

**IMPLEMENTASI DEEP LEARNING UNTUK IMAGE
CLASSIFICATION MENGGUNAKAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK (CNN) PADA
CITRA KEBUN DAN SAWAH**

(Studi Kasus : Klasifikasi Gambar Pada Citra Sawah dan Kebun)



**PROGRAM STUDI STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2020

HALAMAN PERSETUJUAN PEMBIMBING

TUGAS AKHIR

Judul : Implementasi *Deep Learning* untuk *Image Classification* menggunakan Algoritma *Convolutional Neural Network* (CNN) pada Citra Sawah dan Kebun

Nama Mahasiswa : Meiga Isyatan Mardiyah

NIM : 16611087

**TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK
DIUJIKAN**

Yogyakarta, 4 Juni 2020

Pembimbing

(Tuti Purwaningsih, S.Stat., M.Si.)

HALAMAN PENGESAHAN

TUGAS AKHIR

**IMPLEMENTASI DEEP LEARNING UNTUK IMAGE
CLASSIFICATION MENGGUNAKAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK (CNN) PADA
CITRA KEBUN DAN SAWAH**

Nama Mahasiswa : Meiga Isyatan Mardiyah

NIM : 16611087

**TUGAS AKHIR INI TELAH DIUJIKAN
PADA TANGGAL: 15 Juni 2020**

Nama Penguji:

Tanda Tangan

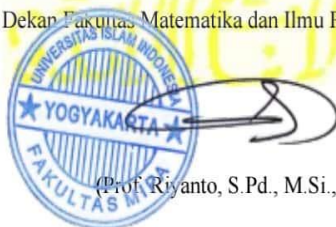
1. Dr.techn.Rohmatul Fajriyah, S.Si., M.Si.

2. Dina Tri Utari, S.Si., M.Sc.

3. Tuti Purwaningsih, S.Stat., M.Si.

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



(Prof. Riyanto, S.Pd., M.Si., Ph.D.)

KATA PENGANTAR

Assalamu'alaikum Warahmatullaahi Wabarakaatuh...

Alhamdulillah "aalamiin, puji syukur senantiasa penulis panjatkan kepada Allah SWT yang Maha Pengasih lagi Maha Panyayang serta penulis panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, Shalawat serta salam Insya Allah selalu tercurah kepada junjungan umat, yaitu Nabi Muhammad SAW sebagai tauladan para sahabat dan pengikutnya yang senantiasa berjuang untuk agama Islam hingga akhir sehingga penulis dapat menyelesaikan Tugas Akhir.

Penyelesaian tugas akhir ini tidak terlepas dari bantuan, dukungan, arahan, dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan terima kasih kepada :

1. Orang tua penulis, Bapak Sutarno dan Ibu Nur Fauziah yang selalu luar biasa mendoakan, memberikan dukungan dan pantang menyerah bekerja keras demi kelancaran studi penulis.
2. Adik-adik, yaitu Istianatul Khairat dan Mhammad Fajar Wicaksono, serta keluarga lainnya yang selalu mendo'akan, memberikan semangat serta dukungan kepada penulis.
3. Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Dr.Edy Widodo, S.Si., M.Si. selaku Ketua Program Studi Statistika.
5. Tuti Purwaningsih, S.Stat., M.Si. selaku dosen pembimbing yang memberikan saran, masukan yang membangun serta arahan yang positif saat bimbingan.
6. Seluruh staff pengajar Program Studi Statistika Universitas Islam Indonesia yang telah memberikan bekal ilmu kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini.
7. Teman teman sepermainan sejak SMP dan SMA yaitu Indrianti Ismayani, Nurzaitun, Rizky Akhirunisyah, Dhurzatun Alyah, Ajeng Permatasari, dan Nur Inayah yang selalu menemani sampai detik ini.

8. Teman-teman sepermainan dan seperjuangan yaitu Adelian Rahma Saraswati, Nabila Wuri Hutaminingsih, Putri Choirunisa dan Rizka Yolanda Werdiningsih Wiyono yang selalu membuat rusuh tetapi saling mengingatkan untuk menyelesaikan studi dalam menyelesaikan tugas akhir ini serta memberikan semangat satu sama lain.
9. Rere Nurkholis yang selalu membatu, memberikan semangat, dan selalu memberikan dukungan sampai tugas akhir ini selesai.
10. Seluruh teman-teman satu bimbingan tugas akhir, yang senantiasa berbagi ilmu dan informasi yang bermanfaat sampai tugas akhir ini selesai.
11. Teman-teman Statistika UII Angkatan 2016 yang sedang berjuang bersama untuk meraih gelar S.Stat, terimakasih atas pengalaman yang berharga selama menjadi mahasiswa Statistika UII.
12. Pihak-pihak lain yang tidak penulis sebutkan satu per satu, yang telah banyak memotivasi dan meningkatkan kembali semangat penulis dalam menyusun tugas akhir.

Demikian Tugas Akhir ini, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan bantuan baik moril maupun materil sehingga tugas akhir ini dapat diselesaikan. Penulis mohon maaf apabila selama proses penyusunan tugas akhir ini terdapat kekhilafan dan kesalahan. Penulis menyadari sepenuhnya keterbatasan kemampuan dalam penulisan tugas akhir ini, oleh karena itu penulis mengharap adanya kritik dan saran yang membangun demi kesempurnaan penyusunan dan penulisan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi semua yang membaca dan membutuhkan, *Aamiin aamiin ya robbal'alamiin.*

Wassalamu 'alaikum Warahmatullaahi Wabarakaatuh...

Yogyakarta, (24 Februari 2020)



(Meiga Isyatan Mardiyah)

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PERSETUJUAN PEMBIMBING	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR	ix
DAFTAR LAMPIRAN.....	x
PERNYATAAN.....	xi
ABSTRAK	xii
ABSTRACT.....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB 2 TINJAUAN Pustaka.....	4
BAB 3 LANDASAN TEORI	9
3.1 Pengertian Sawah	9
3.2 Pengertian Kebun	9
3.3 Citra	10
3.3.1 Pengertian Citra	10
3.3.2 Pengertian Citra Digital	10
3.3.3 Jenis-jenis Citra Digital	12
3.3.4 Pengolahan Citra.....	12
3.4 <i>Artificial Intelligence (AI)</i>	13
3.5 <i>Machine Learning</i>	13
3.6 <i>Deep Learning</i>	14

3.7	<i>Artificial Neural Network (ANN)</i>	16
3.8	<i>Convolutional Neural Network (CNN)</i>	17
3.8.1	Convolution Layer.....	18
3.8.2	Pooling Layer.....	19
3.8.3	Fully Connected Layer.....	20
3.9	<i>Confusion Matrix</i>	21
3.10	<i>Precision, Recall, dan Akurasi</i>	21
BAB 4	METODOLOGI PENELITIAN.....	23
4.1	Populasi dan Sampel Penelitian.....	23
4.2	Variabel dan Definisi Variabel Penelitian.....	23
4.3	Jenis dan Sumber Data.....	23
4.4	Metode Analisis Data.....	23
4.5	Tahapan Penelitian.....	23
4.6	Perangkat Pengujian.....	24
BAB 5	HASIL DAN PEMBAHASAN.....	25
5.1	Dataset Sawah dan Kebun.....	25
5.2	<i>Preprocessing</i> Dataset.....	25
5.3	Skenario Dataset <i>Training</i> dan <i>Testing</i>	26
5.4	Perancangan CNN.....	27
5.4.1	Membandingkan Nilai Epoch.....	28
5.4.2	Membandingkan Jumlah Data Train dan Data Test.....	28
5.5	Arsitektur CNN.....	30
5.6	Hasil Klasifikasi.....	34
BAB 6	PENUTUP.....	36
6.1.	Kesimpulan.....	36
6.2.	Saran.....	36
	DAFTAR PUSTAKA.....	37
	LAMPIRAN.....	40

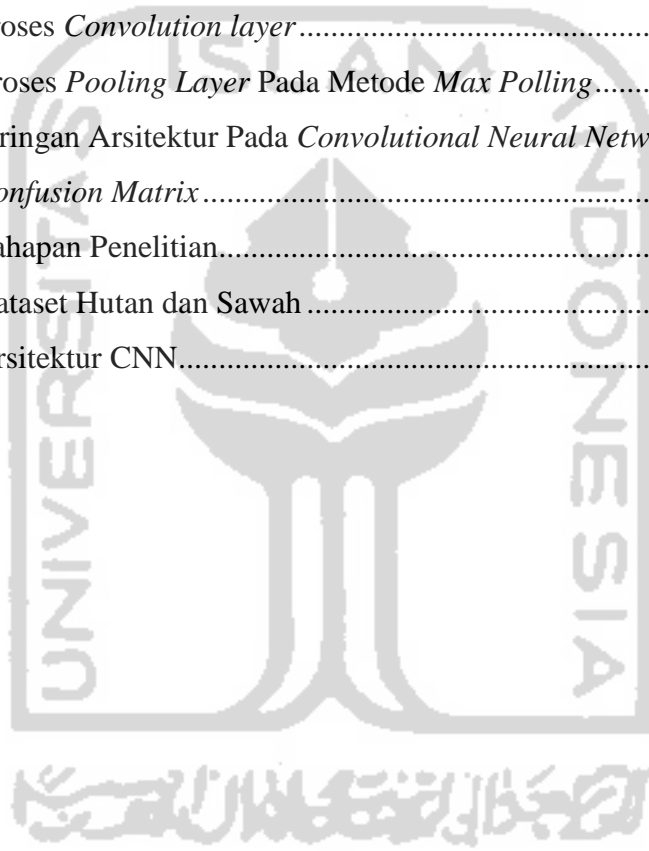
DAFTAR TABEL

Tabel 4.1. Definisi Variabel Penelitian	23
Tabel 5.1. Skenario Pembagian Data	27
Tabel 5.2. Perbandingan Jumlah Data <i>Train</i> dan Data <i>Test</i>	28
Tabel 5.3. Model Terbaik	32
Tabel 5.4. Hasil Klasifikasi Model Terbaik dengan Data <i>Testing</i> (Uji).....	34



DAFTAR GAMBAR

Gambar 3.1. Sawah.....	9
Gambar 3.2. Kebun.....	10
Gambar 3.3. Citra Digital Dua Dimensi.....	12
Gambar 3.4. Jaringan Saraf Manusia	16
Gambar 3.5. Proses <i>Convolution layer</i>	18
Gambar 3.6. Proses <i>Pooling Layer</i> Pada Metode <i>Max Polling</i>	19
Gambar 3.7. Jaringan Arsitektur Pada <i>Convolutional Neural Network</i>	20
Gambar 3.8. <i>Confusion Matrix</i>	21
Gambar 4.1. Tahapan Penelitian.....	24
Gambar 5.1. Dataset Hutan dan Sawah.....	26
Gambar 5.2. Arsitektur CNN.....	30



DAFTAR LAMPIRAN

Lampiran 1 Program R Studio	40
-----------------------------------	----



PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 4 Juni 2020



(Meiga Isyatan Mardiyah)



ABSTRAK

IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA KEBUN DAN SAWAH

(Studi Kasus : Klasifikasi Gambar Pada Citra Kebun dan Sawah)

Meiga Isyatan Mardiyah

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Indonesia merupakan salah satu negara agraris dengan berbagai macam kekayaan alam seperti pertanian dan perkebunan. Pertanian dan perkebunan di Indonesia sangat beraneka ragam, seperti sawah yang dapat menghasilkan padi, kedelai, jagung, umbi-umbian, dan lain-lainnya. Sedangkan, perkebunan di Indonesia seperti kebun dengan hasil berupa kopi, teh, lada, kelapa sawit, pala, dan lain-lain. sawah yang merupakan contoh pertanian dan perkebunan yang merupakan contoh kebun memiliki karakteristik yang sama sehingga sulit untuk dibedakan oleh masyarakat atau orang awam apabila dilihat dengan menggunakan foto udara atau foto yang diambil dari ketinggian tertentu serta membantu pemerintah dalam melihat ketahanan pangan yang ada di Indonesia. Salah satu cara untuk membentuk model yaitu dengan memanfaatkan ilmu komputasi yang salah satu tujuannya adalah untuk mengambil informasi dari citra digital untuk mengenal *obyek* secara otomatis. Salah satu metode *deep learning* yang sedang berkembang saat ini yaitu *Convolutional Neural Network* (CNN). Metode CNN merupakan metode yang masukkan (*input data*) berupa citra. Metode ini memiliki lapisan khusus yang bernama lapisan konvolusi dimana pada lapisan sebuah citra masukkan (*input citra*) akan menghasilkan sebuah pola dari beberapa bagian citra yang nantinya akan lebih mudah untuk diklasifikasikan. Lapisan konvolusi memiliki fungsi pembelajaran citra menjadi lebih efisien untuk diimplementasikan. Oleh karena itu peneliti ingin memanfaatkan metode CNN ini untuk dapat mengklasifikasi kebun dan sawah, dengan tujuan untuk membedakan karakteristik kebun dan sawah. Berdasarkan hasil klasifikasi diperoleh hasil akurasi testing sebesar 75%. Dapat disimpulkan bahwa metode CNN dapat mengklasifikasikan citra kebun dan sawah dengan baik.

