

CONVOLUTION NEURAL NETWORK (CNN) UNTUK PENGKLASIFIKASIAN CITRA MAKANAN TRADISIONAL

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Akhmad Rohim
NIM: 155150200111017



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

CONVOLUTION NEURAL NETWORK (CNN) UNTUK PENGKLASIFIKASIAN CITRA MAKANAN TRADISIONAL

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:
Akhmad Rohim
NIM: 155150200111017

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Juli 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Yuita Arum Sari, S.Kom., M.Kom.
NIK: 201609880715 2 001

Pembimbing II

Tibyani, S.T., M.T.
NIP: 19691101 199512 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Juli 2019



Akhmad Rohim

NIM: 155150200111017

PRAKATA

Puji syukur kehadiran Allah SWT karena atas berkah, rahmat dan kaunia-Nya penulis dapat menyelesaikan skripsi dengan judul “*Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional*”. Skripsi yang ditulis merupakan tugas akhir dalam memenuhi persyaratan memperoleh gelar sarjana Komputer pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Selama pengerjaan skripsi ini penulis mendapatkan banyak bantuan, dukungan dan doa dari berbagai macam pihak. Pada kesempatan ini penulis bermaksud menyampaikan ucapan terimakasih kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Achmad Arwan, S.Kom, M.Kom selaku dosen penasihat akademik yang telah memberikan saran dan nasihat selama menempuh masa studi.
5. Ibu Yuita Arum Sari, S.Kom., M.Kom. selaku dosen pembimbing I yang telah memberikan bimbingan, pengarahan, dan saran kepada penulis selama masa pengerjaan skripsi.
6. Bapak Tibyani, S.T., M.T. selaku dosen pembimbing II yang telah memberikan bimbingan, dan ilmu kepada penulis selama masa pengerjaan skripsi.
7. Seluruh Dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu dan pengetahuannya kepada penulis selama masa studi hingga skripsi ini selesai.
8. Seluruh Civitas Akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan bantuan kepada penulis selama masa studi hingga skripsi ini selesai.
9. Ibu Cholifah, orang tua penulis yang senantiasa mendukung, memberikan motivasi dan doa serta pengorbanannya baik dari segi moril, materi kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
10. Abdulloh, Abdul Halim dan Siti Fatimah kakak penulis yang memberikan dukungan dan selalu siap membantu penulis dalam menyelesaikan skripsi ini.
11. Seluruh keluarga yang tidak dapat peneliti sebutkan satu persatu yang telah senantiasa mendoakan dan mendukung peneliti untuk menyelesaikan skripsi ini dengan baik
12. Teman-temanku yakni Ahmad Fauzi Ahsani, Vicky Viridus, Cindy Inka Sari, Nabila Lubna Irbakanisa, dan teman lainnya yang selalu mensupport dan menjadi tempat berkeluh kesah, teman diskusi, dan sumber pencerahan selama penyusunan skripsi ini.
13. Keluarga Alumni TIF-B 2015 yang telah membantu, memberikan kritik, dan saran kepada penulis.

14. Teman-teman keminatan Komputasi Cerdas yang sudah bersedia meluangkan waktu untuk berdiskusi dan memberikan dukungan dalam proses pengerjaan skripsi ini.
15. Teman-teman seperjuangan Teknik Informatika angkatan 2015 terimakasih untuk kebersamaan dan kerukunan selama empat tahun ini.
16. Seluruh pihak yang tidak dapat peneliti sebutkan satu persatu telah memberikan bantuan kepada peneliti hingga penyelesaian skripsi ini.

Penulis menyadari masih terdapat kekurangan dari skripsi ini, karena itu penulis mengharapkan kritik dan saran guna dapat menyempurnakan skripsi ini. Harapannya semoga skripsi ini dapat memberikan manfaat bagi pembaca, peneliti lainnya dan pihak lainnya.

Malang, 18 Juli 2019

Penulis
akhmad_rohim@student.ub.ac.id



ABSTRAK

Akhmad Rohim, *Convolutional Neural Network* untuk Pengklasifikasian Citra Makanan Tradisional.

Pembimbing: Yuita Arum Sari, S.Kom., M.Kom. dan Tibyani, S.T., M.T.

Masyarakat dalam era digital masa kini memfoto sebelum makan merupakan salah satu budaya gaya hidup. Kemudian hasil foto yang didapat akan diunggah ke media sosial. Penyebaran foto makanan tradisional yang masih kurang dalam mengidentifikasi mendorong penelitian ini untuk melakukan penelitian tentang klasifikasi citra makanan tradisional. Pada ekstraksi fitur klasifikasi citra makanan merupakan hal yang sulit karena makanan bisa secara dramatis bervariasi dalam penampilan seperti bentuk, tekstur, warna, dan sifat visual lainnya. *Convolution Neural Network* (CNN) merupakan metode yang dapat mempelajari sendiri fitur pada citra yang kompleks. Diharapkan hasil evaluasi CNN untuk pengklasifikasian citra makanan tradisional dapat memberikan solusi untuk mengidentifikasi citra makanan tradisional. Hasil penelitian ini menunjukkan dalam membangun arsitektur model *Convolutional Neural Network* untuk pengklasifikasian citra makanan tradisional membutuhkan 4 layer konvolusional, 4 layer *maxpooling* dan 2 layer *Fully connected*. Arsitektur tersebut didapatkan karena mendapatkan nilai *loss value* terkecil dengan nilai 0.000044 pada *epoch* ke 15 saat proses pembelajaran dan mendapatkan nilai 73% presisi, 69% recall dan 69% *F-score*.

