data dengan menggunakan *Python*.

*Google Colab* dilengkapi dengan beberapa modul *Python* bawaan, seperti *numpy* dan *OpenCV* sehingga pengguna dapat memanfaatkan platform ini untuk melakukan pengolahan citra sederhana. Pada bagian ini akan dijelaskan dasar penggunaan *Google Colab* untuk melakukan pengolahan citra digital menggunakan *Python*.

## 2.12 Kajian Pustaka

Penelitian terkait yang dijadikan acuan, bersumber pada penelitian sebelumnya yang berhubungan dengan pengenalan huruf dan angka menggunakan *Convolutional Neural Network*.

**Tabel 2. 1** Jurnal Penelitian Terkait

No	Konten	Deskripsi
1	Judul Paper	Pengenalan Tulisan Tangan Huruf Hijaiyah Menggunakan Metode Convolutional Neural Network
	Penulis	Sunu Ilham Pradika, Budi Nugroho, Eva Yulia Puspaningrum
	Jurnal/Konferensi	Seminar Nasional Informatika Bela Negara (SANTIKA)
	URL	<a href="https://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/35">https://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/35</a>
	Permasalahan	Diperlukan sistem pengenalan tulisan tangan huruf hijaiyah untuk melakukan koreksi otomatis terhadap orang yang belajar menulis huruf hijaiyah, agar dapat digunakan secara mandiri di saat pandemi seperti sekarang.
	Kontribusi	Arsitektur SIP-Net yang dirancang peneliti cukup memuaskan, dengan tingkat akurasi yang kurang lebih sama dengan arsitektur asli dari CNN.
	Metode/Solusi	<ul style="list-style-type: none"> <li><i>Preprocess: resize, initialize color "RGB", rescale</i></li> <li><i>Augmentation: rotation, width and heigh shift, shear, zoom, horizontal flip, filling</i></li> </ul>

		<ul style="list-style-type: none"> <li>• <i>CNN architecture</i></li> <li>• <i>Training</i></li> <li>• <i>Evaluate</i></li> </ul>
	Hasil Utama	Penggunaan CNN untuk pengenalan tulisan tangan diketahui memang lebih unggul dari metode yang lain, akurasi yang didapat dengan objek huruf hijaiyah yaitu sebesar 95,9% untuk data latih dan 99.7% untuk data uji.
	Batasan	Dataset yang digunakan merupakan data sekunder dan tidak terlalu banyak.
2	Judul Paper	Identifikasi Citra Tulisan Tangan Digital Menggunakan Convolutional Neural Network (CNN)
	Penulis	Nahila Khunafa Qudsi, Rosa Andrie Asmara, Arie Rachmad Syulistyo
	Jurnal/Konferensi	Seminar Informatika Aplikatif Polinema
	URL	<a href="http://jurnalti.polinema.ac.id/index.php/SIAP/article/view/344">http://jurnalti.polinema.ac.id/index.php/SIAP/article/view/344</a>
	Permasalahan	Tingkat akurasi pada beberapa penelitian sebelumnya masih rendah sehingga perlu ditingkatkan.
	Kontribusi	Pemrosesan menggunakan metode CNN untuk mendapatkan nilai akurasi terbaik.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Dataset: MNIST</i></li> <li>• <i>CNN: Feature Extraction Layer, Fully Connected Layer</i></li> <li>• <i>Pengujian</i></li> </ul>
	Hasil Utama	CNN menunjukkan akurasi sebesar 98,6 % untuk dataset MNIST dan sebesar 88% untuk data koresponden.
	Batasan	Pengujian hanya dilakukan terhadap dataset, tidak pada model.
3	Judul Paper	Analisa Performa Pengenalan Tulisan Tangan Angka Berdasarkan Jumlah Iterasi Menggunakan Metode Convolutional Neural Network
	Penulis	Siwi Prihatiningsih, Nadhiranisa Shafiy M, Feni Andriani, Nurma Nugraha
	Jurnal/Konferensi	Jurnal Ilmiah Teknologi dan Rekayasa
	URL	<a href="http://dx.doi.org/10.35760/tr.2019.v24i1.1934">http://dx.doi.org/10.35760/tr.2019.v24i1.1934</a>
	Permasalahan	Performa yang dihasilkan dalam penelitian tulisan tangan masih kurang memuaskan.
	Kontribusi	Pembuatan program menggunakan metode CNN yang difokuskan untuk mencapai performa yang baik berdasarkan jumlah iterasi.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Preprocessing: MNIST, One-hot Encoding, reshape, normalization</i></li> <li>• <i>CNN: MNIST preprocessing, convolutional layer, subsampling layer, flatten</i></li> <li>• <i>Analisa Performa</i></li> </ul>
	Hasil Utama	Performa <i>neural network</i> dipengaruhi oleh jumlah iterasi. Akurasi meningkat untuk iterasi 0-20, akan tetapi untuk iterasi dari 100-1000 tidak berpengaruh secara signifikan. Akurasi tertinggi diperoleh pada iterasi ke-1000 sebesar 100,0% pada training, 98,67% pada validasi, dan 98,99% pada testing. Hasil menunjukkan bahwa semakin banyak besar jumlah iterasi yang dilakukan semakin baik performa yang dihasilkan.
	Batasan	Dataset yang digunakan hanya data sekunder.