Kata Kunci: *Convolutional Neural Network (CNN)*, Citra Sawah, Citra Kebun,
Klasifikasi



ABSTRACT

IMPLEMENTATION OF DEEP LEARNING FOR IMAGE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM IN GARDEN AND RICE FIELDS IMAGES

(Case Study: Image Classification of Gardens and Rice Field Imagery)

Meiga Isyatan Mardiyah

Program Studi Statistika, Fakultas MIPA

Universitas Islam Indonesia

Indonesia is an agricultural country with a variety of natural resources such as agriculture and plantations. Agriculture and plantations in Indonesia are very diverse, such as rice fields that can produce rice, soybeans, corn, tubers, and others. Meanwhile, plantations in Indonesia such as gardens with the results of coffee, tea, pepper, oil palm, nutmeg, and others. rice fields which are examples of agriculture and plantations which are examples of gardens have the same characteristics so that it is difficult to be distinguished by the public or ordinary people when viewed using aerial photographs or photographs taken from certain heights and assisting the government in seeing food security in Indonesia. One way to form a model is to utilize computational science, one of which is to take information from digital images to recognize objects automatically. One method of deep learning that is currently developing is Convolutional Neural Network (CNN). CNN method is a method that enters (input data) in the form of an image. This method has a special layer called the convex layer where in an input layer (input image) will produce a pattern of several parts of the image which will be easier to classify later. The convolution layer has an image learning function that is more efficient to implement. Therefore researchers want to utilize this CNN method to be able to classify gardens and rice fields, with the aim to distinguish the characteristics of gardens and rice fields. Based on the classification results obtained 75% accuracy testing results. It can be concluded that the CNN method can classify the image of gardens and rice fields well.

Keywords: *Convolutional Neural Network (CNN), Rice Field Image, Garden Image, Classification.*

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah

Indonesia merupakan salah satu negara agraris dengan berbagai macam kekayaan alam seperti pertanian dan perkebunan. Menurut organisasi pangan dan pertanian dunia, Indonesia memiliki lahan persawahan padi paling produktif di Asia dengan total lahan sawah mencapai 4,9 juta hektar. Masyarakat Indonesia juga banyak sekali yang bekerja sebagai petani dan pekebun karena daerah-daerah di Indonesia sebagian besar mempunyai lahan pertanian dan perkebunan. (Indonesia.go.id,2019).

Salah satu contoh jenis perkebunan yaitu kebun teh, kebun karet, kebun kopi, kebun kelapa sawit, kebun tembakau, kebun tebu dan lain lain. Berdasarkan data Badan Pusat Statistik 2019, Jenis tanaman perkebunan rakyat mencapai 17.430 ribu hektar yang terdiri dari kebun karet dengan luas 3246,1 Ribu hektar (ha), kebun kelapa dengan luas 3380,4 Ribu hektar (ha), kebun kelapa sawit dengan luas 6035,7 Ribu hektar (ha), kebun kopi dengan luas 1215,5 Ribu hektar (ha), kebun kakao dengan luas 1574,3 Ribu hektar (ha), kebun teh dengan luas 51,5 Ribu hektar (ha), kebun jambu mete dengan luas 495,2 Ribu hektar (ha), kebun pala dengan luas 230,9 Ribu hektar (ha), kebun lada dengan luas 180,9 Ribu hektar (ha), kebun cengkeh dengan luas 560,8 Ribu hektar (ha) dengan kebun tanaman semusim seperti kebun gula tebu dengan luas 232,9 Ribu hektar (ha), kebun tembakau dengan luas 204,7 Ribu hektar (ha), dan kebun nilam dengan luas 21,4 Ribu hektar (ha). (Badan Pusat Statistik, 2019).

Pertanian dan perkebunan di Indonesia sangat beraneka ragam, seperti sawah yang dapat menghasilkan padi, kedelai, jagung, umbi-umbian, dan lain-lainnya. Sedangkan, perkebunan di Indonesia seperti kebun dengan hasil berupa kopi, teh, lada, kelapa sawit, pala, dan lain-lain. Akan tetapi, sawah yang merupakan contoh pertanian dan perkebunan yang merupakan contoh kebun memiliki karakteristik yang sama sehingga sulit untuk dibedakan oleh masyarakat atau orang awam apabila dilihat dengan menggunakan foto udara atau foto yang diambil dari ketinggian tertentu. Untuk dapat mengenali dengan pasti bentuk sawah dan kebun

apabila dilihat dengan menggunakan foto udara atau foto menggunakan drone perlu dilakukan pembentukan model yang dapat mengenali secara akurat bentuk sawah dan bentuk kebun serta membantu pemerintah dalam melihat ketahanan pangan di Indonesia. Salah satu cara untuk membentuk model yaitu dengan memanfaatkan ilmu komputasi yang salah satu tujuannya adalah untuk mengambil informasi dari citra digital untuk mengenal *obyek* secara otomatis.

Pada saat ini kemajuan teknologi informasi tidak dapat dihindari. Dengan adanya perkembangan *hardware* dalam memajukan performa komputer serta berkembang pula *software* yang mampu menyerupai kecerdasan manusia (kecerdasan buatan). Zaman sekarang komputer berfungsi untuk dapat menyelesaikan pekerjaan dengan lebih mudah, lebih tangkas, dan dalam waktu yang singkat. Salah satunya teknologi yang dapat memudahkan manusia yaitu *deep learning*. *Deep learning* merupakan cabang ilmu dari *machine learning* yang berbasis jaringan saraf tiruan yang melatih atau mengajarkan suatu tindakan yang dianggap masuk akal bagi manusia. *Deep learning* mampu mengklasifikasikan secara otomatis dari gambar, teks, dan pada video dengan cara menjadikan video kedalam bentuk gambar. (Kusumaningrum,2018).

Salah satu metode *deep learning* yang sedang berkembang saat ini yaitu *Convolutional Neural Network* (CNN). Metode CNN merupakan metode yang masukkan (*input data*) berupa citra atau gambar. Metode ini memiliki lapisan khusus yang bernama lapisan konvolusi dimana pada lapisan sebuah citra masukkan (*input citra*) akan menghasilkan sebuah pola dari beberapa bagian citra yang nantinya akan lebih mudah untuk diklasifikasikan. Lapisan konvolusi memiliki fungsi pembelajaran citra menjadi lebih efisien untuk diimplementasikan. Oleh karena itu peneliti memanfaatkan metode CNN ini untuk dapat mengklasifikasi kebun dan sawah.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini yaitu :

1. Bagaimana implemetasi metode CNN terhadap pengklasifikasi citra kebun dan sawah?

2. Bagaimana hasil tingkat akurasi pengklasifikasi citra kebun dan sawah?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini yaitu:

1. Data yang digunakan yaitu citra hutan dan sawah yang diperoleh dari hasil pencarian di google.
2. Citra yang digunakan memiliki ukuran 32x32 piksel
3. Analisis pengolahan citra/gambar pada penelitian ini menggunakan *software* Rstudio versi 3.5.8 dengan bantuan *package* keras dan EBImage.

1.4 Tujuan Penelitian

Adapun tujuan pada penelitian ini yaitu:

1. Mengetahui implemetasi metode CNN terhadap pengklasifikasi citra kebun dan sawah.
2. Mengetahui hasil tingkat akurasi pengklasifikasi citra kebun dan sawah.

1.5 Manfaat Penelitian

Adapun tujuan pada penelitian ini yaitu:

1. Bagi Pembaca
Memberikan informasi tambahan mengenai metode *Convolutional Neural Network* (CNN)
2. Bagi Masyarakat dan Pemerintah
Dapat membedakan sawah dan kebun dari atas ketinggian dengan menggunakan alat seperti drone serta untuk membantu pemerintah dalam melihat ketahanan pangan di Indonesia
3. Penelitian Lanjutan
Dapat dijadikan acuan untuk penelitian selanjutnya.

BAB 2 TINJAUAN PUSTAKA

Berdasarkan penelitian yang akan pernah dilakukan, serta referensi dari beberapa penelitian terdahulu sangat penting dalam melakukan sebuah penelitian guna untuk mengetahui hubungan antara penelitian yang akan dilakukan dengan penelitian terdahulu, sehingga dengan menambahkan referensi tersebut dapat menghindari adanya suatu duplikasi atau penjiplakan dalam penelitian yang akan dilakukan. Berikut beberapa analisis mengenai penelitian terdahulu yang pernah dilakukan sebelumnya berkaitan dengan data serta metode yang digunakan.

Pada tahun 2016, terdapat penelitian berjudul “Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time” dilakukan oleh Muhammad Zufar dan Budi Setiyono. Dalam penelitian ini menggunakan metode *deep neural network* yaitu *Convolutional Neural Network* sebagai pengenalan wajah secara real-time yang sudah terbukti sangat efisien dalam klasifikasi wajah. Pada penelitian ini terdapat 6 dataset dengan 13 data serta 126 perincian setiap gambarnya. Hasil uji dengan menggunakan konstruksi model *Convolutional Neural Networks* sampai kedalaman lapisan menggunakan *input* dari hasil ekstrasi *Expanded Local Binary Pattern* dengan radius 1 dan *neighbor* 15 mendapatkan rata-rata tingkat akurasi lebih dari 89% dengan \pm frame per detik.

Pada tahun 2018, terdapat penelitian dengan judul “Implementasi metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi” dilakukan oleh Erlyna Nour Arrofiqoh dan Harintaka. Pada penelitian ini menggunakan algoritma CNN untuk membedakan jenis tanaman dengan memberikan label semantik dari objek jenis tanaman. Pada penelitian ini peneliti menggunakan 5 jenis tanaman, yaitu tanaman padi, bawang merah, kelapa, cabai, dan pisang. Data latih (*training*) yang digunakan sebesar 70% dengan banyak gambar sebesar 350 gambar dengan masing-masing tanaman sebanyak 70 sampel gambar. Data *test* (*testing*) yang digunakan 50 data gambar dengan masing-masing tanaman sebanyak 10 sampel gambar. Data validasi menggunakan 100 data untuk

menguji jaringan dengan masing-masing kelas sebanyak 20 sampel. Pada proses *learning* jaringan mendapatkan tingkat akurasi sebesar 100% terhadap data *training*. Pengujian untuk data validasi menghasilkan tingkat akurasi sebesar 93% dan tingkat akurasi terhadap data test sebesar 82%. Hasil penelitian menunjukkan bahwa penggunaan metode CNN mampu untuk pendekatan pengenalan objek secara otomatis dalam membedakan jenis tanaman sebagai bahan pertimbangan interprener dalam menentukan objek pada citra.

Pada tahun 2018, terdapat penelitian dengan judul “Implementasi *Deep Learning* Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network*” dilakukan oleh Sarirotul Ilahiyah dan Agung Nilogiri. Pada penelitian ini peneliti membagi dataset menjadi 10 folder yang terdiri dari 2000 citra yang telah dibagi menjadi data uji (*testing*) dan data latih (*training*) secara acak. Citra latih (*trainng*) pada setiap folder sebanyak 1800 citra, sedangkan untuk data uji (*testing*) sebanyak 200 citra. Hasil penelitian menggunakan *fold cross validation* dengan nilai $k = 10$ didapatkan hasil tertinggi pada percobaan pertama dari fold ke 7 yaitu dengan nilai tingkat akurasi sebesar 90%. Memiliki nilai rata-rata tingkat akurasi menggunakan *fold cross validation* dengan $k = 10$ yaitu sebesar 85,21%. Penelitian ini mampu mengklasifikasikan jenis tumbuhan dengan nilai akurasi sebesar 90.8%.