Kata Kunci: *Convolutional Neural Network*, CNN, Makanan, Makanan Tradisional, *Confusion Matrix*.

ABSTRACT

Akhmad Rohim, *Convolutional Neural Network for Classification of Traditional Food Images.*

Adviser: Yuita Arum Sari, S.Kom., M.Kom. dan Tibyani, S.T., M.T.

People in this digital era take a picture before eating is one of lifestyle. Then the result of the picture will be uploaded to social media. Traditional food's pictures dissemination still less identified encourages this research about the classification of traditional food's image. Extraction of classification features food image is difficult because of food can vary dramatically in appearances such as shape, texture, color, and other visual properties. Convolution Neural Network (CNN) is a method that can learn its own features on a complex image. Hopefully, CNN evaluation results for the classification image of traditional food can provide a solution to identify the image of traditional food. Result of this research in building the architecture of the Convolutional Neural Network model for classification of the traditional food image required 4 conditional layers, 4 max-pooling layers, and 2 fully connected layers. That architecture obtained because it gets the smallest loss value with 0.000044 value on the 15 epoch during the learning process and gets a 73% precision, 69% recall, and 69% F-score.

Keywords: *Convolutional Neural Network, CNN, Food, Traditional Food, Confusion Matrix.*

Daftar Isi

PERSETUJUAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
Daftar Isi	viii
Daftar Gambar.....	x
Daftar Tabel.....	xi
Daftar Lampiran	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka.....	4
2.2 Dasar Teori.....	7
2.2.1 Citra Digital	7
2.2.2 Operasi Konvolusi	8
2.2.3 <i>Neural Network</i>	9
2.2.4 <i>Convolutional Neural Network (CNN)</i>	11
2.2.5 <i>Confusion Matrix</i>	14
BAB 3 METODOLOGI PENELITIAN	17
3.1 Tipe Penelitian	17
3.2 Strategi Penelitian	17
3.2.1 Lokasi Penelitian	18
3.2.2 Teknik Pengumpulan Data	18
3.2.3 Peralatan Pendukung.....	23
BAB 4 PERANCANGAN.....	24
4.1 Perancangan Proses CNN	24



4.1.1 Augmentasi Data.....	25
4.1.2 Inisialisasi Model.....	25
4.1.3 Pelatihan Model.....	27
4.2 Perhitungan Manualisasi	27
4.3 Skenario Pengujian	52
4.3.1 Skenario pengujian <i>Loss</i> dan Akurasi CNN	52
4.3.2 Skenario Pengujian <i>Confusion Matrix</i>	53
BAB 5 Implementasi	54
5.1 Lingkungan Implementasi	54
5.1.1 Lingkungan Perangkat Keras.....	54
5.1.2 Lingkungan Perangkat Lunak	54
5.2 Implementasi Algoritme <i>library</i>	55
5.2.1 Implementasi Augmentasi Data	55
5.2.2 Implementasi Penentuan Arsitektur CNN	56
5.2.3 Implementasi proses pembelajaran CNN	56
BAB 6 Pengujian Dan Analisis.....	58
6.1 Pengujian <i>Loss</i> dan Akurasi pada Data Uji.....	58
6.2 Pengujian <i>Confusion Matrix</i>	61
BAB 7 Penutup	68
7.1 Kesimpulan	68
7.2 Saran.....	68
Daftar pustaka.....	69

Daftar Gambar

Gambar 2.1 Citra Digital	8
Gambar 2.2 Filter citra menggunakan Konvolusi	9
Gambar 2.3 Operasi Konvolusi Matriks	9
Gambar 2.4 Struktur neuron pada otak manusia	10
Gambar 2.5 Struktur Neural Network.....	10
Gambar 2.6 Multi Layer Neural Network.....	11
Gambar 2.7 Arsitektur CNN	12
Gambar 2.8 Layer Konvolusi	13
Gambar 2.9 Max Pooling.....	13
Gambar 3.1 Diagram Alir Penelitian.....	18
Gambar 3.2 Contoh Pengambilan Citra 2	19
Gambar 3.3 Contoh Pengambilan Citra 1	19
Gambar 4.1 Diagram Alir Sistem Secara Umum	24
Gambar 4.2 Perancangan model.....	26
Gambar 4.3 Contoh Citra Perhitungan Manual	27
Gambar 4.4 Contoh Citra Bentuk Angka	27
Gambar 6.1 Grafik perekaman <i>loss</i>	59
Gambar 6.2 Grafik Perekaman Akurasi.....	60
Gambar 6.3 Grafik Hasil Rata-rata Pengujian	67

Daftar Tabel

Tabel 2.1 Kajian Pustaka	5
Tabel 2.2 Hyperparameter pada Convolutional Layer	12
Tabel 2.3 Istilah Confusion Matrix	14
Tabel 3.1 Data Citra	19
Tabel 3.2 Perangkat Keras Pendukung	23
Tabel 3.3 Perangkat Lunak Pendukung	23
Tabel 4.1 Perancangan Kedalaman Model	26
Tabel 4.2 Manualisasi Inisialisasi filter	28
Tabel 4.3 Manualisasi Inisialisasi bobot	28
Tabel 4.4 Tabel Skenario Pengujian <i>Loss</i> dan Akurasi	52
Tabel 4.5 Skenario Pengujian Confusion Matrix	53
Tabel 5.1 Lingkungan Perangkat Keras	54
Tabel 5.2 Lingkungan Perangkat Lunak	54
Tabel 5.3 Implementasi Augmentasi Data	55
Tabel 5.4 Kode Program Arsitektur	56
Tabel 5.5 Kode Program Pelatihan	57
Tabel 6.1 Model CNN	58
Tabel 6.2 Hasil Perekaman Nilai <i>Loss</i>	58
Tabel 6.3 Hasil Perekaman Nilai <i>Loss</i> (Lanjutan)	59
Tabel 6.4 Hasil Perekaman Akurasi Data Latih	60
Tabel 6.5 Pengujian Model 4	61
Tabel 6.6 Hasil Prediksi Makanan 3 pada Model 4	62
Tabel 6.7 Hasil Prediksi Model 4 Terhadap Makanan 3	63
Tabel 6.8 Hasil Pengujian Model 5	64
Tabel 6.9 Hasil Pengujian Model 6	65
Tabel 6.10 Hasil Pengujian Model 7	66