4	Judul Paper	<i>Convolutional Neural Networks for Handwritten Javanese Character Recognition</i>
	Penulis	Chandra Kusuma Dewa, Amanda Lailatul Fadhillah, dan Afiahayati
	Jurnal/Konferensi	IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
	URL	<a href="https://doi.org/10.22146/ijccs.31144">https://doi.org/10.22146/ijccs.31144</a>
	Permasalahan	Aksara Jawa masih jarang digunakan dalam kehidupan sehari-hari karena masyarakat Jawa (terutama) belum semuanya mengenali aksara Jawa.
	Kontribusi	Perangkat lunak berbasis web yang dapat melakukan pengenalan karakter tulisan tangan aksara Jawa.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Data acquisition for handwritten Javanese Characters</i></li> <li>• <i>Building the CNN Model</i></li> <li>• <i>Model training and testing</i></li> <li>• <i>Developing a web based application for handwritten Javanese Characters Classification</i></li> </ul>
	Hasil Utama	Dari percobaan yang dilakukan dapat disimpulkan bahwa akurasi model CNN lebih baik daripada akurasi model MLP untuk tugas pengenalan karakter bahasa Jawa tulisan tangan dengan teknik validasi silang <i>5-fold</i> . Namun, akurasi model CNN untuk dataset karakter Jawa tulisan tangan tidak bisa mencapai 90% di semua <i>fold</i> .
Batasan	Dataset yang digunakan tidak terlalu banyak dikarenakan belum adanya data sekunder untuk aksara Jawa.	
5	Judul Paper	<i>Integrating Wavelet Coefficients and CNN for Recognizing Handwritten Characters</i>
	Penulis	Madhuri Yadav, Ravindra Kr. Purwar
	Jurnal/Konferensi	Proceedings - 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2018
	URL	<a href="https://ieeexplore.ieee.org/document/8897291/">https://ieeexplore.ieee.org/document/8897291/</a>
	Permasalahan	Tantangan besar dalam pengenalan karakter tulisan tangan <i>offline</i> adalah bentuk karakter yang serupa dan perbedaan gaya menulis. Karakter yang bentuknya serupa membuat tugas klasifikasi menjadi kompleks dan gaya penulisan yang berbeda mengurangi terbacanya karakter.
	Kontribusi	Pengembangan arsitektur CNN yang menghasilkan waktu pelatihan lebih cepat dan akurasi yang lebih tinggi.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Preprocessing</i></li> <li>• <i>Feature Extraction</i></li> <li>• <i>Classification</i></li> <li>• <i>Combine of Wavelet Coefficients and CNN</i></li> </ul>
	Hasil Utama	Akurasi yang dicapai sebesar 96,58%, hal ini menunjukkan bahwa arsitektur CNN yang telah dikembangkan mencapai keadaan stabil jauh lebih cepat daripada CNN sebelumnya dan juga mencapai tingkat pengenalan yang tinggi dibandingkan dengan CNN sebelumnya.
Batasan	Data yang digunakan tidak terlalu banyak	