Pada tahun 2019, terdapat penelitian dengan judul “Klasifikasi Citra Multi-Kelas Menggunakan *Convolutional Neural Network*” dilakukan oleh Kamal Hasan Mahmud, Adiwijaya dan Said Al Faraby. Pada penelitian ini terdapat 1000 kelas dan akan diambil 100 kelas yang akan digunakan untuk daataset. Penelitian ini untuk mengetahui strategi pembelajaran CNN yang mampu dan memiliki performa terbaik dengan berbagai skenario. Skenario terbaik yang mampu didapatkan dengan melakukan traning dengan ukuran gambar kecil, lalu melakukan training kembali dengan memperbesar ukuran gambar. Pada skenario ini mendapatkan tingkat akurasi sebesar 75,82%. Skenario ini juga memiliki performa yang paling baik dalam klasifikasi keseluruhan berdasarkan ukuran evaluasi *confusion matrix*.

Pada tahun 2019, terdapat penelitian dengan judul “Klasifikasi Citra Buah Menggunakan *Convolutional Neural Network*” dilakukan oleh Febian Fitra Maulana dan Naim Rochmawati. Penelitian ini menggunakan metode *deep learning*

yang digunakan dalam *image processing* adalah *Convolutional Neural Network* (CNN). Dataset yang diolah adalah dataset citra buah-buahan yang berasal dari dataset Fruit-360. Kelas data yang digunakan yaitu 15 kelas dari 111 kelas pada dataset Fruit-360. Hasil dari proses learning didapatkan model CNN dengan tingkat akurasi 100% dan nilai loss sebesar 0,012, pada proses pengujian model CNN yang menggunakan 45 sampel citra buah didapatkan tingkat akurasi sebesar 91,42%

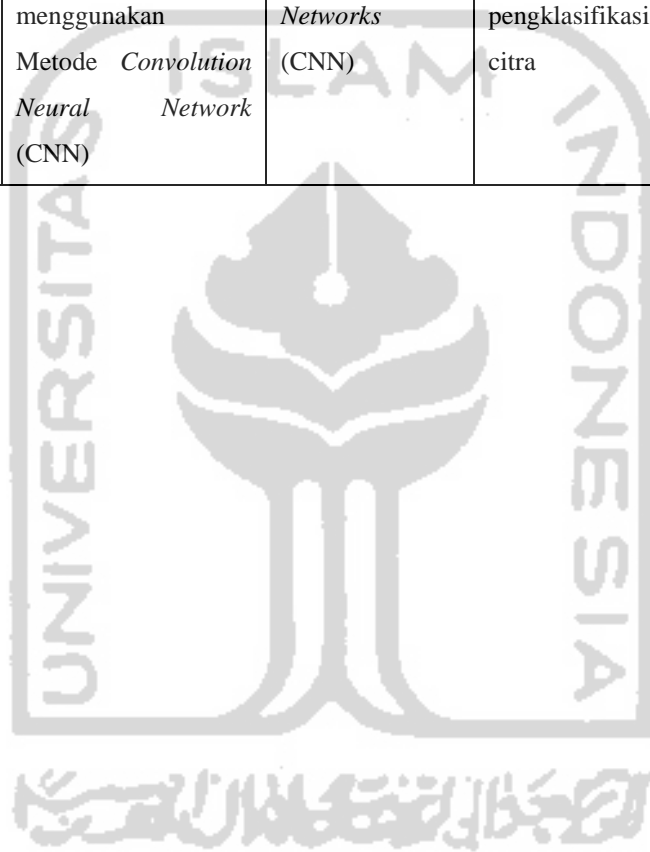
Pada tahun 2019, terdapat penelitian dengan judul “*Convolution Neural Network* (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional” dilakukan oleh Akhmad Rohim, Yuita Arum Sari, dan Tibyani. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk pengklasifikasian citra makanan tradisional. Jenis citra makanan terdiri dari 20 citra jenis makanan tradisional. Dengan jenis makan tradisional terdiri dari Wingko babat, Gethuk, Roti Bakar, Lumpia, Bakso Bakar, Lemper, Bugis, Ondeonde, Klepon, Tetel, Ketela Goreng, Kue Kucur, Putu Ayu, Molen, Bikang, Kue Sus, Pukis, serabi, Pisang Ijo, dan Risoles. Data citra dibagi atas 2 yaitu data latih (*training*) sebanyak 280 citra dan data *test* (*testing*) sebanyak 100 citra. Hasil penelitian ini menunjukkan dalam membangun arsitektur model *Convolutional Neural Network* (CNN) untuk pengklasifikasian citra makanan tradisional membutuhkan 4 *layer* konvolusi, 4 *layer max-pooling* dan 2 *layer fully connected*. Arsitektur tersebut didapatkan karena mendapatkan nilai *loss value* paling kecil dengan nilai loss sebesar 0.000044 pada *epoch* ke 15 saat proses pembelajaran dan mendapatkan nilai 73% presisi, 69% *recall* dan 69% *Fscore*.

Pada tahun 2019, terdapat penelitian dengan judul “Klasifikasi Citra Genus Panthera menggunakan Metode *Convolution Neural Network* (CNN)” dilakukan oleh Gusti Alfahmi Anwar, dan Desti Rimirasih. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk klasifikasi genus panther terdiri dari harimau, jaguar, macan tutul, dan singa. Dataset yang digunakan berupa citra/gambar harimau, jaguar, macan tutul, dan singa. Data latih (*training*) sebanyak 3840 citra, data validasi sebanyak 960 citra, dan data *test* (*testing*) sebanyak 800 citra. Hasil tingkat akursi untuk data *training* sebesar 92,31%, untuk data validasi yaitu sebesar 81,88%, dan untuk data *testing* mendapatkan hasil sebesar 68%. Hasil

akurasi prediksi didapatkan dari nilai F1-Score pada pengujian didapatkan untuk citra harimau sebesar 78%, untuk citra jaguar sebesar 70%, untuk citra macan sebesar 37%, dan untuk singa sebesar 74%.

No	Peneliti	Judul	Metode	Persamaan	Perbedaan
1.	Muhammad Zufar dan Budi Setiyono (2016)	<i>Convolutional Neural Networks</i> untuk Pengenalan Wajah Secara <i>Real-Time</i>	<i>Convolutional Neural Networks</i> (CNN)	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
2.	Erlyna Nour Arrofiqoh dan Harintaka (2018)	Implementasi metode <i>Convolutional Neural Network</i> Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi	<i>Convolutional Neural Networks</i> (CNN)	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
3.	Sarirotul Ilahiyah dan Agung Nilogiri (2018)	Implementasi <i>Deep Learning</i> Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan <i>Convolutional Neural Network</i>	<i>Convolutional Neural Networks</i> (CNN)	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
4.	Kamal Hasan Mahmud, Adiwijaya dan Said Al Faraby (2019)	Klasifikasi Citra Multi-Kelas Menggunakan <i>Convolutional Neural Network</i> "	<i>Convolutional Neural Networks</i> (CNN)	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
5.	Febian Fitra Maulana dan Naim Rochmawati (2019)	Klasifikasi Citra Buah Menggunakan <i>Convolutional Neural Network</i> "	<i>Convolutional Neural Networks</i> (CNN)	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
No	Peneliti	Judul	Metode	Persamaan	Perbedaan

6.	Akhmad Rohim, Yuita Arum Sari, dan Tibyani (2019)	<i>Convolution Neural Network (CNN)</i> Untuk Pengklasifikasian Citra Makanan Tradisional	<i>Convolutional Neural Networks (CNN)</i>	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda
7.	Gusti Alfahmi Anwar, dan Desti Riminarsih (2019)	Klasifikasi Citra Genus Panthera menggunakan Metode <i>Convolution Neural Network (CNN)</i>	<i>Convolutional Neural Networks (CNN)</i>	Melakukan penelitian tentang pengklasifikasi data citra	Objek yang digunakan berbeda



BAB 3 LANDASAN TEORI

3.1 Pengertian Sawah

Sawah menurut para ahli yaitu usaha pertanian yang dilakukan atau dilaksanakan pada suatu tanah basah yang memerlukan air untuk mengairi lahan pertanian atau irigasi. Jenis tanaman yang utama untuk pertanian pada sawah adalah padi. Dalam bersawah, pengolahan lahan dilakukan secara bersungguh-sungguh agar mendapatkan hasil yang optimal dan merupakan pertanian menetap. Sawah sangat bermanfaat bagi manusia yang berada di bumi karena tanpa sawah makan pagi dan sejenis bahan pangan tidak akan bisa makan, dimana diketahui bahwa padi yang menghasilkan beras merupakan makanan pokok masyarakat Indonesia.



Gambar 3.1. Sawah

3.2 Pengertian Kebun

Menurut Undang-undang Republik Indonesia pasal 1 poin 1 nomor 18 tahun 2004 tentang perkebunan, perkebunan adalah segala kegiatan yang mengusahakan tanaman tertentu pada tanah dan/atau media tumbuh lainnya dalam ekosistem yang sesuai, mengolah dan memasarkan barang dan jasa hasil tanaman tersebut, dengan bantuan ilmu pengetahuan dan teknologi, permodalan serta manajemen untuk mewujudkan kesejahteraan bagi pelaku usaha perkebunan dan masyarakat.

Kebun adalah lahan pertanian atau usaha tani yang sudah menetap, guna untuk ditanami tanaman tahunan secara permanen, baik sejenis maupun secara campuran. Tanaman yang biasa ditanam di lahan kebun yaitu seperti teh, kopi, cengkeh, karet, dan lain-lain.



Gambar 3.2. Kebun

3.3 Citra

3.3.1 Pengertian Citra

Citra dapat diartikan sebagai suatu fungsi dua dimensi $f(x,y)$, dimana x dan y merupakan koordinat (x,y) yang disebut intensitas. (Gonzales, Rafael, Wood, 2008). Citra digital adalah citra $f(x,y)$ yang telah digitalisasi atau memakai sistem digital baik dari segi koordinat area maupun pada nilai intensitasnya. Sebuah citra digital terdiri dari sejumlah elemen yang disebut *picture element* atau pixel.

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu citra yang memiliki sifat analog dan citra yang memiliki sifat digital. Citra analog yaitu citra yang berifat continue seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Sedangkan citra digital yaitu citra yang dapat diperoleh oleh komputer (Sutojo, 2009).

3.3.2 Pengertian Citra Digital

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun atas kolom dan baris. Dengan kata lain, sampling pada citra menyatakan kecil besarnya ukuran pixel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*greyscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra (Basuki, 2005).

Pada sebuah komputer, citra digital dipetakan menjadi elemen pixel berbentuk matriks dua dimensi dan bentuk *grid*. Setiap pixel memiliki angka yang menunjukkan *channel* warna. Angka pada setiap pixel disimpan secara berurutan oleh sebuah komputer serta sering dikurung untuk keperluan kompresi maupun

pengolahan tertentu. Pada citra digital dapat diwakili oleh sebuah matriks yang terdiri dari N baris dan M kolom, dimana perpotongan anatar baris dan kolom disebut dengan *pixel*. *Pixel* kependekan dari *picture element* yaitu elemen terkecil dari citra. *Pixel* memiliki dua parameter yaitu koordinat dan intensitas atau warna dari *pixel* pada titik itu. Oleh karena itu, citra dapat dituliskan kedalam sebuah matriks :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-0,1) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (3.1)$$

Berdasarkan rumuas diatas, suatu citra f(x,y) dapat dituliskan kedalam fungsi matematis seperti dibawah ini :

$$\begin{aligned} 0 \leq x \leq M - 1 \\ 0 \leq y \leq N - 1 \\ 0 \leq f(x,y) \leq G - 1 \end{aligned}$$

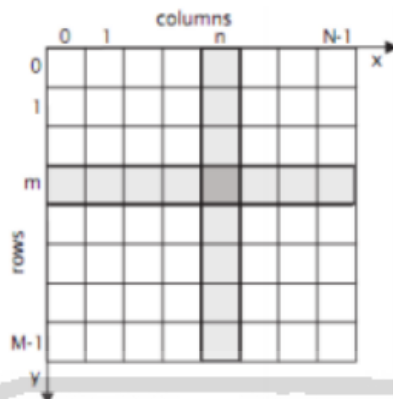
Dimana :

- M = Jumlah baris pada citra
- N = Jumlah kolom pada citra
- G = Nilai skala *grayscale*

Besarnya nilai M,N dan G merupakan perpangkatan dari dua seperti yang terlihat pada persamaan berikut :

$$M = 2^m ; N = 2^n ; G = 2^k \quad (3.2)$$

Dimana nilai m, n, dan k merupakan bilangan positif. Interval (0,G) disebut dengan keabuan (*greyscale*). Besarnya nilai g tergantung pada proses digitalisasinya. Biasanya keabuan 0 menyatakan berwarna hitam dan 1 menyatakan berwarna putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (drajat keabuan).