Daftar Lampiran

LAMPIRAN A HASIL KELUARAN PREDIKSI MODEL 4	70
LAMPIRAN B HASIL KELUARAN PREDIKSI MODEL 5.....	72
LAMPIRAN C HASIL KELUARAN PREDIKSI MODEL 6.....	74
LAMPIRAN D HASIL KELUARAN PREDIKSI MODEL 7	76



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Makanan mencerminkan karakteristik suatu suku bangsa (Ferguson, 2010). Makanan bisa merepresentasikan suatu negara, dan menjadi simbol penyebaran budayanya. Berbagai negara mengupayakan penyebaran kulinernya secara masif dan ekstensif ke berbagai negara lainnya. Makanan khas atau makanan tradisional telah dianggap sebagai duta promosi dan bentuk pendekatan kultural yang efektif. Banyak orang Indonesia yang belum pernah ke Jepang, tapi sudah terbiasa mengonsumsi sushi dan ramen. Termasuk Italia, banyak orang Indonesia yang gemar menyantap pizza atau pasta.

Di masyarakat yang sudah mapan, kebutuhan untuk memuaskan rasa lapar adalah sebuah gaya hidup (Feist & Handriatno, 2010). Memfoto sebelum Makan merupakan salah satu budaya gaya hidup di era digital masa kini. Kemudian hasil foto yang didapat akan diunggah ke media sosial. Dengan adanya budaya seperti itu banyak foto makanan tradisional yang tersebar di media sosial, namun terkadang kita lupa bahkan tidak mengetahui nama makanan tersebut. Maka penyebaran memperkenalkan makanan tradisional akan sedikit terhambat. Kurangnya identifikasi citra makanan daerah asal terhadap kuliner Indonesia adalah hal buruk. Padahal, Indonesia memiliki berbagai macam makanan tradisional yang khas dari setiap daerahnya.

Makanan sebagai kebutuhan manusia merupakan cara yang tepat untuk memperkenalkan Indonesia ke dunia. Penyebaran foto makanan tradisional yang masih kurang dalam mengidentifikasi mendorong penelitian ini untuk melakukan penelitian tentang klasifikasi citra makanan tradisional. klasifikasi dalam penelitian ini menggolongkan citra makanan untuk mengidentifikasi makanan yang ingin diketahui, berdasarkan ciri khas setiap citra.

Pada ekstraksi fitur klasifikasi citra makanan merupakan hal yang sangat sulit karena makanan bisa secara dramatis bervariasi dalam penampilan seperti bentuk, tekstur, warna, dan sifat visual lainnya (He, Xu, Khanna, Boushey, & Delp, 2013). Berbagai metode klasifikasi punya cara masing-masing dalam mengekstraksi fitur. Proses ekstraksi fitur yang digunakan pada umumnya memiliki cara sangat terbatas dimana hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi pada jenis citra apapun. Hal tersebut dikarenakan berbagai perbedaan antar citra antara lain perbedaan sudut pandang, perbedaan skala, perbedaan kondisi pencahayaan, deformasi objek, dan sebagainya. (Kumari & Saxena, 2018) mengulas beberapa hasil penelitian mengenai segmentasi dan klasifikasi tumor otak. Dalam ulasannya Kumari mendeskripsikan kelebihan setiap penelitian dan memberikan komentar untuk penelitian lanjut, bahkan Kumari dapat menyimpulkan kelebihan dan kekurangan setiap metode yang digunakan untuk deteksi tumor otak. *Convolution Neural Network* (CNN) merupakan metode yang dapat mempelajari sendiri fitur pada citra yang kompleks (Kumari & Saxena, 2018).

Pada tahun 1989, Yann LeCun dan teman-temannya berhasil melakukan klasifikasi citra kode zip menggunakan kasus khusus dari *Feed Forward Neural Network* dengan nama *Convolution Neural Network* (CNN). Metode *Deep Learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia sehingga memiliki kemampuan mengolah informasi citra. Pengenalan citra berdasarkan ekstraksi fitur yang dapat berkembang berdasarkan pelatihan, pembelajaran, dan pemahaman (Aggarwal & Gagandeep, 2018).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, dapat dirumuskan masalah yang akan dibahas dalam penelitian ini. Berikut adalah rumusan masalah dalam penelitian ini.

1. Bagaimana hasil evaluasi arsitektur *Convolutional Neural Network* untuk pengklasifikasian citra makanan tradisional ?

1.3 Tujuan

Tujuan utama yang diharapkan dari penelitian ini adalah sebagai berikut.

