

Implementasi Metode *Convolutional Neural Network* Menggunakan Arsitektur LeNet-5 untuk Pengenalan *Doodle*

Muhammad Rafly Alwanda^{*1}, Raden Putra Kurniawan Ramadhan², Derry Alamsyah³

^{1,2}STMIK Global Informatika MDP; Jl. Rajawali No.14, +62(711)376400/376360

^{1,2}Program Studi Teknik Informatika, STMIK Global Informatika MDP, Palembang

e-mail: *rafly.alwanda17@mhs.mdp.ac.id, radenputrakurniawan@mhs.mdp.ac.id,
derry@mdp.ac.id

Abstrak

Pengenalan objek sampai saat ini banyak diterapkan dalam berbagai bidang, contohnya pada pengenalan tulisan tangan (*handwritten*). Penelitian ini memanfaatkan kemampuan CNN menggunakan arsitektur LeNet-5 untuk pengenalan jenis doodle dengan 5 objek gambar yaitu baju, celana, kursi, kupu – kupu dan sepeda. Setiap objek doodle terdiri dari 30 gambar dengan total jumlah dataset sebanyak 150 gambar. Hasil pengujian menunjukkan bahwa skenario pertama, kedua dan keempat objek sepeda lebih dikenali dengan nilai *accuracy* 93% - 98%, *recall* 86% - 93% dan *precision* 81% - 93%, objek baju lebih dikenali pada skenario ketiga dengan nilai *accuracy* 94%, *recall* 86%, dan *precision* 83%.

Kata kunci—*Convolutional Neural Network, Doodle, LeNet-5, Handwritten*

Abstract

Recognition of objects to date has been widely applied in various fields, for example in handwritten recognition. This research utilizes the ability of CNN to use LeNet-5 architecture for the introduction of doodle types with 5 object images, namely clothes, pants, chairs, butterflies and bicycles. Each doodle object consists of 30 images with a total dataset of 150 images. The test results show that the first, second and fourth scenarios of bicycle objects are more recognized with an accuracy value of 93% - 98%, recall 86% - 93% and precision 81% - 93%, clothes objects are more recognized in the third scenario with an accuracy value of 94%, 86% recall, and 83% precision.

Keywords— *Convolutional Neural Network, Doodle, LeNet-5, Handwritten*

1. PENDAHULUAN

Pengenalan objek sampai saat ini telah banyak diterapkan dalam berbagai bidang, salah satunya adalah pengenalan tulisan tangan (*handwritten*). Ada banyak jenis metode yang digunakan untuk melakukan pengenalan citra, yaitu Kecerdasan Buatan (*Artificial Intelligence*), Pembelajaran Mesin (*Machine Learning*) dan Pembelajaran Dalam (*Deep Learning*). Kecerdasan Buatan (*Artificial Intelligence*) adalah jenis metode yang memiliki cara kerja meniru kecerdasan seperti makhluk hidup untuk menyelesaikan persoalan, sedangkan Pembelajaran Mesin (*Machine Learning*) merupakan pendekatan dari Kecerdasan Buatan yang dapat menirukan atau menggantikan perilaku manusia untuk menyelesaikan masalah dan Pembelajaran Dalam (*Deep Learning*) adalah teknik dalam Pembelajaran Mesin yang menirukan syaraf manusia yang merupakan bagian fundamental dari otak [1]. Salah satu contoh metode klasifikasi yang digunakan dalam pembelajaran mesin yaitu *Multi Layer Perceptron*

(MLP). Penelitian sebelumnya banyak yang menggunakan metode tersebut untuk melakukan pengenalan objek tulisan tangan, contohnya pada pengenalan karakter tulisan tangan alpabet dalam bahasa Bangladesh menggunakan metode *Multi Layer Perceptron* (MLP) mendapatkan tingkat akurasi sebesar 75,08% [2], lalu penelitian pengenalan tulisan tangan berupa kata kunci menggunakan bahasa Arab, Inggris dan Farsi dengan metode *Multi Layer Perceptron* (MLP) menghasilkan tingkat akurasi sebesar 83% pada tulisan Arab, 77% pada tulisan Inggris dan 69% pada tulisan Farsi [3].

Deep Learning adalah sebuah model jaringan syaraf tiruan yang sekarang ini mulai banyak digunakan dan dikembangkan pada kasus pengenalan citra dikarenakan perkembangan teknologi *Graphic Processing Unit* (GPU). *Deep Learning* dinilai dapat memiliki tingkat akurasi yang tinggi. Salah satu contoh *Deep Learning* yang digunakan untuk pengenalan citra adalah *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) adalah pengembangan dari metode *Multi Layer Perceptron* (MLP). Metode *Convolutional Neural Network* (CNN) lebih baik daripada metode *Multi Layer Perceptron* (MLP) dikarenakan metode *Convolutional Neural Network* (CNN) memiliki kedalaman jaringan yang tinggi dan sering diaplikasikan pada data pengenalan citra sehingga mampu menghasilkan tingkat akurasi tinggi dan hasil yang baik, sedangkan metode *Multi Layer Perceptron* (MLP) dinilai kurang baik daripada metode *Convolutional Neural Network* (CNN) dikarenakan metode *Multi Layer Perceptron* (MLP) tidak menyimpan informasi spasial dari data pengenalan citra dan menganggap bahwa setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik [4].