Gambar 3.3. Citra Digital Dua Dimensi

(Sumber : Tompunu, R. K, 2010)

3.3.3 Jenis-jenis Citra Digital

Banyak cara untuk menyimpan citra digital pada sebuah memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Beberapa jenis citra digital yang sering digunakan yaitu citra *biner*, citra warna, dan citra *grayscale* (Sutoyo, 2009).

1. Citra Biner atau monokrom. Banyaknya dua warna, yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini.
2. Citra *Grayscale* atau skala keabuan. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya.
3. Citra Warna atau *true color*. Setiap *pixel* pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar. Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap *pixel* mempunyai kombinasi warna sebanyak $28 \times 28 \times 28 = 224 = 16$ juta warna lebih.

3.3.4 Pengolahan Citra

Pengolahan citra adalah suatu proses pengolahan citra dengan memanfaatkan serta menggunakan komputer agar menjadi sebuah citra yang memiliki kualitas yang baik. Tujuan pengolahan citra yaitu untuk memperbaiki kualitas citra sehingga

mampu diinterpretasi atau dijelaskan dengan mudah oleh manusia atau semua mesin komputer.

3.4 *Artificial Intelligence (AI)*

Kecerdasan Buatan atau *Artificial Intelligence (AI)* adalah teknik atau metode yang diterapkan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup untuk menyelesaikan sebuah permasalahan. Kecerdasan buatan (*artificial intelligence*) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dilakukan lebih baik oleh manusia (Rich, 1991). *Artificial Intelligence (AI)* berfungsi untuk mengetahui atau memodelkan proses berpikir manusia dan mendesain mesin sehingga bisa meniru perilaku manusia (McCarthy, 1956). Mesin bisa berperan seperti manusia dengan dibekali pengetahuan serta kemampuan kelogisan pikir yang baik. Fungsi dari *Artificial Intelligence (AI)* yaitu untuk membuat robot yang dapat menyerupai kecerdasan atau bahkan lebih dari kecerdasan yang dimiliki oleh manusia.

Artificial Intelligence (AI) dibangun berdasarkan sistem yang memiliki keahlian serupa manusia pada ranah tertentu disebut dengan *soft computing*. *Soft computing* merupakan perubahan baru dalam membangun sistem cerdas yang mampu menyesuaikan dan bekerja lebih baik jika terjadi perubahan pada lingkungan. *Soft computing* juga mengusahakan adanya toleransi terhadap ketidakpastian, ketidaktepatan, dan kebenaran parsial sehingga dapat diselesaikan, dipecahkan dan dikendalikan dengan mudah agar sesuai dengan kenyataan.

3.5 *Machine Learning*

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. Pengertian sederhana machine learning adalah untuk membangun sebuah algoritma yang memungkinkan suatu program komputer untuk dapat belajar dan melakukan tugasnya sendiri tanpa adanya perintah dari penggunanya. Algoritma ini bekerja dengan cara membuat sebuah model dari masukkan (*input*) untuk dapat bisa menghasilkan suatu pengambilan keputusan berdasarkan data yang digunakan. *Machine learning* memiliki hubungan langsung dengan *computational statistics* yang berpusat pada

pembuatan keputusan yang berdasarkan pada penggunaan komputer. Beberapa penerapan dari *machine learning* yaitu *image processing*, *face*, *text analysis*.

Pada *machine learning* terdapat 3 komponen yang utama yaitu :

a. *Supervised Learning*

Data yang digunakan dilengkapi dengan label/kelas yang menunjukkan klasifikasi atau kelompok data berada. Model yang dihasilkan yaitu model prediksi dari data yang telah diberi kelas.

b. *Unsupervised Learning*

Data yang digunakan tidak memiliki label/kelas sehingga harus mencari struktur dari data yang digunakan, setelah itu melakukan pengelompokan berdasarkan informasi yang dimiliki.

c. *Reinforcement Learning*

Pembelajaran tentang apa yang akan dilakukan dan bagaimana memetakan situasi ke dalam aksi untuk dapat menghasilkan *reward* yang maksimal. Pembelajar tidak diberitahu aksi mana yang akan dipilih, tetapi lebih pada menemukan aksi mana yang dapat menghasilkan *reward* maksimal dengan cara mencoba menjalankannya.

3.6 *Deep Learning*

Deep learning adalah salah satu teknik pada *machine learning* yang memanfaatkan banyak *layer* pengolahan informasi *non-linear* untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi (Deng dan Yu, 2014).

Menurut (Goodfellow, dkk. 2016) *deep learning* adalah sebuah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer yang menggunakan konsep hierarki. Konsep hierarki membuat komputer mampu mempelajari konsep yang kompleks dengan menggabungkan dari konsep-konsep yang lebih sederhana. Jika digambarkan sebuah graf bagaimana konsep tersebut dibangun di atas konsep yang lain, graf ini akan dalam dengan banyak layer, hal tersebut menjadi alasan disebut sebagai *deep learning*.

Deep Learning merupakan salah satu bidang dari *Machine Learning* yang menggunakan jaringan syaraf tiruan untuk penerapan permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk

Supervised Learning. *Supervised Learning* yaitu tipe *learning* dimana termuat variabel input dan variabel output dengan menggunakan satu algoritma atau lebih untuk mempelajari fungsi pemetaan dari input ke output. Dengan menambahkan lebih banyak lapisan maka model pembelajaran bisa mewakili data citra berlabel dengan lebih baik.

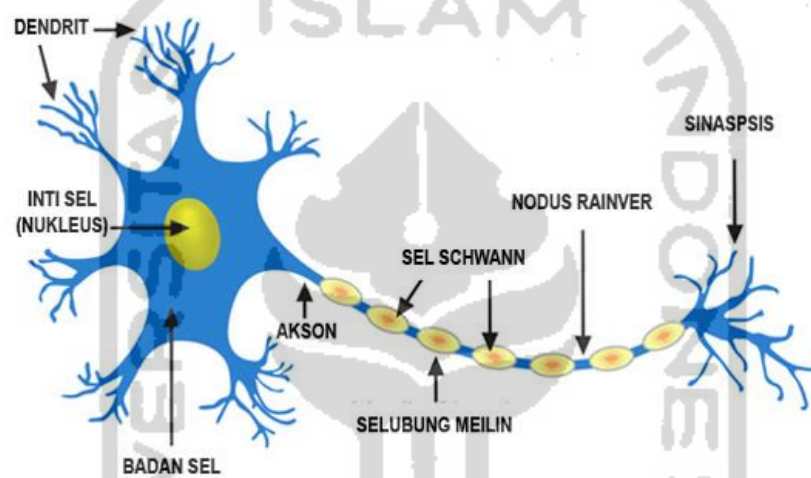
Penerapan konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditanggungkan pada algoritma *Machine Learning* yang telah ada sehingga komputer sekarang dapat belajar dengan akurasi, kecepatan, dan memiliki skala yang besar. Prinsip ini terus tumbuh hingga *Deep Learning* semakin sering dimanfaatkan pada komunitas riset dan industri untuk mendukung dan membantu memecahkan masalah pada data yang besar seperti, *Speech recognition*, *Natural Language Processing*, dan *Computer vision*.

Feature Engineering adalah salah satu karakteristik utama dari *Deep Learning* untuk mengekstrak pola yang berfungsi dan berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari, dianalisis, dan dikuasai karena kumpulan data dan jenis data yang berbeda membutuhkan pendekatan teknik yang berbeda juga.

Algoritma yang digunakan pada *Feature Engineering* dapat menemukan pola umum yang penting untuk membedakan antara kelas. Dalam *Deep Learning*, metode CNN atau *Convolutional Neural Network* sangatlah bagus dalam mendapatkan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *Deep Learning* penggunaan GPU (*Graphics Processing Unit*) sudah sangatlah umum (Danukusumo, 2017).

3.7 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) merupakan tiruan dari sistem jaringan syaraf biologi yang terdapat pada otak manusia. Dalam otak manusia telah tersusun **milyaran neuron yang saling berhubungan. Hubungan ini disebut dengan *Synapses*. Komponen neuron terdiri dari satu inti sel yang akan melakukan pemrosesan informasi, satu akson (*axon*) dan minimal satu dendrit. Informasi yang masuk akan diterima oleh dendrit. Selain itu, dendrit juga menyertasi akson sebagai keluaran dari suatu pemrosesan informasi.**



Gambar 3.4. Jaringan Saraf Manusia

Cara kerja dari sistem syaraf diatas adalah di mulai dari sinyal masuk melalui dendrit menuju ke *cell body*. Kemudian sinyal akan di proses didalam *cell body* berdasarkan fungsi tertentu (*Summation Proses*). Jika sinyal hasil proses melebihi nilai ambang batas (*Threshold*) tertentu maka sinyal tersebut akan membangkitkan neuron untuk meneruskan sinyal tersebut. Sedangkan jika hasil proses dibawah nilai ambang batasnya maka sinyal tersebut akan dihalangi (*Inhibited*). Kemudian sinyal yang diteruskan akan menuju ke *axon* dan terakhir menuju ke neuron lainnya melewati *synapse*.

ANN merupakan sistem adaptif (mudah menyesuaikan dengan keadaan) yang dapat mengubah strukturnya untuk memecahkan suatu masalah berdasarkan informasi internal maupun eksternal. Menurut Pham dalam jurnal Hermantoro (Pham, 1994) mengatakan bahwa ANN bersifat fleksibel terhadap inputan data dan menghasilkan *output* respon konsisten. ANN telah banyak digunakan dalam area

yang luas. Menurut Kumar & Haynes (Kumar, 2003) dalam jurnal Ulil Hamida (Hamida, 2014) menjelaskan, penerapan ANN dapat mengidentifikasi beberapa aplikasi yaitu:

- 1). Estimasi/prediksi (aproksimasi fungsi, peramalah)
- 2). Pengenalan Pola (klasifikasi, diagnosis, dan analisis diskriminan)
- 3). Klustering (pengelompokan tanpa adanya pengetahuan sebelumnya).

3.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. 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 [4]. 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 [1]. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi Image Net Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra. (E.P, I. W., Wijaya, A. Y., & Soelaiman, R. 2016)

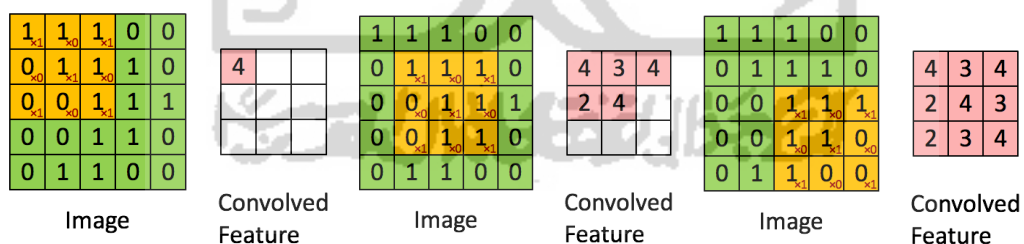
Convolutional Neural Network (CNN) adalah salah satu algoritma dari *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron (MLP)* yang dirancang untuk mengolah data dalam bentuk grid, salah satunya yaitu citra dua dimensi, misalnya gambar ataupun suara. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra atau gambar. Pada dasarnya klasifikasi citra dapat digunakan dengan MLP, tetapi dengan metode MLP kurang tepat untuk digunakan karena tidak

menyimpan informasi spasial dari data citra dan berpendapat untuk setiap piksel adalah fitur (karakteristik) yang independen sehingga menghasilkan hasil yang kurang baik.