1. Mengetahui hasil evaluasi *Convolutional Neural Network* untuk pengklasifikasian citra makanan tradisional.

1.4 Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut

1. Membantu pengembang memilih klasifikasi dalam pengolah citra makanan.
2. Memberikan referensi untuk pengembangan dan pemajuan klasifikasi pengolahan citra di bidang ilmu komputasi cerdas.

1.5 Batasan Masalah

Berdasarkan rumusan masalah yang telah dipaparkan dan agar penelitian lebih berfokus maka dilakukan suatu batasan masalah sebagai berikut.

1. Jenis citra makanan terdiri dari 20 jenis makanan tradisional yaitu Wingko babat, Gethuk, Roti Bakar, Lumpia, Bakso Bakar, Lemper, Bugis, Onde-onde, Klepon, Tetel, Ketela Goreng, Kue Kucur, Putu Ayu, Molen, Bikang, Kue Sus, Pukis, Serabi, Pisang Ijo, dan Risoles.
2. Pengambilan data secara primer menggunakan kamera *smartphone* Meizu m4 dengan besar citra 1064 x 614.

1.6 Sistematika Pembahasan

Adapun sistematika laporan yang ditunjukkan untuk memberikan gambaran dan uraian dari laporan secara garis besar adalah sebagai berikut.

BAB I PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika penyusunan laporan.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini menguraikan konteks masalah yang berisi dasar teori dan konsep serta ilmu yang terkait tentang klasifikasi citra makanan menggunakan metode *Convolution Neural Network* (CNN).

BAB III METODOLOGI

Pada bab ini berisi metode yang digunakan dalam penelitian yang terdiri dari studi kepustakaan, analisis dan pemodelan kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, dan kesimpulan.

BAB III PERANCANGAN

Pada bab ini berisi metode yang digunakan dalam penelitian yang terdiri dari studi kepustakaan, analisis dan pemodelan kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian sistem, dan kesimpulan.

BAB V HASIL DAN PEMBAHASAN

Pada bab ini berisi mengenai hasil dari klasifikasi citra makanan menggunakan metode *Convolution Neural Network* (CNN) beserta pembahasan dari hasil tersebut.

BAB VI PENUTUP

Pada bab ini menguraikan kesimpulan tentang hasil klasifikasi citra makanan menggunakan metode *Convolution Neural Network* (CNN) beserta saran untuk pengembangan penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab tinjauan pustaka ini berisi uraian dan pembahasan tentang kajian pustaka dan dasar teori dalam penelitian ini. Metode yang digunakan dalam klasifikasi citra makanan menggunakan metode *Convolution Neural Network* (CNN). Maka dalam bab ini dijelaskan mengenai kajian pustaka, klasifikasi citra makanan menggunakan metode *Convolution Neural Network*.

2.1 Kajian Pustaka

Terdapat beberapa penelitian sebelumnya yang memiliki keterkaitan dengan penelitian ini baik dalam objek maupun metode yang digunakan, diantaranya klasifikasi citra makanan, dan *Convolution Neural Network* (CNN).

(Kumari, 2018) mengulas beberapa hasil penelitian mengenai segmentasi dan klasifikasi tumor otak. Dalam ulasannya kumari mendeskripsikan kelebihan setiap penelitian dan memberikan komentar untuk penelitian lanjut, bahkan kumari dapat menyimpulkan kelebihan dan kekurangan setiap metode yang digunakan untuk deteksi tumor otak. Sesuai dengan tujuannya yaitu membantu ahli radiologi dalam menemukan tumor lebih akurat dan efisien dengan waktu minimum dan akurasi maksimum, kumari menyimpulkan bahwa metode yang sering digunakan adalah algoritme klustering, akan tetapi *Convolution Neural Network* (CNN) merupakan metode baru dalam penelitian citra yang memiliki komputasi lebih efisien daripada metode lain yang ada. Karena CNN dilatih langsung oleh modalitas gambar sehingga mempelajari fitur kompleks langsung dari data.

Penelitian kedua, yaitu penelitian yang dilakukan oleh ye he dkk (2013). Penelitian ditujukan untuk penilaian diet yang mencatat asupan gizi harian menggunakan citra makanan yang diambil hendak akan makan. Dalam penelitiannya ye he menggunakan metode segmentasi *variant local* karena berdasarkan hasil penelitian Estrada (2009). Akan tetapi *variant local* menurut ye he masih mengalami *under-segmented* dan *over-segmented* karena masih bergantung pada pilihan parameter masukkan tertentu. Ye he mengatasi masalah tersebut menggunakan *confidence scores*. Akhirnya ye he dapat memperkirakan berat makanan berdasarkan relasi berat area, Interpolasi linier dari N area terdekat dalam data pelatihan.

Penelitian ketiga yaitu penelitian yang dilakukan oleh Zufar dan Setiyono (2016). Metode deep neural networks yaitu *Convolutional Neural Networks* (CNN) sebagai pengenalan wajah secara real-time. Metode diimplementasikan dengan bantuan library OpenCV untuk deteksi multi wajah dan perangkat Web Cam M-Tech 5MP. Dalam penyusunan arsitektur model *Convolutional Neural Networks* dilakukan konfigurasi inialisasi parameter untuk mempercepat proses training jaringan. Hasil uji coba dengan menggunakan konstruksi model *Convolutional Neural Networks* sampai kedalaman 7 lapisan dengan *input* dari hasil ekstraksi Extended Local Binary Pattern dengan radius 1

dan neighbor 15 menunjukkan kinerja pengenalan wajah meraih rata-rata tingkat akurasi lebih dari 89% dalam \mp 2 frame per detik.