Arsitektur pada *Convolutional Neural Network* (CNN) memiliki kemampuan untuk mengekstraksi fitur secara otomatis. Banyak macam arsitektur *Convolutional Neural Network* (CNN) yang populer untuk digunakan, contohnya LeNet-5 (1998), AlexNet (2012), ZFNet (2013), VGGNet (2014), GoogLeNet (2014), ResNet (2015), FractalNet (2016) dan arsitektur yang lainnya [5]. LeNet-5 adalah arsitektur CNN yang terdiri dari suatu jaringan berlapis banyak sehingga LeNet-5 memiliki jumlah parameter bebas atau jumlah lapisan yang lebih banyak daripada arsitektur lainnya, sedangkan AlexNet, GoogleNet, VGGNet dan ResNet membutuhkan terlalu banyak waktu untuk komputasi dan konfigurasi perangkat keras untuk melatih jaringan syaraf tiruan. LeNet-5 adalah metode CNN pertama yang telah berhasil diterapkan untuk pengenalan digital [6]. LeNet-5 lebih banyak digunakan pada klasifikasi tulisan tangan (*handwritten*) sedangkan AlexNet, ZFNet, VGGNet, GoogleNet dan ResNet sering digunakan untuk klasifikasi gambar, segmentasi, deteksi objek dan penggabungan gambar [7]. Penggunaan arsitektur LeNet-5 yang dikombinasikan dengan metode CNN mampu mencapai tingkat akurasi hingga 98% dengan hanya menggunakan fitur biner yang sederhana [8], pada penelitian klasifikasi tulisan tangan menggunakan arsitektur LeNet-5 sebagai ekstraksi untuk klasifikasi tulisan tangan mendapatkan tingkat akurasi sebesar 98% [9] dan penelitian yang membandingkan antara arsitektur LeNet-5 dan VGG Net dalam rekognisi angka tahun pada prasasti peninggalan kerajaan Majapahit dapat disimpulkan bahwa Performansi LeNet-5 ini lebih baik daripada model VGG Net dengan jarak perbandingan sebesar 11,39% pada 10 *epoch* dalam waktu proses 40223 detik [10].

Penelitian dilakukan untuk mengenali jenis *doodle* yang digambar pada *canvas* berbasis *web* berdasarkan data dan informasi yang diperoleh dari data citra. Konsep *doodle* diaplikasikan pada penelitian ini dikarenakan *doodle* adalah gambaran atau suatu goresan yang dapat dibuat dengan media apapun seperti kertas, spidol, pensil, kuas dan lainnya [11], *doodle* juga dapat dibuat secara langsung tanpa menggunakan sketsa sehingga karya seni ini lebih familiar dengan dunia masyarakat luas, selain itu tujuan lain dari pemilihan *doodle* sebagai objek penelitian ini adalah untuk menarik perhatian mahasiswa agar dapat mengasah tingkat kreativitas dan dapat berfikir lebih luas karena untuk melakukan penggambaran *doodle* hanya membutuhkan imajinasi yang ada didalam pikiran secara spontan pada dasarnya dapat menghasilkan bentuk – bentuk yang imajinatif dan beragam serta dapat digambar dengan kombinasi banyak media.

2. METODE PENELITIAN

2.1 Identifikasi Masalah

Penelitian dimulai dengan mencari topik – topik mengenai perkembangan penelitian tentang pengenalan *doodle*.

2.1.1 Doodle

Doodle adalah gambaran atau goresan yang dapat dibuat dengan menggunakan media seperti kertas, spidol pensil, kuas, pulpen dan lainnya dengan penggambaran pola yang ramai. *Doodle* juga biasa disebut gambaran ilustrasi yang berupa gambaran tangan secara manual atau *sketching*. Berikut ini adalah contoh dari penggambaran *doodle*.



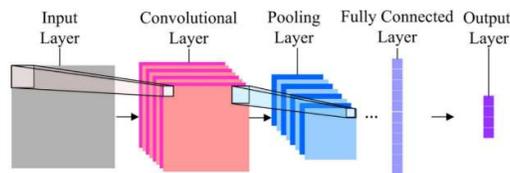
Gambar 1. Contoh Gambaran *Doodle*

2.2 Studi Literatur

Tahapan ini merupakan pencarian beberapa jurnal dan buku yang terkait mengenai pengenalan doodle atau handwriting dengan menggunakan berbagai macam metode yang digunakan, jurnal yang terkait dengan metode Convolutional Neural Network (CNN) dan berbagai macam arsitektur CNN yang digunakan seperti LeNet-5

2.2.1 Convolutional Neural Network

Convolutional Neural Network adalah pengembangan dari Multi Layer Perceptron (MLP) yang digunakan untuk mengolah data dua dimensi. Convolutional Neural Network (CNN) termasuk dalam Deep Neural Network yang banyak digunakan pada citra dengan jaringan yang tinggi dan banyak diaplikasikan pada data penelitian citra. Convolutional Neural Network (CNN) terdiri dari lapis masukan (input layer), lapis keluaran (output layer) dan sejumlah lapisan tersembunyi (hidden Layers). Pada lapisan tersembunyi (hidden layers) berisi convolutional layers, pooling layers dan fully connected layers [12]. Berikut ini adalah gambar lapisan pada metode Convolutional Neural Network.