6	Judul Paper	<i>Cursive Handwriting Recognition System Using Feature Extraction and Artificial Neural Network</i>
	Penulis	Utkarsh Dwivedi, Pranjal Razput, Manish Kumar Sharma
	Jurnal/Konferensi	<i>International Research Journal of Engineering and Technology (IRJET)</i>
	URL	<a href="https://irjet.net/archives/V4/i3/IRJET-V4I3576.pdf">https://irjet.net/archives/V4/i3/IRJET-V4I3576.pdf</a>
	Permasalahan	Waktu pemrosesan yang lama dan akurasi yang belum cukup tinggi pada sistem pengenalan tulisan tangan tegak bersambung.
	Kontribusi	Penggunaan skema ekstraksi fitur diagonal untuk pengenalan karakter tulisan tangan tegak bersambung menghabiskan waktu yang lebih sedikit untuk melatih jaringan saraf.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Image Acquisition</i></li> <li>• <i>Pre-processing : noise removing, binarization, edge detection, dilation and filling</i></li> <li>• <i>Segmentation</i></li> <li>• <i>Feature Extraction</i></li> <li>• <i>Classification and Recognition using Neural Network</i></li> </ul>
	Hasil Utama	Sistem pengenalan jaringan saraf dilatih menggunakan metode ekstraksi fitur horizontal, vertikal, dan diagonal. Hasil dari pengujian tersebut didapat akurasi sebesar 97% dengan metode diagonal.
Batasan	Citra input yang digunakan berasal dari tulisan tangan yang dipindai lalu disimpan ke dalam format .jpeg.	
7	Judul Paper	<i>Deep Learning for Handwritten Javanese Character Recognition</i>
	Penulis	Rismaniyati, Khadijah, Adi Nurhadiyatna
	Jurnal/Konferensi	Proceedings - 2017 1st International Conference on Informatics and Computational Sciences, ICICoS 2017
	URL	<a href="https://ieeexplore.ieee.org/document/8276338/">https://ieeexplore.ieee.org/document/8276338/</a>
	Permasalahan	Karakter Jawa yang jarang digunakan berpengaruh pada kemampuan untuk mengenali karakter tersebut, terutama generasi muda dari suku Jawa.
	Kontribusi	Membandingkan kinerja metode CNN dan DNN dalam mengklasifikasikan tulisan tangan aksara Jawa.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Data Acquisition</i></li> <li>• <i>Preprocessing Data</i></li> <li>• <i>Classification using CNN and DNN</i></li> <li>• <i>Evaluation by K-fold Cross Validation</i></li> </ul>
Hasil Utama	Percobaan dilakukan dengan validasi silang 10 kali lipat dan memvariasikan tingkat pembelajaran, jumlah filter dan jumlah <i>neuron</i> di lapisan tersembunyi. Hasil pengujian menunjukkan model CNN mampu mencapai akurasi terbaik sebesar 70.22% menggunakan 100 <i>neuron</i> pada <i>hidden layer</i> dengan 20 filter, sedangkan DNN mencapai akurasi yang lebih rendah sebesar 64,41% menggunakan 2 <i>hidden layer</i> dan 100-150 <i>neuron</i> .	
Batasan	Belum adanya data primer berisikan aksara Jawa	
8	Judul Paper	<i>Arabic Handwritten Characters Recognition using Convolutional Neural Network</i>
	Penulis	Ahmed El-Sawy, Mohamed Loey, and Hazem El-Bakry
	Jurnal/Konferensi	WSEAS TRANSACTIONS on COMPUTER RESEARCH