Tabel 2.1 Kajian Pustaka

No	Judul	Objek	Metode Proses	Keterangan
1	<i>Review of Brain Tumor Segmentation and Classification</i>	Citra MRI tumor otak	K-Means, Fuzzy C-Means, CNN, SVM, ensemble base classifier, PSO, dan Adaboost	Setiap metode memiliki kelebihan dan kekurangan masing-masing, akan tetapi CNN merupakan metode baru yang lebih efisien dari yang lain.
2	<i>Food Image Analysis: Segmentation, Identification And Weight Estimation</i>	Makanan	Ekstraksi fitur warna yaitu <i>Scalable Color Descriptor (SCD)</i> , <i>Color Structure Descriptor (CSD)</i> , <i>Dominan Color Descriptor (DCD)</i> dan Warna <i>Layout Descriptor (CLD)</i> dalam standar <i>MPEG-7</i> Ekstraksi fitur tekstur yaitu <i>Gradient Orientation Spatical Dependence Matrix (GOSDM)</i> , <i>Entropy-Based Categorization and Fractal Dimension Estimation (EFD)</i> dan <i>GaborBased Image Decomposition and Fractal Dimension Estimation (GFD)</i> . Klasifikasi menggunakan KNN	Klasifikasi makanan sangat sulit karena makanan bisa secara dramatis bervariasi dalam penampilan seperti bentuk, tekstur, warna dan sifat visual lainnya.

Tabel 2.1 Kajian Pustaka (Lanjutan)

3	<i>Convolutional Neural Networks</i> untuk Pengenalan Wajah Secara Real-Time	Wajah Manusia	<i>Convolutional Neural Network</i>	menunjukkan kinerja pengenalan wajah meraih rata-rata tingkat akurasi lebih dari 89% dalam \mp 2 frame per detik.
4	Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi	Teknologi UAV (<i>Unmanned Aerial Vehicle</i>)	<i>Convolutional Neural Network</i>	Penelitian menggunakan 5 kelas jenis tanaman, yaitu kelas tanaman padi, bawang merah, kelapa, pisang, dan cabai. Proses learning jaringan menghasilkan akurasi 100% terhadap data training. Pengujian terhadap data validasi menghasilkan akurasi 93% dan akurasi terhadap data tes 82%.
5	Penerapan <i>Deep Learning</i> Menggunakan <i>Convolutional Neural Network</i> Untuk Klasifikasi Citra Wayang Punakawan	Citra Wayang Punakawan	<i>Convolutional Neural Network</i>	Hasil yang diperoleh adalah model dapat mengenali dan mengklasifikasikan data citra uji dengan akurasi sebesar 91.6 %

Tabel 2.2 Kajian Pustaka (Lanjutan)

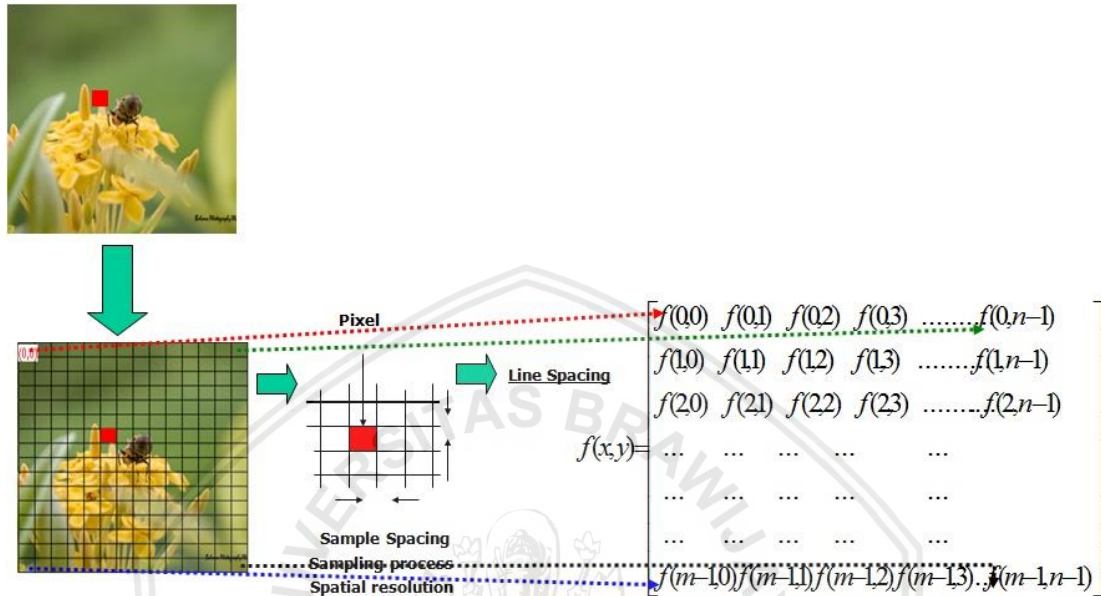
6	Implementasi <i>Convolutional Neural Networks</i> Untuk Klasifikasi Citra Tomat Menggunakan <i>Keras</i>	Citra Tomat	<i>Convolutional Neural Network</i>	Pengujian dengan sampel 100 citra tomat menunjukkan tingkat akurasi sebesar 90% yang dinilai telah mampu melakukan identifikasi kelayakan buah tomat.
7	<i>Convolutional Neural Networks for Handwritten Javanese Character Recognition</i>	<i>Javanese Character</i>	<i>Convolutional Neural Networks</i>	Hasil pengujian menunjukkan akurasi dari model CNN mampu mengungguli akurasi dari model MLP meskipun CNN membutuhkan waktu latih yang lebih lama dibandingkan dengan MLP.