Gambar 2. Model Lapisan *Convolutional Neural Network*

Convolutional layer adalah lapisan yang digunakan untuk melakukan operasi konvolusi pada *output layer* sebelumnya. *Layer* ini termasuk blok utama pada *Convolutional Neural Network* (CNN) yang didalamnya terdiri dari *filter – filter* yang di pelajari secara acak untuk melakukan operasi konvolusi yang bertujuan sebagai ekstraksi fitur untuk mempelajari representasi fitur dari *input layer*. Tujuan dilakukannya operasi konvolusi pada data citra untuk mengekstraksi fitur dari *input* citra. Konvolusi tersebut akan menghasilkan transformasi *linear* dari data yang di *input* sesuai informasi spasial yang tersedia pada data. Bobot pada *layer*

tersebut akan menspesifikasikan *kernel* konvolusi yang digunakan sehingga *kernel* konvolusi dapat dilatih berdasarkan *input* pada *Convolutional Neural Network* (CNN) [4].

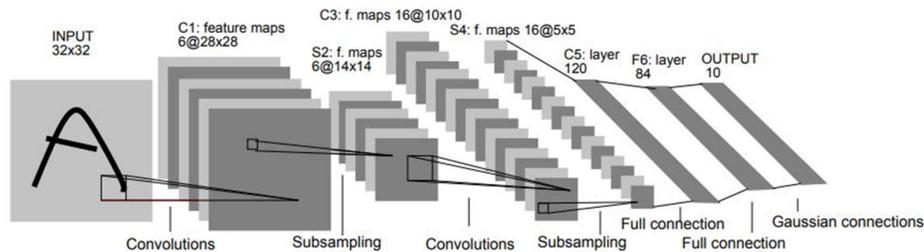
Pooling layer adalah lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map* (*downsampling*) sehingga mempercepat komputasi karena parameter yang diperbarui semakin sedikit. Selain itu, berguna untuk mengekstraksi fitur dominan sehingga proses pelatihan model lebih efektif. Ada dua jenis *pooling layer*, yaitu *max pooling* dan *average pooling* [13]. *Max pooling* mengembalikan nilai maksimum dari bagian gambar yang dicakup oleh kernel sedangkan *average pooling* mengembalikan nilai rata-rata dari bagian gambar yang dicakup oleh kernel.

Fully connected layer adalah lapisan yang digunakan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Untuk mendapatkan hasil keluaran dari *layer* ini tidak dibutuhkan operasi konvolusi, tetapi menggunakan komputasi perkalian matriks yang diikuti dengan *bias offset*. Dengan penggunaan operasi tersebut, setiap *neuron* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya, sehingga *layer* ini disebut sebagai *fully connected layer* [12].

Fungsi aktivasi atau disebut juga *neuron* merupakan fungsi non-linear yang memungkinkan sebuah jaringan syaraf tiruan untuk menyelesaikan permasalahan non-trivial. Setiap fungsi aktivasi mengambil sebuah nilai dan melakukan operasi matematika. Fungsi aktivasi pada arsitektur *Convolutional Neural Network* (CNN) terletak pada perhitungan akhir keluaran *feature map* atau sesudah proses perhitungan konvolusi atau *pooling* untuk menghasilkan suatu pola fitur [14].

2. 2.2 LeNet-5

LeNet-5 adalah suatu jaringan yang memiliki lapisan banyak berbasis *Convolutional Neural Network* (CNN) pertama kali yang dikenalkan oleh Yann LeCun. LeNet 5 terbentuk dari pengembangan – pengembangan versi LeNet sebelumnya, sehingga LeNet 5 ini mempunyai jumlah lapisan yang lebih banyak daripada versi LeNet sebelumnya. Seiring perkembangan dan majunya kecepatan teknologi dari komputer, maka lapisan LeNet 5 ini mengalami penambahan sehingga komputer dapat lebih cepat melakukan komputasi atau perhitungan matematis secara cepat [8]. Berikut ini adalah gambar arsitektur LeNet-5.