Secara metode, CNN adalah sebuah arsitektur yang dapat dilatih dan memiliki beberapa tahapan. Masukan atau *input* dan keluaran atau *output* dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga *layer* yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*. Berikut penjelasan tentang arsitektur yang dimiliki oleh *Convolutional Neural Network* :

3.8.1 Convolution Layer

Convolution layer melakukan operasi konvolusi pada *output* dari lapisan sebelumnya. *Layer* adalah sebuah proses utama yang mendasar pada CNN. *Convolution layer* merupakan lapisan utama yang penting untuk digunakan. Konvolusi merupakan suatu istilah matematis yang dalam pengolahan citra berarti mengimplementasikan sebuah kernel (kotak berwarna kuning) pada citra disemua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar 3.5, sedangkan kotak berwarna hijau secara keseluruhan merupakan citra yang akan dikonvolusi. Kernel (kotak berwarna kuning) bergerak dari sudut kiri atas ke sudut kanan bawah. Sehingga hasil konvolusi dari citra dapat dilihat pada gambar disebelah kanannya.



Gambar 3.5. Proses *Convolution layer*

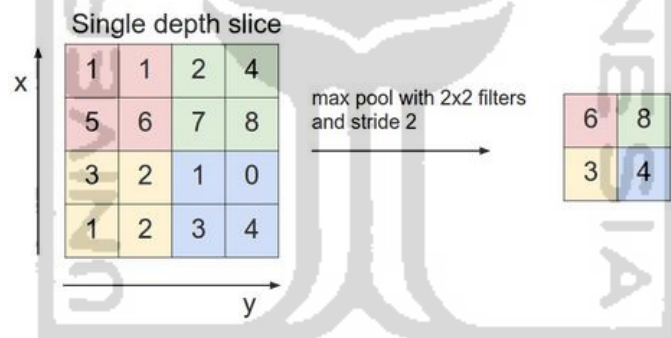
(Sumber : <https://www.cc.gatech.edu/~san37/post/dlhc-cnn/>)

Tujuan konvolusi pada data citra yaitu untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai dengan informasi spasial pada data. Bobot pada layer menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.

3.8.2 Pooling Layer

Pooling Layer merupakan lapisan yang menggunakan fungsi *feature map* sebagai masukan dan mengerjakannya dengan beraneka macam operasi statistik berdasarkan nilai piksel paling dekat. *Pooling layer* pada model CNN kebanyakan disisipkan secara teratur sesudah beberapa *convolution layer*. *Pooling layer* yang dimasukkan di antara lapisan konvolusi secara berurutan dalam arsitektur model *Convolutional Neural Network* dapat secara progresif mengurangi ukuran volume *output* pada *feature map*, sehingga jumlah parameter dan perhitungan di jaringan berkurang atau menurun, serta untuk mengendalikan *overfitting*.

Pooling layer digunakan untuk mengambil nilai maksimal (*max-pooling*) atau nilai rata-rata (*average pooling*) dari bagian-bagian piksel pada citra. Metode *pooling* yang paling sering digunakan dalam CNN yaitu metode *max-pooling*. *Max-pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil setelah itu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti pada Gambar 3.6.



Gambar 3.6. Proses *Pooling Layer* Pada Metode *Max Polling*

(Sumber : <http://cs231n.github.io/convolutional-networks/>)

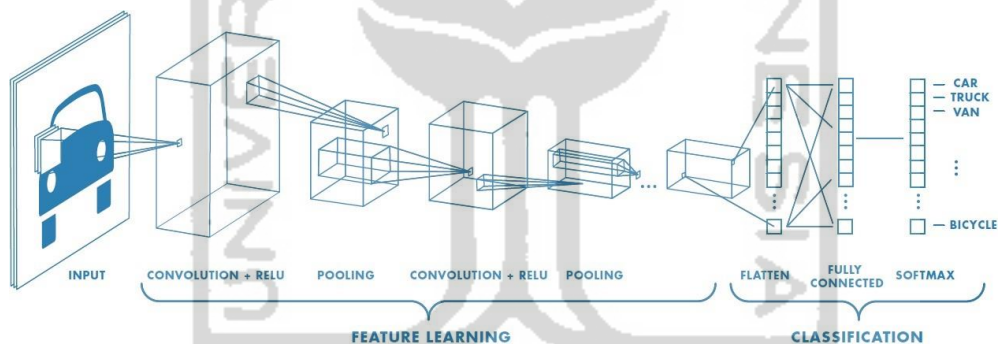
Pada gambar 3.4 kotak yang berwarna merah, kuning, hijau dan biru pada sisi kiri merupakan kelompok kotak yang akan dipilih nilai maksimum (Nilai Tertinggi). Sehingga hasil dari proses *Pooling Layer* Pada Metode *Max Polling* dapat dilihat pada kumpulan kotak disebelah kanan. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi atau pergeseran. Penggunaan *pooling layer* pada CNN memiliki tujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan. *Stride* merupakan parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai

stride adalah satu, maka filter akan bergeser sebanyak satu piksel secara horizontal kemudian vertikal. Semakin kecil *stride* yang digunakan, maka semakin detail informasi yang diperoleh dari sebuah input, akan tetapi membutuhkan komputasi yang lebih jika ingin dibandingkan dengan *stride* yang besar.

3.8.3 Fully Connected Layer

Fully-connected layer adalah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya sama halnya seperti jaringan saraf tiruan biasa. Pada setiap aktivasi dari lapisan sebelumnya harus diubah menjadi data satu dimensi sebelum dihubungkan pada semua *neuron* di *fully-connected layer*. Lapisan ini biasa digunakan pada metode *Multi-Layer Perceptron (MLP)* serta bertujuan untuk mengolah data sehingga dapat diklasifikasikan

Berikut adalah gambar jaringan arsitektur pada *Convolutional Neural Network* :



Gambar 3.7. Jaringan Arsitektur Pada *Convolutional Neural Network*
(Sumber : Medium Samuel Sena, 2017)

Berdasarkan gambar 3.7 diatas, Tahap pertama pada jaringan arsitektur CNN yaitu tahap konvolusi. Tahap ini dilakukan dengan memanfaatkan atau menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan dengan fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), setelah dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses *pooling* ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

3.9 Confusion Matrix

Confusion matrix adalah metode yang digunakan untuk mengevaluasi kinerja klasifikasi pohon keputusan (Tayefi, et al., 2017). *Confusion matrix* merupakan sebuah *table* yang terdiri dari banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi. Tabel ini diperlukan untuk mengukur kinerja suatu model klasifikasi (Swastina L, 2013).

Confusion matrix adalah alat yang berguna untuk menganalisis seberapa baik *classifier*/pengelompokan dapat mengenali tupel dari kelas yang berbeda (Han, Kamber, & Pei, 2012). Dalam pembuatan *table confusion matrix* ada empat hal yang harus diketahui, yaitu :

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
	FALSE	FN (<i>False Negative</i>)	TN (<i>True Negative</i>)

Gambar 3.8. *Confusion Matrix*

Keterangan:

1. *True Positive* (TP) : jumlah data set positif yang diklasifikasikan sebagai positif juga.
2. *True Negative* (TN) : jumlah data set negatif yang diklasifikasikan sebagai negatif juga.
3. *False Positive* (FP) : jumlah data set negatif yang diklasifikasikan sebagai positif.
4. *False Positive* (FP) : jumlah data set positif yang diklasifikasikan sebagai negatif.

3.10 Precision, Recall, dan Akurasi

a. Precision

Precision yaitu tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Untuk menghitung nilai *precision* menggunakan rumus sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (3.3)$$

b. Recall

Recall yaitu tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Untuk menghitung nilai *recall* menggunakan rumus sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (3.4)$$

c. Akurasi

Akurasi yaitu tingkat kedekatan antara nilai prediksi dengan nilai aktual. Akurasi dapat diartikan berupa kebenaran ataupun akurasi kesalahan. Untuk menghitung nilai Akurasi menggunakan rumus sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.5)$$



BAB 4 METODOLOGI PENELITIAN

4.1 Populasi dan Sampel Penelitian

Populasi dalam penelitian ini yaitu citra sawah dan citra kebun sebanyak 100 citra. Sampel yang digunakan dalam penelitian ini sebanyak 100 citra sawah dan citra kebun, setiap kategori (sawah dan kebun) terdiri atas 50 citra.

4.2 Variabel dan Definisi Variabel Penelitian

Berikut variabel yang digunakan dalam penelitian ini ditampilkan dalam Tabel 4.1 tentang penjelasan dan definisi variabel :

Tabel 4.1. Definisi Variabel Penelitian

Variabel	Definisi Variabel Penelitian
Sawah	Citra berupa Tanaman Padi
Kebun	Citra berupa Kebun

Data gambar dibagi kedalam 2 kelompok atau kategori yaitu data *train* dan data *test*.

4.3 Jenis dan Sumber Data

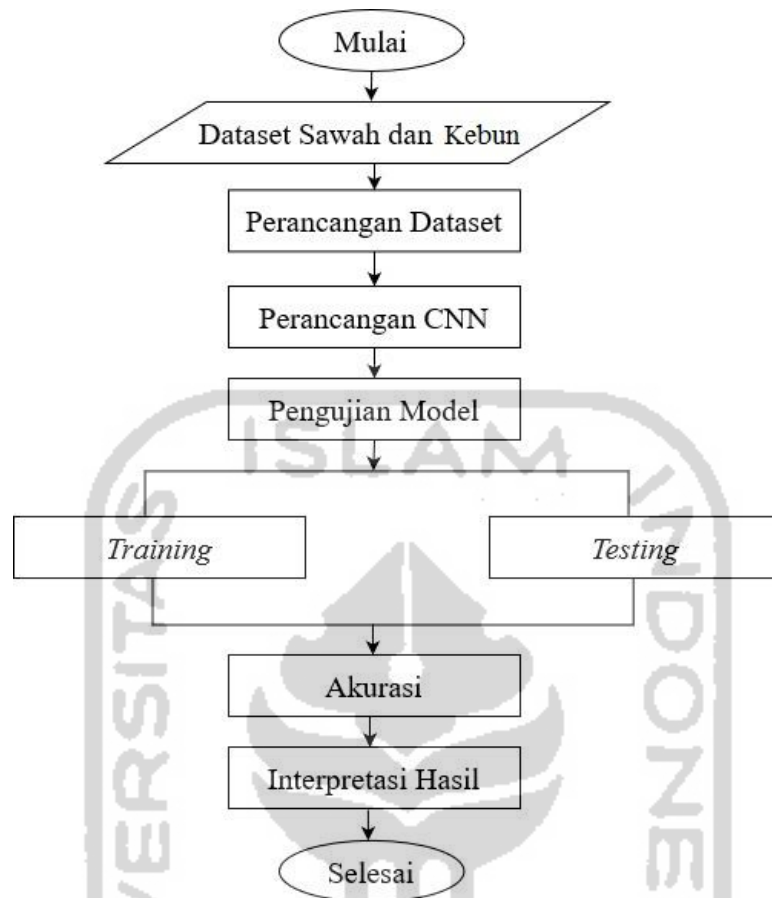
Jenis data yang digunakan pada penelitian ini adalah data sekunder. Data sekunder adalah data yang didapatkan atau diperoleh secara tidak langsung atau dari sumber yang sudah tersedia. Peneliti memperoleh data citra kebun dan sawah dengan *me-download* nya pada *search engine google images*.

4.4 Metode Analisis Data

Software yang digunakan pada penelitian ini adalah *software RStudio 3.5.8*. Metode analisis data yang digunakan dalam penelitian ini adalah metode *Convolutional Neural Network* yang bertujuan untuk mengklasifikasikan citra Sawah dan citra Kebun.