2.2 Dasar Teori

2.2.1 Citra Digital

Definisi citra menurut kamus webster adalah suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda. Citra dapat dikelompokkan menjadi citra tampak dan tak tampak (Achmad, 2013). Citra tampak adalah cahaya yang direfleksikan dari sebuah objek (citra optis). Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik. Citra tampak seperti foto, gambar, lukisan, apa yang nampak dilayar monitor, serta hologram (citra optis). Sedangkan citra tak tampak merupakan citra yang cenderung tidak dapat disentuh dan hanya dapat dilihat serta dapat diolah. Citra tak tampak agar dapat dilihat mata manusia maka harus dirubah menjadi citra tampak terlebih dahulu, seperti ditampilkan di monitor, dicetak di atas kertas, dan lain sebagainya. Citra tak tampak seperti citra gambar pada file (citra digital), citra yang dinyatakan dalam fungsi matematis. Disamping itu ada juga citra fisik tak tampak seperti peta densitasis material, citra distribusi panas bumi, kulit manusia.

Diantara beberapa jenis citra tersebut yang dapat diolah komputer hanya citra digital. Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (Abdul Kadir, 2013). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan di Gambar 2.1.



Gambar 2.1 Citra Digital

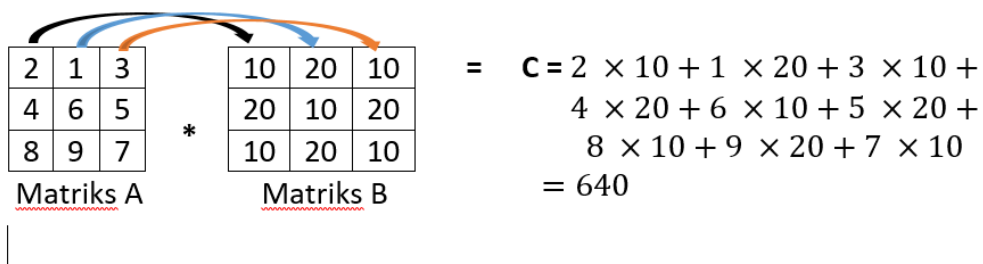
Suatu citra berukuran $N \times M$ pixel yang memiliki intensitas beragam pada tiap pikselnya, direpresentasikan secara numerik dalam bentuk matriks yang berukuran N baris dan M kolom. Sebuah piksel memiliki fungsi $f(x, y)$.

- x menyatakan kolom.
- y menyatakan baris.
- Piksel pojok kiri atas mempunyai koordinat $(0, 0)$ dan piksel kanan bawah mempunyai koordinat $(N-1, M-1)$

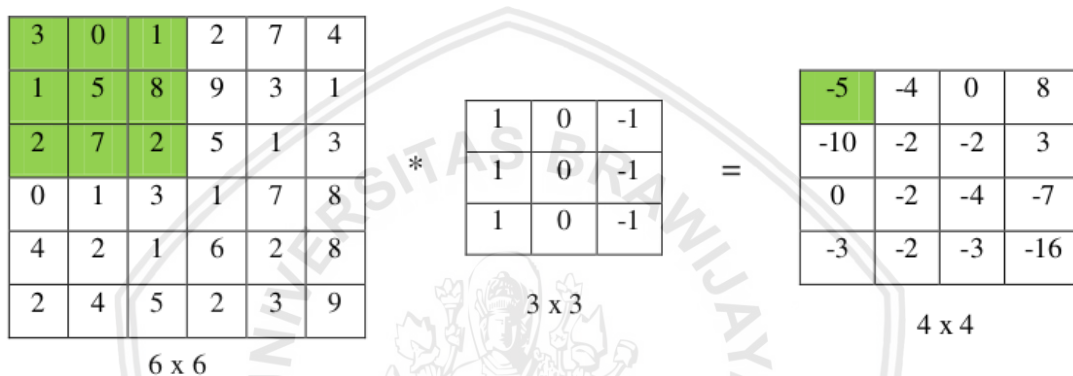
Apabila jenis citra lain hendak diolah komputer maka harus diubah menjadi citra digital terlebih dahulu. Kegiatan untuk mengubah informasi citra fisik non-digital menjadi digital disebut digitalisasi.

2.2.2 Operasi Konvolusi

Konvolusi (*convolution*) adalah sebuah proses dimana citra dimanipulasi dengan menggunakan eksternal *mask* atau subwindows untuk menghasilkan citra yang baru. Secara matematis menurut (Madenda, 2015) konvolusi adalah jumlah total dari hasil kali antara setiap elemen yang bersesuaian (memiliki posisi koordinat yang sama) dalam dua matriks atau dua vektor, seperti yang ditunjukkan Gambar 2.2.