Gambar 3. Arsitektur LeNet-5 pada *Convolutional Neural Network*

Input untuk LeNet-5 adalah gambar skala abu-abu yang berukuran 32x32x1 melalui lapisan *convolutional* dengan 6 *feature maps* dengan ukuran *filter* 5x5 dan satu *stride*. 6 *feature maps* ini adalah *channel* dari gambar yang sudah dilakukan operasi konvolusi dengan setiap ukuran 28x28x6. *Stride* digunakan untuk mengontrol seberapa besar pergeseran dari suatu *filter* pada *layer* saat melewati serangkaian data. Pada lapisan ini digunakan fungsi aktivasi tanh yang terdapat pada *kernel size*. Untuk mendapatkan dimensi yang baru, dapat digunakan persamaan (1) dan (2).

$$output\ width = \left(\frac{w - Fw + 2}{Sw} \right) + 1 \quad (1)$$

$$output\ height = \left(\frac{h - Fh + 2p}{sh} \right) + 1 \quad (2)$$

Kemudian Lapisan kedua (S2) merupakan lapisan *pooling layer* dengan ukuran *filter* 2x2, 6 *feature maps* dan dua *stride*. Pada lapisan ini masih sama dengan lapisan sebelumnya menggunakan fungsi aktivasi tanh. Dengan menggunakan rumus (1) dan (2) maka didapatkan dimensi gambar yang dihasilkan menjadi 14x14x6.

Selanjutnya ada lapisan *convolutional* kedua dengan 16 *feature maps* yang memiliki ukuran *filter* 5x5 dengan fungsi aktivasi tanh dan satu *stride*, dengan menggunakan rumus (1) dan (2) maka didapatkan dimensi gambar berukuran 10x10x16.

Lapisan keempat (S4) merupakan *pooling layer* dengan jenis *average pooling layer* atau *max pooling layer* yang memiliki *filter* berukuran 2x2 dengan aktivasi tanh dan dua *stride*. Lapisan ini hampir sama dengan lapisan kedua (S2), hanya saja pada lapisan ini memiliki 16 *feature maps*. Pada lapisan ini terdapat 400 *nodes* yang akan dihubungkan dengan 5x5x16. Dengan menggunakan rumus (1) dan (2) sehingga didapatkan dimensi gambar 5x5x16.

Lapisan kelima (C5) merupakan *fully connected layer* dengan 120 *feature maps* pada masing – masing ukuran 1X1 dengan fungsi aktivasi tanh. Masing – masing dari 120 *nodes* di lapisan kelima terhubung ke semua 400 *nodes* yang ada di lapisan keempat (S4). 120 *nodes* ini telah ditetapkan dari arsitektur LeNet-5. Untuk mendapatkan parameter latih dapat digunakan persamaan berikut ini:

$$trainable\ params = (input\ maps + bias) \times feature\ maps \quad (3)$$

Untuk mendapatkan jumlah *nodes* yang terhubung, dapat digunakan persamaan berikut ini:

$$connections = trainable\ params \times feature\ maps\ size \quad (4)$$

Pada kasus ini, bias selalu bernilai 1. Setelah dihitung menggunakan rumus (3) maka didapatkan parameter latih sebanyak 48120 dan didapatkan jumlah *nodes* yang terhubung yang dihitung menggunakan rumus (4) sebanyak 48120.

Untuk lapisan keenam (F6) merupakan *fully connected layer* dengan *nodes* yang berjumlah sebanyak 84 *nodes*. Untuk mendapatkan parameter latih dan jumlah *nodes* yang terhubung dapat digunakan rumus (3) dan (4), maka didapatkan parameter latih sebanyak 10164 *nodes*.

Pada lapisan terakhir atau *output layer* merupakan *fully connected* yang menggunakan fungsi aktivasi *softmax* dengan *size* 5 berdasarkan dengan hasil *output* gambar yang diklasifikasikan.

2. 2.3 Tensorflow

Tensorflow adalah sebuah *library* pada *machine learning* atau sering disebut dengan *deep learning*, Tensorflow mampu digunakan untuk melakukan pengenalan suara, mengenali wajah dalam sebuah foto dan pengenalan objek pada gambar yang dapat digunakan pada sebuah aplikasi perangkat lunak yang berbasis *open source* [15].

2. 2.4 Keras

Keras merupakan perangkat lunak jaringan yang berbasis *open source* ditulis dengan menggunakan bahasa Python. Keras juga dapat dijalankan menggunakan MXNet, Tensorflow, Deeplearning4j, theano atau CNTK yang khusus dirancang guna mempercepat eksperimen yang berhubungan dengan *deep learning*. Keras dikembangkan untuk eksperimen proyek ONEIROS

(*Open-ended Neuro-Electronic Intelligent Robot Operating System*) yang dikelola dan ditulis oleh Francois Chollet, seorang insinyur yang bekerja di Google [16].