	URL	<a href="https://www.researchgate.net/publication/335360305">https://www.researchgate.net/publication/335360305</a>
	Permasalahan	Dataset yang digunakan pada penelitian sebelumnya tidak cukup banyak sehingga hasil yang didapat kurang efektif, dan berkembangnya gaya tulisan tangan karakter Arab memerlukan sistem pengenalan yang semakin akurat.
	Kontribusi	Data input yang lebih banyak dan beragam, berasal dari berbagai rentang usia yang diujikan menggunakan CNN untuk meningkatkan akurasi pengenalan karakter tulisan tangan Arab.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• <i>Preprocessing : convert RGB image to grayscale image, filtering, and smoothing</i></li> <li>• <i>Training and recognizing</i></li> </ul>
	Hasil Utama	CNN bekerja lebih baik dalam mengelola data yang besar. Akurasi yang dicapai sebesar 94.9% pada pengujian gambar, dengan 13440 data <i>training</i> dan 3360 data <i>testing</i> .
	Batasan	Keberagaman data yang digunakan menjadikan perbedaan tiap karakter
9	Judul Paper	Pengenalan Huruf Dan Angka Tulisan Tangan Menggunakan Metode Convolution Neural Network (CNN)
	Penulis	Sam'ani, M.Haris Qamaruzzaman
	Jurnal/Konferensi	Journal Speed – Sentra Penelitian Engineering dan Edukasi
	URL	<a href="http://speed.web.id/ejournal/index.php/speed/article/view/296">http://speed.web.id/ejournal/index.php/speed/article/view/296</a>
	Permasalahan	Pengenalan pola tulisan tangan merupakan satu dari banyak hal yang banyak digunakan dalam kehidupan sehari-hari.
	Kontribusi	Citra input yang digunakan terdiri dari huruf dan angka yang dibaca perhuruf, perkata, dan perkalimat.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• Pengumpulan Data</li> <li>• Implementasi CNN</li> </ul>
	Hasil Utama	Uji coba <i>testing</i> dengan inputan berupa huruf dan angka tunggal sebanyak 184 citra uji diperoleh jawaban benar sebanyak 153 dan jawaban salah sebanyak 31 Sedangkan hasil uji coba dengan input berupa kata dan kalimat yang terdiri dari 191 huruf dan angka dengan jumlah jawaban benar sebanyak 158 dan jawaban salah sebanyak 33 huruf yang tidak dapat dikenali dengan benar.
	Batasan	Citra input kurang diperbaiki saat proses <i>preprocessing</i> , dan jumlah dataset yang kurang banyak.
10	Judul Paper	Perbandingan Kinerja <i>CNN LeNet 5</i> dan <i>Extreme Learning Machine</i> pada Pengenalan Citra Tulisan Tangan Angka
	Penulis	Desi Fitriati
	Jurnal/Konferensi	Jurnal Teknologi Terpadu
	URL	<a href="https://journal.nurulfikri.ac.id/index.php/jt/article/download/45/33">https://journal.nurulfikri.ac.id/index.php/jt/article/download/45/33</a>
	Permasalahan	Proses pencapaian eror yang diinginkan berpengaruh pada waktu yang dibutuhkan saat proses komputasi klasifikasi.
	Kontribusi	CNN <i>LeNet 5</i> dibandingkan dengan ELM untuk mengetahui mana yang lebih unggul dalam akurasi dan komputasi waktu.
	Metode/Solusi	<ul style="list-style-type: none"> <li>• Data Penelitian</li> <li>• Pra-pengolahan Citra Digital dengan Teknik Bilevel Sauvola</li> </ul>



		<ul style="list-style-type: none"> <li>• Ekstraksi Fitur menggunakan operasi ambang batas (<i>Thresholding</i>)</li> <li>• Klasifikasi secara terpisah menggunakan <i>Extreme Learning Machine</i> dan <i>CNN LeNet 5</i></li> </ul>
	Hasil Utama	Pengembangan sistem klasifikasi tulisan tangan angka menggunakan metode <i>CNN LeNet 5</i> lebih unggul dalam hal akurasi yaitu mencapai 98,04% untuk 10.000 data sekunder MNIST dan 78,14% untuk 700 data primer, sedangkan metode ELM lebih unggul dalam hal komputasi waktu yang mencapai 0,00078 mili detik namun dengan akurasi 10% - 30%.
	Batasan	Tidak adanya proses segmentasi untuk sistem pengenalan pola tulisan tangan angka.

Berdasarkan *literatur review* yang dilakukan, didapatkan kesimpulan bahwa pengenalan pola dapat dilakukan pada objek yang bermacam-macam, diantaranya tulisan tangan berupa huruf maupun angka. Metode dengan akurasi tertinggi dicapai oleh *Convolutional Neural Network*. Oleh sebab itu, penelitian ini akan mengimplementasikan metode tersebut untuk melakukan pengenalan pola. Objek yang dipilih yaitu tulisan tangan berupa huruf alfabet karena belum banyak penelitian dalam bidang tersebut dengan akurasi yang didapat pun masih dapat ditingkatkan.