Gambar 2.2 Operasi Konvolusi Matriks

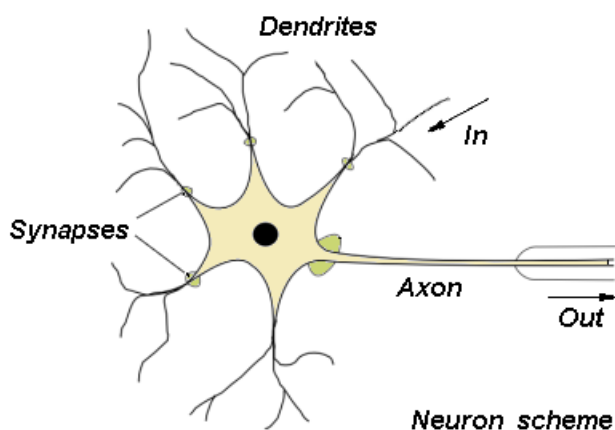


Gambar 2.3 Filter citra menggunakan Konvolusi

Gambar 2.3 merupakan contoh proses pemfilteran menggunakan konvolusi antara citra dengan ukuran 6x6 piksel dan *subwindows* (kernel) ukuran 3x3 piksel. Konvolusi dilakukan melalui proses *masking*, yaitu memetakan elemen sentral matriks kernel 3x3 terhadap setiap piksel citra 6x6. Pemetaan dimulai dengan piksel kedua dari baris maupun kolom. Hasil dari konvolusi berupa citra dengan ukuran 4x4 piksel. Konvolusi sangat banyak dipergunakan dalam pengolahan citra untuk memperhalus (*smoothing*), menajamkan (*crispening*), mendeteksi tepi (*edge detection*), serta efek lainnya.

2.2.3 Neural Network

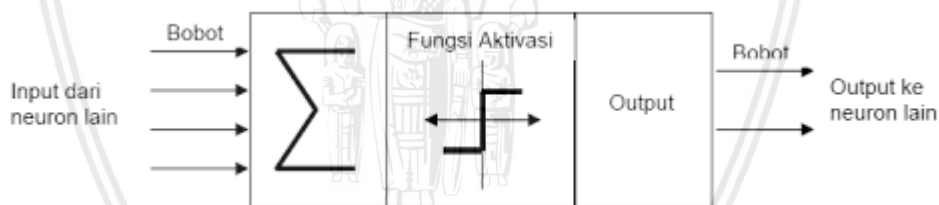
Neural Network merupakan cabang ilmu kecerdasan buatan untuk pemrosesan informasi. Ide mendasar dari *neural network* adalah mengadopsi mekanisme berpikir sebuah sistem atau aplikasi yang menyerupai otak manusia, baik untuk pemrosesan berbagai sinyal elemen yang diterima, toleransi terhadap kesalahan, dan *parallel processing* (Widodo Budiharto, 2014). Otak manusia memiliki sekitar 10¹¹ neuron. Neuron berfungsi memroses informasi yang masuk. Satu neuron memiliki 1 akson dan minimal 1 dendrit. Setiap sel syaraf terhubung dengan dengan syaraf lain, jumlahnya mencapai sekitar 10⁴ sinapsis.



Gambar 2.4 Struktur neuron pada otak manusia

Berdasarkan gambar 2.4 dapat dilihat beberapa bagian otak manusia, yaitu :

- Dendrit (*Dendrites*) berfungsi untuk mengirimkan impuls yang diterima ke badan sel syaraf.
- Akson (*Axon*) berfungsi untuk mengirimkan impuls dari sel ke jaringan lain.
- Sinapsis berfungsi sebagai unit fungsional diantara dua sel syaraf.

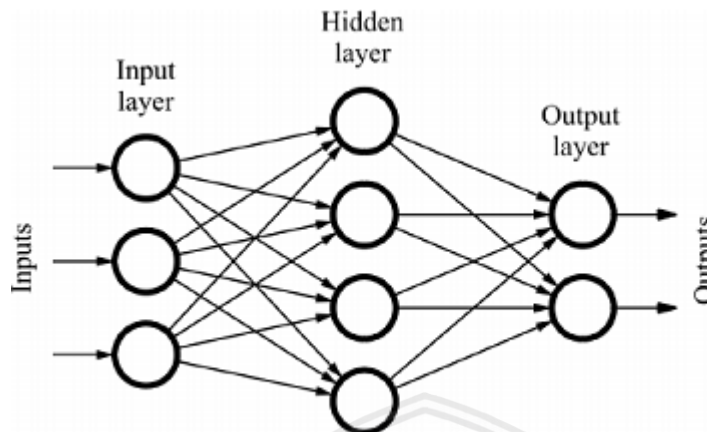


Gambar 2.5 Struktur Neural Network

Gambar 2.5 menjelaskan struktur *Neural Network* secara mendasar, yaitu *input* berfungsi seperti dendrit, *output* berfungsi seperti akson, dan Fungsi aktivasi berfungsi seperti sinapsis.

Neural network dibuat dari sebuah node atau unit yang memiliki link untuk menghubungkan antara satu node dengan node lainnya. Link digunakan untuk melakukan propagasi aktivasi dari node pertama ke node selanjutnya. Setiap link memiliki bobot numerik. Bobot berfungsi untuk menentukan kekuatan serta penanda dari sebuah konektivitas. *Neural network* terdiri beberapa neuron. Neuron dikelompokkan kedalam beberapa *layer*. Neuron yang terdapat pada tiap *layer* dihubungkan dengan neuron pada *layer* lainnya. Informasi yang diterima *layer input* dilanjutkan ke *layer-layer* berikutnya hingga mencapai *layer* terakhir atau *layer output*. *Layer* yang terletak diantara *input* dan *output* disebut sebagai *hidden layer*. Seperti yang ditunjukkan pada gambar 2.6 bahwa dalam neural

network terdapat *layer input*, *hidden layer*, *layer output*, dan setiap link memiliki bobot. Namun tidak semua *Neural Network* memiliki *hidden layer*.