2.3 Pengumpulan Data

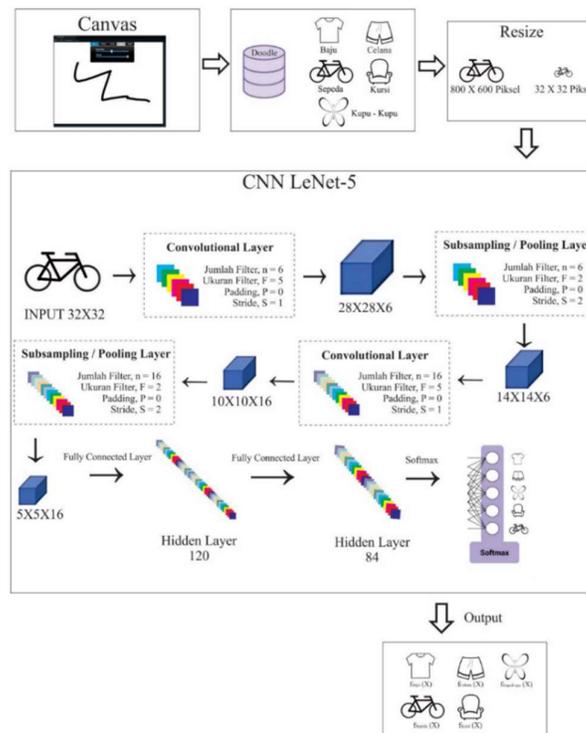
Pengumpulan data menggunakan *dataset* yang berjumlah 150 gambar yang didapatkan dari 30 orang dengan rentang usia 17 sampai 24 tahun, masing – masing orang menggambar 5 jenis *doodle* yaitu sepeda, celana, baju, kursi dan kupu – kupu. Setiap objek *dataset* memiliki resolusi sekitar 32x32 piksel dan berformat *.JPG. Berikut adalah jumlah data yang terdapat pada setiap jenis *doodle* dapat dilihat pada tabel 1.

Tabel 1. Data Setiap Jenis *Doodle*

Jenis – Jenis <i>Doodle</i>	Jumlah Masing – Masing <i>Dataset</i>
Sepeda	30 Gambar
Celana	30 Gambar
Baju	30 Gambar
Kursi	30 Gambar
Kupu - Kupu	30 Gambar
Total	150 Gambar

2.4 Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem, proses ini dapat dilihat pada gambar berikut.



Gambar 4. Rancangan Metodologi

Perancangan sistem dilakukan dengan membuat perancangan aplikasi *canvas* berbasis *web* yang digunakan untuk melakukan penggambaran *doodle*, selain itu *canvas* juga dibutuhkan sebagai media untuk melakukan pengambilan *dataset*. *Dataset* yang telah didapatkan akan di *resize* sesuai *input* citra yang diterima oleh arsitektur LeNet-5 yaitu 32x32 piksel. Setelah *dataset* di *resize*, kemudian dilakukan pengolahan menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur LeNet-5. Diawali dengan *input layer* dengan ukuran 32x32 piksel, lalu dilakukan operasi konvolusi pada *convolutional layer* dengan *filter* 5x5 sehingga dimensi gambar berubah menjadi 28x28x6 piksel, setelah itu dilakukan lagi operasi konvolusi pada *convolutional layer* dengan jumlah 16 *feature maps* sehingga dimensi gambar berkurang menjadi 10x10x16 piksel, lalu ke tahap *pooling layer* dengan jumlah 16 *feature maps* sehingga ukuran piksel menjadi 5x5x16, lalu dilakukan transformasi dimensi menjadi bentuk linear pada *fully connected layer* dengan jumlah 120 *feature maps* terhubung ke semua 400 *node*, dari tahap *input layer* sampai tahap *fully connected layer* menggunakan fungsi aktivasi tanh untuk mendapatkan nilai *output*, setelah itu dilakukan perkalian matriks diikuti dengan 84 *bias offset* dan tahap terakhir pada *softmax*. Pada bagian *softmax* ini keluaran *output* akan menghasilkan bentuk neuron yang saling terhubung kepada objek yang telah ditentukan seperti baju, celana, kursi, kupu-kupu dan sepeda, hasil akhir *output* berupa persentase tergantung peluang tingkat kemiripan. Setelah proses selesai, maka data hasil telah didapatkan dan dapat digunakan pada saat pengujian untuk mengenali objek *doodle*.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi

Tahap ini mengimplementasikan proses pada penelitian untuk memperoleh hasil yang dibutuhkan pada penelitian ini.

3.1.1 Implementasi Metode Convolutional Neural Network (CNN)

Pada tahap ini dilakukan *resize* terhadap citra *doodle* sebanyak 150 gambar yang digambar pada *canvas* untuk dilatih, citra awal berukuran 800x600 piksel lalu di *resize* menjadi 32x32 piksel. Kemudian setiap jenis citra *doodle* diberi label sesuai dengan klasifikasinya lalu dilakukan proses pelatihan (*training*) menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur LeNet-5. Pada penelitian ini digunakan dua cara pelatihan yang berbeda, yaitu menggunakan *average pooling* pada *pooling layer* untuk data latih pertama dan menggunakan *max pooling* pada *pooling layer* untuk data latih kedua, sehingga didapatkan 2 model yang berbeda, yaitu model data latih menggunakan *average pooling* dan model data latih menggunakan *max pooling*. Untuk tahap pelatihan menggunakan *average pooling* dapat dilihat pada Gambar 5.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 6)	156
average_pooling2d (AveragePo	(None, 14, 14, 6)	0
conv2d_1 (Conv2D)	(None, 10, 10, 16)	2416
average_pooling2d_1 (Average	(None, 5, 5, 16)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 120)	48120
dense_1 (Dense)	(None, 84)	10164
dense_2 (Dense)	(None, 5)	425
Total params: 61,281		
Trainable params: 61,281		
Non-trainable params: 0		