4.5 Tahapan Penelitian

Tahapan atau langkah-langkah dalam penelitian ini digambarkan melalui *flowchart* pada gambar 4.1 berikut:



Gambar 4.1. Tahapan Penelitian

4.6 Perangkat Pengujian

Perangkat pengujian dilakukan dengan laptop dengan spesifikasi sebagai berikut :

1. Inter core i3-6006U
2. 4 GB RAM
3. GPU : NVIDIA GeForce 940MX
4. Sistem operasi Windows
5. Bahasa Pemrograman R Studio

BAB 5 HASIL DAN PEMBAHASAN

Pada penelitian ini, peneliti melakukan klasifikasi Sawah dan Kebun dengan menggunakan algoritma *Convolutional Neural Network (CNN)*. Proses utama dalam pembuatan model ini diawali dengan *training data*. Pada proses ini bertujuan untuk pembentukan model yang akan digunakan pada pengujian data *testing*. Parameter untuk mengukur tingkat keberhasilan model adalah nilai tingkat akurasi. Nilai tingkat akurasi model dapat ditentukan dengan melakukan pengujian menggunakan data *testing*. Pada proses training menggunakan *packages Keras* dan *EImage* pada *software R Studio*.

5.1 Dataset Sawah dan Kebun

Data yang digunakan dalam penelitian ini merupakan data primer yang diperoleh dengan cara *men-download* satu persatu dari *search engine google*. Beberapa citra memiliki format gambar PNG, peneliti ingin mengubah format gambar ke format JPG dengan cara melalui situs <https://png2jpg.com/>. Alasan peneliti mengganti format PNG ke JPG agar format foto disamaratakan format JPG. Data yang digunakan oleh peneliti berupa data citra sawah sebanyak 50 citra dan citra kebun sebanyak 50 citra. Penggunaan data citra sebanyak 100 dikarenakan data tersebut telah cukup untuk melakukan penelitian mengenai citra sawah dan kebun dengan dua klasifikasi. Semakin banyak dataset yang digunakan mesin akan bisa lebih banyak belajar pada citra sebagai objek dimana mesin dapat lebih pintar dalam mengklasifikasikan sesuai jenis atau kategorinya sehingga hasil untuk tingkat akurasi semakin akurat. Dataset yang digunakan harus bervariasi atau diambil dari sudut pandang yang berbeda-beda agar algoritma CNN mampu mengenali objek dan belajar lebih banyak lagi mengenai objek yang diteliti.

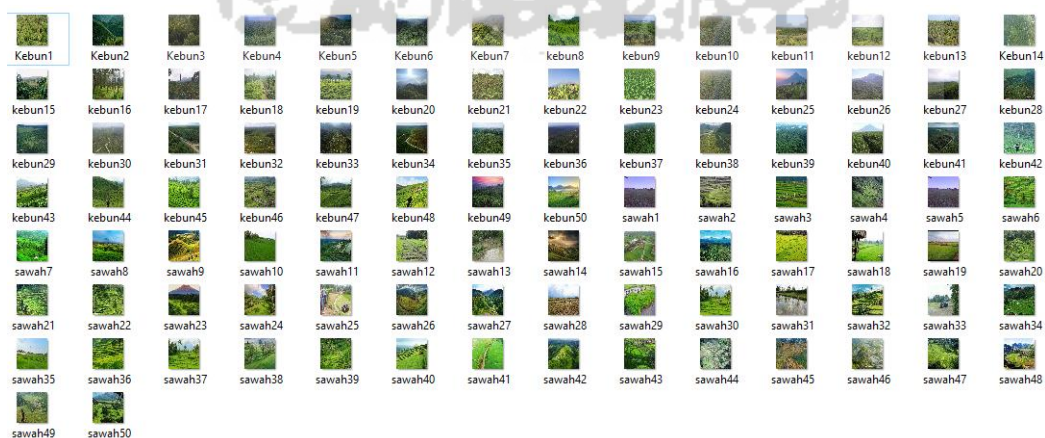
5.2 Preprocessing Dataset

Dari hasil pemilihan data dan pembuatan dataset, kemudian peneliti melakukan *pre-processing* dataset yakni *me-resize* data untuk menyamakan unsur dari gambar yang akan dianalisis dan memperjelas fitur gambar misalnya seperti data gambar harus memiliki kesamaan fitur baik itu ukuran, *channel* warna, dll.

Untuk melakukan *pre-processing* digunakan *package* EBImage yang tersedia pada *software* R, *package* ini akan memudahkan tahapan *pre-processing* yang dilakukan. Pada peneliti ini gambar di *resize* dari ukuran asli yang besar serta ukurannya berbeda-beda diubah menjadi sebesar 32x32 pixel dengan kualitas gambar 80%. Ukuran citra yang digunakan adalah nilai pangkat dua, nilai yang lebih kecil dari 32 adalah 16, sedangkan yang lebih besar dari 32 adalah 64, penggunaan dataset dengan ukuran 16x16 menjadikan informasi pixel banyak yang hilang, sedangkan penggunaan dataset dengan ukuran 64x64 menjadikan pemrosesan training menjadi lambat, sesuai dengan kemampuan PC atau Laptop yang digunakan untuk memproses algoritma CNN. Sehingga pada akhirnya peneliti menggunakan dataset dengan ukuran yang digunakan adalah sebesar 32x32 *pixel*. Ketika *input image* berbeda tidak ada pengaruh terhadap hasil klasifikasi. Untuk *me-resize* gambar digunakan *package* EBImage dengan perintah *resize* (gambar, w, h) dimana w adalah lebar dan h adalah tinggi yang ukurannya telah ditetapkan sebelumnya untuk *me-resize* gambar, baik menjadikan gambar lebih besar, ataupun lebih kecil.

5.3 Skenario Dataset Training dan Testing

Pada penelitian ini dataset citra dijadikan ke dalam satu folder yang bernama DataKebunSawah yang berisi 100 data citra. Kebun dan sawah diberi nama sesuai dengan nama katogorinya yakni kebun dan sawah, kemudian diikuti dengan nomor atau angka dimulai dari 1 sampai 50 yang menunjukkan barisan keberapa data tersebut, berikut gambaran dataset kebun dan sawah yang dibuat oleh peneliti



Gambar 5.1. Dataset Hutan dan Sawah

Pada gambar diatas menunjukkan data citra yakni 2 kategori kebun dan sawah yang berada dalam folder DataKebunSawah, karena penamaan data citra menggunakan nama kategorinya masing-masing, maka data citra akan memiliki letak yang otomatis berurutan sesuai dengan abjad dari nama kategori, yakni letak katogori ke-1 merupakan kebun dan kategori ke-2 merupakan sawah.

Dataset yang telah tersedia selanjutnya akan ditentukan jumlah datanya yang dibagi menjadi data *training* dan *testing*. Peneliti menentukan setiap kategori menggunakan data citra sebanyak 50 dimana dengan 3 perbandingan skenario data *training* yakni 90%,80%, dan 70%, untuk skenario 70% data *training* yang digunakan sebesar $(50 \times 70) : 100 = 35$ sehingga untuk data *testing* $50 - 35 = 15$, untuk skenario 80% data *training* yang digunakan sebesar $(50 \times 80) : 100 = 40$ sehingga data *testing* sebesar $50 - 40 = 10$, serta untuk skenario 90% data *training* yang digunakan sebesar $(50 \times 90) : 100 = 45$, maka data *testing* sebesar $50 - 45 = 5$ gambar. Berikut tabel pembagian data

Tabel 5.1. Skenario Pembagian Data

Skenario 70% : 30%		Skenario 80% : 20%		Skenario 90% : 10%	
Data <i>Training</i>	Data <i>Testing</i>	Data <i>Training</i>	Data <i>Testing</i>	Data <i>Training</i>	Data <i>Testing</i>
70	30	80	20	90	10

5.4 Perancangan CNN

Pada tahap ini peneliti akan melakukan beberapa percobaan untuk membandingkan model menggunakan arsitektur *Convolutional Neural Network* (CNN) untuk memperoleh hasil klasifikasi terbaik. Adapun tahapan untuk memperoleh arsitektur CNN yang terbaik dilihat dari parameter yang digunakan. Pada penelitian ini peneliti membanding nilai *epoch* dan membandingkan jumlah data *train* dan data *test*.

Untuk mengetahui perbedaan dan mencari arsitektur yang terbaik, hasil diperoleh dari perancangan arsitektur CNN akan peneliti jelaskan lebih rinci dibawah ini:

5.4.1 Membandingkan Nilai *Epoch*

Parameter yang akan dibandingkan untuk memperoleh arsitektur terbaik yaitu parameter *epoch*. *Epoch* adalah ketika seluruh dataset sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal untuk sekali putaran. Karena satu *epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer maka perlu membaginya kedalam satuan kecil (*batches*) (Digmi, 2018).

Epoch adalah ketika seluruh dataset sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal untuk sekali putaran, karena satu *Epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer maka dari itu perlu membaginya kedalam satuan kecil (*batches*). (Digmi,2018).

Nilai *epoch* yang akan dibandingkan pada percobaan ini yaitu *epoch* 50, 65, dan 80. Berdasarkan hasil *training* diperoleh hasil perbandingan nilai akurasi untuk setiap *epoch*. Berikut tabel perbandingan *epoch*:

Tabel 5.2. Perbandingan Nilai *Epoch*

Data	<i>Epoch</i> 50	<i>Epoch</i> 65	<i>Epoch</i> 80
	Akurasi	Akurasi	Akurasi
<i>Training</i>	0.775	0.9125	0.9625
<i>Los Training</i>	0.4750121	0.2588218	0.1201931
<i>Testing</i>	0.6	0.7	0.75
<i>Los Testing</i>	0.5834902	0.9526954	1.622669

Berikut hasil yang diperoleh dari perbandingan *epoch* 50, 65, dan 80 menunjukkan bahwa semakin besar *epoch* yang digunakan maka semakin baik hasil akurasi yang diperoleh. Tingkat akurasi paling tinggi yaitu *epoch* 80. Banyak *epoch* yang digunakan membuat algoritma terlatih lebih banyak lagi dan mampu mengenal *pattern* lebih baik lagi. Banyak *epoch* membuat model semakin baik.

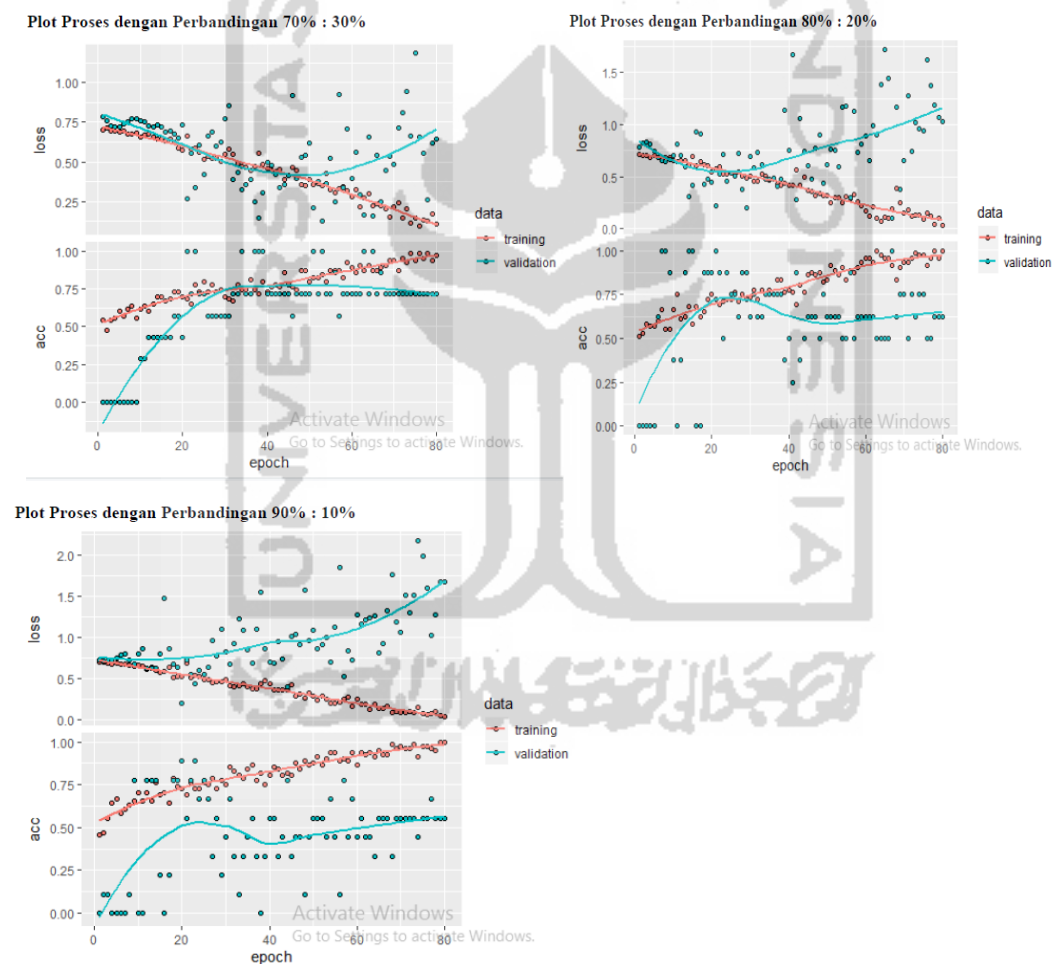
5.4.2 Membandingkan Jumlah Data *Train* dan Data *Test*

Setelah membandingkan nilai *epoch*, peneliti selanjutnya akan membandingkan jumlah data *train* dan data *test* 70%:30%, 80%:20%, dan 90%:10% dengan arsitektur yang optimal sebelumnya, hasil yang diperoleh setelah membandingkan jumlah data *train* dan data *test* disajikan dalam bentuk tabel dibawah.