Gambar 5. Ringkasan dari Arsitektur LeNet-5 menggunakan *Average Pooling*

Pada *Convolutional Layer*, citra yang di masukkan berukuran 32x32 piksel dilakukan operasi konvolusi dengan *filter size* 5x5 dan *feature maps* yang berjumlah 6 *channel* sehingga dimensi gambar berubah menjadi 28x28x6 piksel, lalu pada *average pooling* dilakukan pengurangan dimensi dari *feature maps* (*downsampling*) sehingga gambar berubah menjadi 14x14x6 piksel, setelah itu dilakukan lagi operasi konvolusi pada *convolutional layer* dengan jumlah 16 *feature maps* sehingga dimensi gambar berkurang menjadi 10x10x16 piksel, lalu ke tahap *pooling layer* dengan jumlah 16 *feature maps* sehingga ukuran piksel menjadi 5x5x16. Pada tahap ini *feature maps* yang dihasilkan masih berbentuk *multi dimensional array*, sehingga dilakukan *flatten* atau *reshape feature maps* menjadi bentuk 1 dimensi *array* agar dapat digunakan sebagai *input* dari *fully connected layer* dengan jumlah 120 *feature maps* yang terhubung ke semua 400 *node*. Dimulai dari tahap *convolutional layer* sampai tahap *fully connected layer* / *dense layer* menggunakan fungsi aktivasi *tanh* untuk mendapatkan nilai *output*, setelah itu dilakukan perkalian matriks diikuti dengan sebanyak 84 *bias offset* dan tahap terakhir pada *softmax* keluaran *output* sebanyak 5 *node* yang akan menghasilkan bentuk neuron yang saling terhubung kepada objek doodle yang telah diberikan label, seperti baju, celana, kupu – kupu, kursi, dan sepeda. Hasil data yang telah dilatih akan disimpan kedalam model yang dapat digunakan untuk pengujian. Selanjutnya melakukan pelatihan menggunakan *max pooling* pada *pooling layer*, untuk tahap – tahap pelatihannya dapat dilihat pada Gambar 6.

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 28, 28, 6)	156
max_pooling2d_10 (MaxPooling)	(None, 14, 14, 6)	0
conv2d_11 (Conv2D)	(None, 10, 10, 16)	2416
max_pooling2d_11 (MaxPooling)	(None, 5, 5, 16)	0
flatten_5 (Flatten)	(None, 400)	0
dense_15 (Dense)	(None, 120)	48120
dense_16 (Dense)	(None, 84)	10164
dense_17 (Dense)	(None, 5)	425
Total params: 61,281		
Trainable params: 61,281		
Non-trainable params: 0		

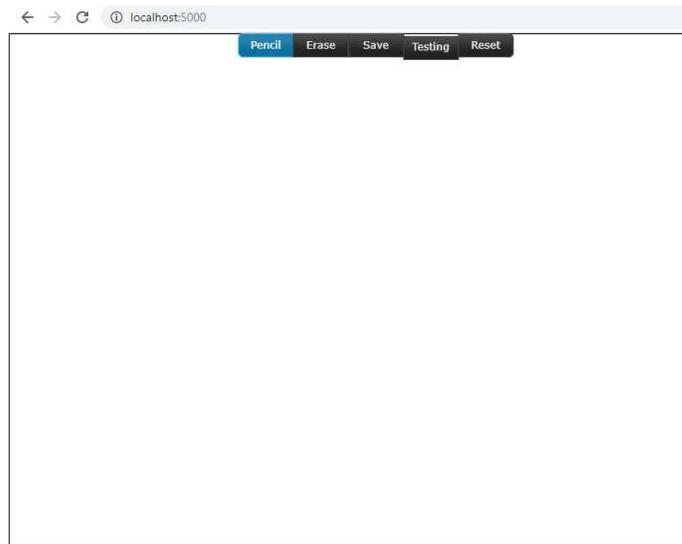
Gambar 6. Ringkasan dari Arsitektur LeNet-5 menggunakan *Max Pooling*

3. 1.2 Implementasi Antarmuka Aplikasi Canvas

Tampilan akhir antarmuka aplikasi *canvas* yang dibangun adalah sebagai berikut :

3. 1.2.1 Halaman Depan Canvas

Gambar 7 adalah tampilan halaman pertama aplikasi yang muncul ketika aplikasi dijalankan. Halaman ini digunakan untuk menggambar objek *doodle* yang akan dilakukan pengujian. Pada halaman ini terdapat tombol *pencil* yang dapat digunakan untuk menambahkan goresan pada *canvas*, tombol *erase* untuk menghapus goresan pada *canvas*, tombol *save* untuk menyimpan goresan pada *canvas*, tombol *testing* untuk melakukan pengujian pada objek *doodle* yang telah digambar dan tombol *reset* digunakan untuk mengatur ulang *canvas* agar kembali pada posisi awal.



Gambar 7. Tampilan Halaman Depan *Canvas*

3. 1.2.2 Halaman *Testing Canvas*

Gambar 8 adalah tampilan halaman pengujian, halaman ini tampil jika pengguna menekan tombol *testing* pada gambar 4.4. Pada halaman ini terdapat tampilan hasil dari pengujian menggunakan *average pooling* dan hasil dari pengujian menggunakan *max pooling* beserta label dari gambar yang di tebak. Selain itu, terdapat tombol *start* yang digunakan untuk kembali kehalaman depan aplikasi *canvas*.



Gambar 8. Tampilan Halaman *Testing Canvas*

3. 2 Evaluasi

Pada tahap ini terdapat empat skenario pengujian, yaitu :

1. Penggambaran objek *doodle* secara baik dengan menggunakan *max pooling* pada *pooling layer*.
2. Penggambaran objek *doodle* secara garis putus - putus dengan menggunakan *max pooling* pada *pooling layer*.
3. Penggambaran objek *doodle* secara baik menggunakan *average pooling* pada *pooling layer*.

4. Penggambaran objek *doodle* secara garis putus – putus menggunakan *average pooling* pada *pooling layer*.

Keempat skenario ini digunakan pada setiap pengujian pengenalan jenis *doodle*. Sebanyak 300 gambar citra digunakan untuk pengujian yang terdiri dari 150 gambar baik dan 150 gambar dengan objek garis putus - putus. Gambar yang digunakan untuk pengujian didapatkan dari 30 orang dengan masing – masing orang menggambar 5 jenis *doodle* yaitu baju, celana, kupu – kupu, kursi, dan sepeda secara baik dan secara garis putus – putus. Hasil dari pengujian akan dihitung untuk mendapatkan tingkat keberhasilan dengan menggunakan *confusion matrix* agar didapatkan nilai *precision*, *recall*, dan *accuracy*. Berikut ini adalah tabel hasil pengujian per-kelas.

Tabel 2. Hasil Pengujian Per-Kelas

Skenario		Confusion Matrix	Objek Doodle				
			Baju	Celana	Kupu - Kupu	Kursi	Sepeda
Max Pooling	Citra Gambar Baik	Accuracy	92,66%	92,67%	89,33%	92,67%	95,33%
		Recall	90,00%	93,33%	63,33%	70,00%	90,00%
		Precision	77,14%	75,67%	79,17%	91,30%	87,10%
	Citra Gambar Putus - Putus	Accuracy	92,67%	94,67%	88,00%	92,00%	98,00%
		Recall	90,00%	90,00%	60,00%	76,67%	96,67%
		Precision	77,14%	84,38%	75,00%	82,14%	93,55%
Average Pooling	Citra Gambar Baik	Accuracy	94,00%	86,00%	67,37%	79,33%	90,00%
		Recall	86,67%	86,67%	31,43%	6,67%	83,33%
		Precision	83,87%	60,47%	61,11%	40,00%	71,43%
	Citra Gambar Putus - Putus	Accuracy	91,33%	83,33%	86,00%	80,67%	93,33%
		Recall	83,33%	90,00%	66,67%	10,00%	86,67%
		Precision	75,76%	55,10%	64,52%	60,00%	81,25%

Berikut ini adalah tabel yang menunjukkan hasil pengujian keseluruhan berdasarkan skenario.

Tabel 3. Hasil Pengujian Skenario ke-1

Kategori	Baju	Celana	Kupu - Kupu	Kursi	Sepeda	Total
Total Data Uji Benar	27	28	19	21	27	122
Total Seluruh Data yang Diuji	30	30	30	30	30	150
$Accuracy \text{ Keseluruhan} = \frac{\Sigma \text{ Data Uji Benar}}{\Sigma \text{ Total Uji}} 100\% = \frac{122}{150} \times 100\% = 81.3\%$						

Tabel 4. Hasil Pengujian Skenario ke-2

Kategori	Baju	Celana	Kupu - Kupu	Kursi	Sepeda	Total
Total Data Uji Benar	27	27	18	23	29	124
Total Seluruh Data yang Diuji	30	30	30	30	30	150

$$Accuracy \text{ Keseluruhan} = \frac{\Sigma \text{ Data Uji Benar}}{\Sigma \text{ Total Uji}} 100\% = \frac{124}{150} \times 100\% = 82.6\%$$

Tabel 5. Hasil Pengujian Skenario ke-3

Kategori	Baju	Celana	Kupu - Kupu	Kursi	Sepeda	Total
Total Data Uji Benar	26	26	22	2	25	101
Total Seluruh Data yang Diuji	30	30	30	30	30	150
$Accuracy \text{ Keseluruhan} = \frac{\Sigma \text{ Data Uji Benar}}{\Sigma \text{ Total Uji}} 100\% = \frac{101}{150} \times 100\% = 67.3\%$						