Tabel 5.2. Perbandingan Jumlah Data *Train* dan Data *Test*

Data	70%:30%	80%:20%	90%:10%
<i>Training</i>	0.95714	0.9625	0.95556
<i>Loss Training</i>	0.12589	0.1201931	0.177167
<i>Testing</i>	0.6	0.75	0.7
<i>Loss Testing</i>	1.0396	1.622669	1.885138

Berdasarkan tabel 5.3 diatas, dapat diketahui bahwa nilai akurasi tertinggi diperoleh dengan menggunakan perbandingan 80%:20% dengan data *train* 80 dan data *test* 20. Hasil akurasi yang diperoleh yakni sebesar 0.9625 atau 96.25% untuk akurasi *training* dan sebesar 0.75 atau 75% untuk akurasi *testing*. Perbandingan ketiga hasil tersebut dapat dilihat prosesnya pada plot dibawah.

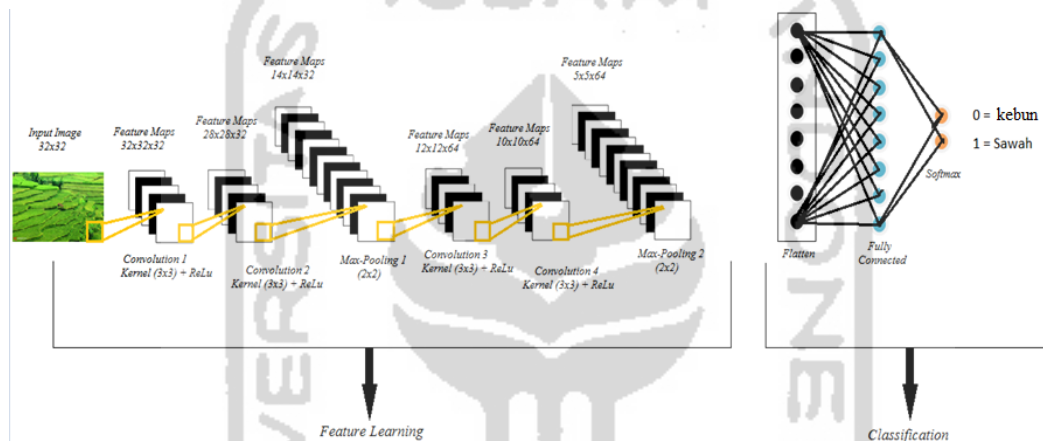


Gambar 5.2. Plot Proses Perbandingan Skenario Data *Train* 70%, 80%, dan 90%

Pada proses plot diatas terbagi menjadi 2 bagian, yakni bagian atas dan bagian bawah. Plot proses bagian atas merupakan plot proses *loss training* dan plot proses *loss testing*, sedangkan plot proses bagian bawah merupakan plot proses untuk nilai

akurasi *training* dan *testing*. Perbandingan ketiga plot diatas dapat dilihat dengan jelas perbedaannya pada nilai *loss* yang diperoleh, dengan menggunakan 90% nilai *loss* yang diperoleh lebih tinggi dibandingkan yang 70% dan 80%. Nilai *loss* yang diperoleh mencapai ± 1.8 . Nilai *loss* yaitu nilai kesalahan yang dibuat oleh model. Nilai *loss* yang terbaik yaitu nilainya semakin rendah atau kecil yang berarti bahwa hanya sedikit kesalahan yang diperoleh model saat mengklasifikasikan citra dengan tepat dan akurat.

5.5 Arsitektur CNN



Gambar 5.2. Arsitektur CNN

Berdasarkan gambar 5.3 diatas diketahui bahwa arsitektur CNN terbagi dalam dua tahapan yaitu *feature learning* dan *classification*. Berikut dibawah ini penjelasan dari kedua tahapan.

1. Tahapan pertama yaitu *feature learning*, terdiri atas:

a. *Convolutional layer*

Proses konvolusi memanfaatkan apa yang disebut dengan filter, seperti layaknya citra, filter memiliki ukuran lebar, tinggi, dan ketebalan sesuai dengan jumlah channel gambar. Konvolusi merupakan proses pertama yang dilakukan setelah menerima gambar yang di-*input* pada arsitektur. Konvolusi pertama berguna untuk memperhalus, menajamkan, dan mendeteksi tepi untuk pengolahan gambar/citra. Proses *convolution2D* (konvolusi) terjadi sebanyak 4 kali, dimana untuk *pooling* pertama terjadi sebanyak 2 kali konvolusi (*Convolution 1* dan *Convolution 2*) serta *pooling*

ke dua terjadi sebanyak 2 kali konvolusi (*Convolution 3* dan *Convolution 4*). Peneliti menggunakan ukuran kernel (3x3) dengan tujuan ukuran tersebut nantinya akan bergeser ke semua bagian *pixel* gambar/citra yang di *input* sebesar 32x32 maka dari itu, pergeseran tersebut dimaksud dengan operasi konvolusi. Kemudian filter yang digunakan sebanyak 32 guna untuk mengkonversi setiap filter ke seluruh bagian citra. Setelah proses konvolusi selesai akan dihasilkan sebuah *activation map* atau *feature maps*, dibantu dengan fungsi ReLu yang berguna mnegubah nilai negatif menjadi 0 dan nilai positif akan menjadi nilai aktivasi nya sendiri.

b. *Pooling Layer*

Pooling layer merupakan tahap setelah dilakukannya *convolutional layer*. Penelitian ini menggunakan *max-pooling* dengan ukuran 2x2 digunakan untuk menghitung *feature maps* dengan ukuran 16x16. Maka pada setiap pergeseran filter nilai yang diambil adalah nilai yang terbesar pada area 2x2, *max-pooling* bertujuan untuk mereduksi *size* citra sehingga mempercepat komputasi dan untuk mengatasi *overfitting*.

2. Tahapan kedua yaitu *classification*, yang terdiri dari:

a. *Flatten*

Setelah mendapatkan hasil *output* dari proses konvolusi yaitu *feature maps* yang berbentuk *multidimensional array*. Selanjutnya ke proses *flatten*, proses *flatten* berguna untuk menghasilkan sebuah vektor yang akan digunakan untuk *input* dari *Fully Connected Layer*.

b. *Fully Connected Layer*

Menggunakan metode *dropout* agar menonaktifkan sejumlah *edge* yang terhubung ke semua neuron agar tidak terjadi *overfitting*. Selanjutnya masuk ke tahap klasifikasi dimana, menggunakan aktivasi *softmax* agar yang diklasifikasi pada *input* bisa sesuai dengan kategori yang telah ditentukan. Terdiri 2 kategori yaitu 0 mewakili kebun dan 1 mewaikili sawah.

Setelah arsitektur CNN telah selesai dirancang maka didapatkan model yang terbaik, memiliki jumlah total parameter sebanyak 475.938 dengan rincian pada tabel 5.4 dibawah sebagai berikut:

Tabel 5.3. Model Terbaik

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
conv2d_1 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
dropout (Dropout)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_3 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 256)	409856
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 2)	514
Total params : 475.938		
Trainable params : 475.938		
Non-trainable params : 0		

Parameter yang dilatih pada model ini berjumlah 475.938 parameter. Setiap proses konvolusi akan membuat ukuran citra semakin kecil. Hingga ukuran terakhir yang digunakan sebelum masuk pada *fully-connected layer* yaitu 5x5 piksel dan 64 parameter. Selanjutnya dilakukan perubahan terhadap data matriks menjadi vektor agar dapat masuk kedalam proses *fully connected*, sehingga terdapat sebanyak 1600 neuron yang akan diteruskan. Kemudian jumlah neuron tersebut ditambahkan dengan proses *drop out*, dan kemudian dilakukan klasifikasi atas 2 kategori hutan dan sawah yang telah di-*training*.

Proses pertama kali yakni *input* gambar, dengan ukuran 32x32x3 merupakan ukuran panjang x lebar dari gambar. Konvolusi pertama menggunakan kernel 3x3

dengan filter 32 lapisan yang kemudian diaktivasi menggunakan aktivasi relu untuk memperoleh *nodes* paling besar untuk kemudian dilanjutkan pada proses konvolusi kedua, sehingga hasil konvolusi pertama memiliki *output* ukuran gambar sebesar $30 \times 30 \times 32$ dan parameter yang dihasilkan berjumlah $((3 \times 3 \times 3) + 1) \times 32 = 896$. Lalu proses konvolusi kedua, meneruskan data yang telah diproses pada proses sebelumnya, yakni ukuran gambar $32 \times 32 \times 3$ dengan menggunakan kernel 3×3 dan filter 32, maka diperoleh *output* ukuran gambar menjadi $28 \times 28 \times 32$ dan jumlah parameter sebesar $((3 \times 3 \times 32) + 1) \times 32 = 9248$, yang diperoleh dari perhitungan.

Setelah melakukan 2 kali konvolusi diawal, kemudian dilanjutkan pada proses *pooling*, dengan menggunakan jenis *pooling max* dan ukuran 2×2 maka untuk setiap 2 kotak piksel gambar akan direduksi menjadi 1 kotak piksel gambar, sehingga akan mengurangi ukuran pada gambar, yang pada awalnya berukuran 28×28 setelah di *pooling* berubah menjadi sebesar 14×14 . Pada proses ini parameter tidak perlu dihitung, oleh karena itu tidak terdapat jumlah parameter atau total parameter bernilai 0. Setelah itu dilanjutkan pada proses *dropout*, dengan menggunakan nilai batas atau *learning rate* sebesar 0,25 artinya untuk nilai piksel dibawah 0,25 akan dibuang atau tidak ikut digunakan pada proses selanjutnya.

Proses selanjutnya yaitu melakukan konvolusi ketiga, masih dengan ukuran kernel yang sama yakni 3×3 , namun filternya menjadi 64, dan *input* gambar berukuran 14×14 , dengan proses konvolusi yang sama, diperoleh *output* ukuran gambar menjadi 12×12 dan *output* parameter sebesar $((3 \times 3 \times 32) + 1) \times 64 = 18496$ parameter. Lalu dilanjutkan proses konvolusi ke-empat dengan *input* gambar berukuran 12×12 dan menggunakan kernel 3×3 serta filter sebesar 64 sehingga diperoleh hasil *output* ukuran gambar menjadi 10×10 dan *output* parameter sebesar $((3 \times 3 \times 64) + 1) \times 64 = 36928$.