Tabel 6. Hasil Pengujian Skenario ke-4

Kategori	Baju	Celana	Kupu - Kupu	Kursi	Sepeda	Total
Total Data Uji Benar	25	27	20	3	26	101
Total Seluruh Data yang Diuji	30	30	30	30	30	150
$Accuracy \text{ Keseluruhan} = \frac{\Sigma \text{ Data Uji Benar}}{\Sigma \text{ Total Uji}} 100\% = \frac{101}{150} \times 100\% = 67.3\%$						

4. KESIMPULAN

Berdasarkan hasil pengujian pada pengenalan *doodle* menggunakan metode *Convolutional Neural Network* (CNN) yang telah dilakukan dan telah diuraikan dalam laporan ini, maka dapat ditarik kesimpulan sebagai berikut.

1. Pada pengujian skenario pertama, kedua dan keempat objek *doodle* sepeda lebih dikenali dibandingkan keempat objek lainnya dengan nilai *accuracy* sebesar 93% - 98%, *recall* 86% - 93% dan *precision* sebesar 81% - 93%, sedangkan pada pengujian skenario ketiga objek *doodle* baju lebih dikenali dibandingkan keempat objek lainnya dengan nilai *accuracy* 94%, *recall* 86%, dan *precision* 83%. Maka dapat disimpulkan penggunaan metode *Convolutional Neural Network* (CNN) dengan arsitektur LeNet-5 pada penelitian ini objek *doodle* sepeda lebih banyak dikenali.
2. Untuk pengenalan *doodle* menggunakan metode *Convolutional Neural Network* (CNN), penggunaan *max pooling* lebih baik daripada menggunakan *average pooling* pada *pooling layer* yang terdapat pada arsitektur LeNet-5 dikarenakan dalam pengenalan objek tingkat akurasi keseluruhan yang dihasilkan rata – rata sebesar 81% pada *max pooling* dan 67% pada *average pooling*, ketika melakukan penggambaran secara baik maupun penggambaran secara garis putus – putus.

5. SARAN

Adapun saran – saran yang diberikan untuk dilakukan pada penelitian selanjutnya adalah sebagai berikut.

1. Menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur lain seperti AlexNet, VGGNet, GoogLeNet, ResNet, FractalNet dan arsitektur yang lainnya.
2. Menambahkan *Transfer Learning* pada arsitektur yang akan digunakan.

3. Menambahkan lebih banyak jumlah citra *doodle* untuk hasil yang lebih baik.
4. Menggunakan fungsi aktivasi lain seperti sigmoid atau reLu.

DAFTAR PUSTAKA

- [1] Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Jurnal Teknologi Indonesia*.
- [2] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M., & Basu, D. K. (2012). *Handwritten Bangla Alphabet Recognition using an MLP Based Classifier*. 285–291. Diambil dari <http://arxiv.org/abs/1203.0882>
- [3] Tavoli, R., & Keyvanpour, M. (2018). A method for handwritten word spotting based on particle swarm optimisation and multi-layer perceptron. *IET Software*, 12(2), 152–159. <https://doi.org/10.1049/iet-sen.2017.0071>
- [4] Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- [5] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... Asari, V. K. (2018). *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches*. Diambil dari <http://arxiv.org/abs/1803.01164>
- [6] Wang, S., Wu, T. H., Shao, T., & Peng, Z. X. (2019). Integrated model of BP neural network and CNN algorithm for automatic wear debris classification. *Wear*, 426–427(September 2018), 1761–1770. <https://doi.org/10.1016/j.wear.2018.12.087>
- [7] Zhou, T., Ruan, S., & Canu, S. (2019). A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 100004. <https://doi.org/10.1016/j.array.2019.100004>
- [8] Fitriati, D. (2016). Perbandingan Kinerja CNN LeNet 5 Dan Extreme Learning Machine Pada Pengenalan Citra Tulisan Tangan Angka. *Jurnal Teknologi Terpadu*, 2(1), 10–16.
- [9] Zhao, H., & Liu, H. (2019). Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition. *Granular Computing*, 0(0), 0. <https://doi.org/10.1007/s41066-019-00158-6>
- [10] Septianto, T., Setyati, E., & Santoso, J. (2018). *Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit*. 6(April), 106–109. <https://doi.org/10.14710/jtsiskom.6.3.2018.106-109>
- [11] Mariana, G. (2017). *Cara Mudah Menggambar Doodle Menggunakan Pensil*. Diambil dari <https://books.google.co.id/books?isbn=602103659X>
- [12] Suyanto. (2018). *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika Bandung.
- [13] Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks. Diambil 5 September 2019, dari <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [14] Zufar, M. (2016). *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real - Time*. 5(2), 72–77.
- [15] Rampasek, L., & Goldenberg, A. (2016). TensorFlow: Biology's Gateway to Deep Learning? *Cell Systems*, 2(1), 12–14. <https://doi.org/10.1016/j.cels.2016.01.009>
- [16] Chollet, F. (2015). Keras: The Python Deep Learning library. Diambil dari <https://keras.io>