Setelah proses konvolusi ke-empat selanjutnya kembali dilakukan *pooling*, masih dengan *max pooling* berukuran 2×2 dan proses *dropout* sehingga menghasilkan *output* ukuran gambar dari 10×10 menjadi 5×5 yang dimana gambar akan semakin mengecil. Kemudian masuk pada proses yang akan membentuk lapisan *flatten* untuk mengubah data menjadi vektor. *Output* lapisan *flatten* yakni sebesar $5 \times 5 \times 64 = 1600$. Selanjutnya diteruskan ke *layer dense* dengan jumlah unit

yang telah ditentukan yakni sebesar 256 units. *Output* yang diperoleh yakni sebesar $(1600 \times 256) + 256 = 409.856$ parameter.

Kemudian dilanjutkan pada *layer dense* kedua yakni dengan jumlah unit sebesar jumlah kelas yang akan diklasifikasikan, pada kasus ini terdapat 2 kategori gambar yaitu gambar sawah dan hutan maka ukuran unit yang digunakan yaitu sebanyak 2. Kemudian diperoleh *output* parameter sebesar $(256 \times 2) + 2 = 514$ parameter. Sehingga untuk total keseluruhan parameter yang diperoleh yakni sebesar 475.938 sama dengan jumlah parameter yang dilatih untuk *deep learning*.

5.6 Hasil Klasifikasi

Setelah mendapatkan arsitektur CNN terbaik, kemudian dilanjutkan dengan hasil klasifikasi. Menggunakan data uji sebanyak 20 citra dengan jumlah setiap kategori sebanyak 10 citra. Tabel hasil klasifikasi terbaik dari data *testing* akan peneliti sajikan dibawah ini:

Tabel 5.4. Hasil Klasifikasi Model Terbaik dengan Data *Testing* (Uji)

Prediksi	Aktual	
	Kebun	Sawah
Kebun	6	1
Sawah	4	9

Berdasarkan table 5.5 diatas, didapatkan hasil klasifikasi menggunakan data *testing*. Diperoleh jumlah citra kebun yang di prediksi sebagai citra kebun sebanyak 6 citra dan terdapat missing data yang tidak terprediksi sebagai citra kebun sebanyak 4 citra. Selanjutnya jumlah citra sawah yang di prediksi sebagai citra sawah sebanyak 9 citra dan terdapat *missing* data yang tidak terprediksi sebagai citra sawah sebanyak 1 citra. Perhitungan hasil klasifikasi dan nilai akurasi dapat dihitung dengan menggunakan rumus berikut:

$$\begin{aligned}
 \text{Testing Accuracy} &= \frac{\text{Total Prediksi benar Testing}}{\text{jumlah Total Testing}} \\
 &= \frac{15}{20} = 75\%
 \end{aligned}$$

Berdasarkan hasil perhitungan diatas diperoleh nilai akurasi berdasarkan hasil klasifikasi dengan arsitektur CNN terbaik menggunakan data *testing*/uji

sebesar 75% dimana pada hasil klasifikasinya citra kebun yang terprediksi sesuai citra kebun aktual nya hanya 6 citra dan untuk citra sawah aktual nya hanya 9 citra.



BAB 6 PENUTUP

6.1. Kesimpulan

Berdasarkan langkah-langkah analisis yang telah peneliti lakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Implementasi CNN untuk pengklasifikasian citra hutan dan sawah yaitu dengan meng-input citra gambar berukuran 32x32 pixel, skenario perbandingan dataset *train* sebesar 80% dan dataset *test* sebesar 20%, ukuran filter 32, ukuran kernel 3x3, ukuran *learning rate* 0.25 menggunakan aktivasi ReLu, jumlah *epoch* 80, batch size 32 dimana menggunakan jenis gambar berwarna (RGB).
2. Hasil dari akurasi *testing* yang didapatkan berdasarkan arsitektur terbaik untuk pengklasifikasian citra kebun dan citra sawah sesuai kategorinya sebesar 75%, dimana untuk hasil pengklasifikasian citra kebun sebanyak 6 citra yang mampu terdeteksi sesuai dengan kategorinya dari 10 objek citra yang diuji, untuk objek citra sawah sebanyak 9 citra yang mampu terdeteksi sesuai dengan kategorinya dari 10 objek citra yang diujikan.

6.2. Saran

Adapun saran dalam penelitian ini sebagai upaya perbaikan untuk penelitian selanjutnya sebagai berikut:

1. Menambahkan data citra kebun dan sawah lebih banyak lagi untuk memperoleh akurasi yang akurat.
2. Menambahkan beberapa parameter sebagai pembanding guna untuk mendapatkan aritektur yang lebih baik lagi.

DAFTAR PUSTAKA

- Akhmad Rohim, Y. A. (2019). Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* , 7037-7042 .
- Basuki, Ahmad, 2005, Pengolahan Citra Digital Menggunakan Visual Basic, Graha Ilmu: Yogyakarta.
- Badan Pusat Statistik (2019). Perkebunan. Retrieved Juni 28, 2020, from <https://www.bps.go.id/subject/54/perkebunan.html>
- Cs231n (2016). Proses Pooling layer pada Metode Max-pooling. Retrieved Maret 20, 2020, from <http://cs231n.github.io/convolutional-networks/>
- Danukusumo, K. (2017). *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU. Tugas Akhir.*
- Deng, L. dan Yu, D. 2014. *Deep Learning: Methods and Applications, Foundations and Trends in Signal Processing*. 7. 3-4. 197-387.
- Digmi, I. 2018. *Memahami Epoch, Batch size, dan Iteration*. Diperoleh pada 15 Maret 2020 dari : <https://imam.digmi.id/post/memahami-epoch-batch-size-dan-iteration/>
- E.P, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101. *Jurnal Teknik ITS Vol.5, No. 1.*
- Goodfellow, I., Bengio, Y., dan Courville, A.. 2016. *Deep Learning*. Diperoleh pada 19 Februari 2020 dari <http://goodfeli.github.io/dlbook/>
- Gusti Alfahmi Anwar, d. D. (2019). KLASIFIKASI CITRA GENUS PANTHERA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN). *Jurnal Ilmiah Informatika Komputer* .
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques. Third Edition*. United States of America: Elsevier.
- Harintaka, E. N. (2018). IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI TANAMAN PADA CITRA RESOLUSI TINGGI. *Geomatika Volume 24 No.2.*

- Indonesia, P. I. *Indonesia.co.id*. Retrieved maret 19, 2020, from <https://indonesia.go.id/ragam/pariwisata/pariwisata/sawah-berbentuk-jaring-laba-laba-di-flores>
- Indonesia, P. I. *Indonesia.co.id*. Retrieved maret 19, 2020, from <https://www.indonesia.go.id/narasi/indonesia-dalam-angka/ekonomi/usaha-berbasis-kayu-tetap-jalan-hutan-pun-tetap-lestari>
- Kamal Hasan Mahmud, A. d. (2019). Klasifikasi Citra Multi-Kelas Menggunakan Convolutional Neural Network. *e-Proceeding of Engineering : Vol.6, No.1 April*, 2127.
- Kusumaningrum, T. F. (2018). *IMPLEMENTASI CONVOLUTION NEURAL NETWORK (CNN) UNTUK KLASIFIKASI JAMUR KONSUMSI DI INDONESIA MENGGUNAKAN KERAS*. ((Studi Kasus : Jamur Kuping, Jamur Merang dan Jamur Tiram). Yogyakarta : SKRIPSI. UNIVERSITAS ISLAM INDONESIA
- Nilogiri, S. I. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem & Teknologi Informasi Indonesia)*, Vol. 3, No. 2.
- Rich, Elaine dan Knight, Kevin. 1991. *Artificial Intelligence*. McGraw-Hill Inc., New York.
- Rochmawati, F. F. (2019). Klasifikasi Citra Buah Menggunakan Convolutional Neural Network. *Journal of Informatics and Computer Science*.
- Sena, Samuel (2017). *Pengenalan Deep Learning Part 7 : Convolutional Neural Network*. Retrieved Maret 20, 2020, from <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>.
- Setiyono, M. Z. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *JURNAL SAINS DAN SENI ITS*.
- Sungtae's Awesome (2017). Proses *Convolution layer*. Retrieved Maret 20, 2020, from <https://www.cc.gatech.edu/~san37/post/dlhc-cnn/>
- Sutoyo, T, dkk. 2009, *Teori Pengolahan Citra Digital*, Penerbit Andi: Yogyakarta.

Tompunu, R. K. (2010). PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, gambar Representasi citra digital dalam 2 dimensi.

Woods, Richard E., Gonzales, Rafael E. 2008. *Digital Image Processing Third Edition*. Pearson Education. New Jersey.



LAMPIRAN

Lampiran 1 Program R Studio

```
# Load packages
library(keras)
library(EBImage)
library(tensorflow)

setwd('D:/Data222/')
save_in <- ("D:/Hasilll/")

# Load images names
images <- list.files()

# Main loop resize images
w <- 32
h <- 32
for(i in 1:length(images))
{
  # Try-catch is necessary since some images
  # may not work.
  result <- tryCatch({
    # Image name
    imgname <- images[i]
    # Read image
    img <- readImage(imgname)
    # Path to file
    img_resized <- resize(img, w = w, h = h)
    path <- paste(save_in, imgname, sep = "")
    # Save image
    writeImage(img_resized, path, quality = 80)
    # Print status
    print(paste("Done",i,sep = " ")),
    # Error function
    error = function(e){print(e)}
  }
}
```

```

#Analisis CNN
# Read Images
setwd('D:/Hasilll/')
images <- list.files()
images
summary(images)
list_of_images = lapply(images, readImage)
head(list_of_images)
display(list_of_images[[14]])

#create train
train <- list_of_images[c(1:40, 51:90)]
str(train)
display(train[[27]])

#create test
test <- list_of_images[c(41:50, 91:100)]
test
display(test[[1]])
par(mfrow = c(10,10))
for (i in 1:80) plot(train[[i]])

# Resize & combine
str(train)
for (i in 1:80) {train[[i]] <- resize(train[[i]], 32, 32)}
for (i in 1:20) {test[[i]] <- resize(test[[i]], 32, 32)}
for (f in 1:80) {print(dim(train[[f]]))}
train <- combine(train)
str(train)
x <- tile(train, 5)
display(x, title='Pictures')
test <- combine(test)

y <- tile(test, 2)
display(y, title = 'Pics')

```

```

# Reorder dimension
train <- aperm(train, c(4,1,2,3))
test <- aperm(test, c(4,1,2,3))
str(train)
str(test)

# Response
trainy <- c(rep(0,40),rep(1,40))
testy <- c(rep(0,10),rep(1,10))

# One hot encoding
trainLabels <- to_categorical(trainy)
trainLabels
testLabels <- to_categorical(testy)
testLabels

# Model
model <- keras_model_sequential()
model %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu',
                input_shape = c(32, 32, 3)) %>%
  layer_conv_2d(filters = 32,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.25) %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_conv_2d(filters = 64,
                kernel_size = c(3,3),
                activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2,2)) %>%
  layer_dropout(rate = 0.25) %>%

```



```

layer_flatten() %>%
layer_dense(units = 256, activation = 'relu') %>%
layer_dropout(rate=0.25) %>%
layer_dense(units = 2, activation = 'softmax') %>%
compile(loss = 'categorical_crossentropy',
        optimizer = optimizer_sgd(lr = 0.01,
                                   decay = 1e-6,
                                   momentum = 0.9,
                                   nesterov = T),
        metrics = c('accuracy'))
summary(model)
# Fit model
history <- model %>%
  fit(train,
      trainLabels,
      epochs = 80,
      batch_size = 32,
      validation_split = 0.1)
plot(history)

# Evaluation & Prediction - train data
model %>% evaluate(train, trainLabels)
pred <- model %>% predict_classes(train)
pred
table(Predicted
      = pred, Actual = trainy)
prob <- model %>% predict_proba(train)
cbind(prob, Predicted_class = pred, Actual = trainy)

# Evaluation & Prediction - test data
model %>% evaluate(test, testLabels)
pred <- model %>% predict_classes(test)
pred
table(Predicted = pred, Actual = testy)
prob <- model %>% predict_proba(test)
cbind(prob, Predicted_class = pred, Actual = testy)

```