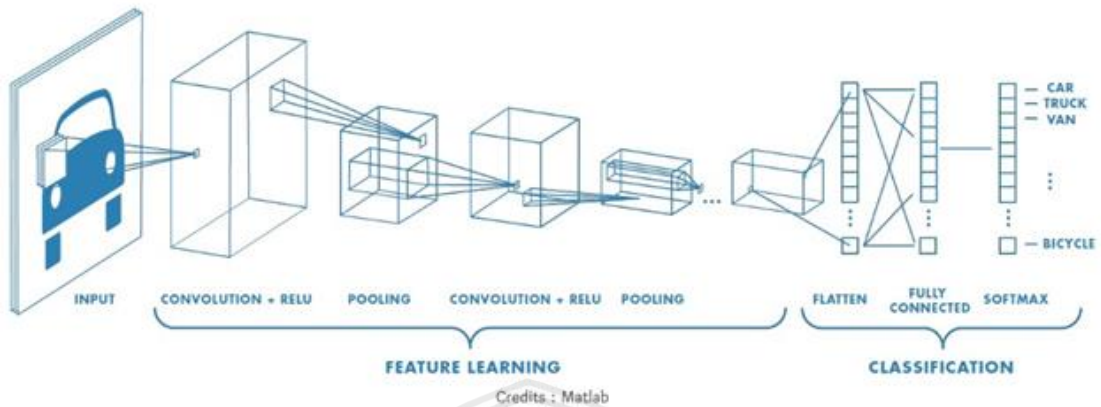
Gambar 2.6 Multi Layer Neural Network

2.2.4 Convolutional Neural Network (CNN)

CNN adalah variasi dari Multi Layer Perceptron yang terinspirasi dari jaringan syaraf manusia. Penelitian awal yang mendasari penemuan ini pertama kali dilakukan oleh Hubel dan Wiesel, yang melakukan penelitian visual cortex pada indera penglihatan kucing. *Visual cortex* pada hewan sangat powerful dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan model-model baru, salah satunya LeNet. LeNet adalah salah satu jaringan saraf convolutional pertama yang membantu mendorong bidang Pembelajaran Dalam. Karya perintis oleh Yann LeCun ini dinamai LeNet5 setelah banyak iterasi sukses sebelumnya sejak tahun 1988. Pada saat itu arsitektur LeNet digunakan terutama untuk tugas pengenalan karakter seperti membaca kode pos, angka, dll. Arsitektur jaringan dengan menggunakan CNN ditunjukkan pada Gambar 2.7 Struktur CNN terdiri dari *input*, proses ekstraksi fitur, proses klasifikasi dan output. Proses ekstraksi dalam CNN terdiri dari beberapa lapisan tersembunyi atau *hidden layer*, yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan *pooling*. CNN bekerja secara hierarki, sehingga output pada lapisan konvolusi pertama digunakan sebagai *input* pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari *fully connected* dan fungsi aktivasi (*softmax*) yang keluarannya berupa hasil klasifikasi (Katole et al., 2015).

Convolutional Neural Networks merupakan suatu *layer* yang memiliki susunan neuron 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman ditentukan setiap *layer* nya. *Input layer* kedalaman berdasarkan warna citra seperti merah, biru, dan hijau maka kedalaman citra awal lebar x tinggi x 3 gabungan warna. Pada *layer* ekstraksi fitur kedalaman berdasarkan

banyaknya neuron. Secara umum proses yang ada pada CNN ialah *Convolution*, *Pooling*, *Fully connected*, dan Fungsi Aktivasi *Softmax*



Gambar 2.7 Arsitektur CNN

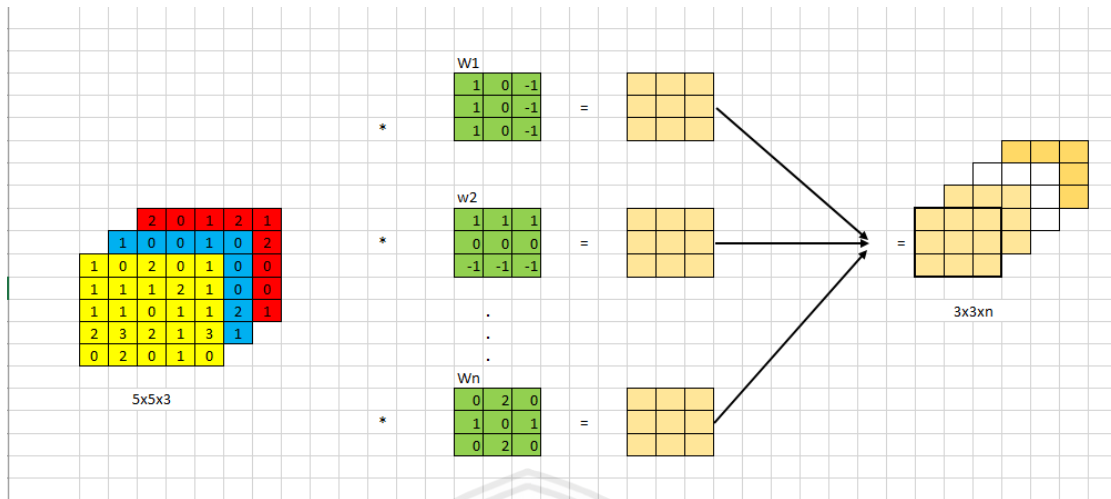
1. Convolution Layer

Layer yang pertama kali menerima *input* gambar langsung pada arsitektur. Operasi pada *layer* ini sama dengan operasi konvolusi yaitu melakukan operasi kombinasi linier filter terhadap daerah lokal seperti perhitungan Gambar 2.2. Filter atau bobot (W_n) merupakan representasi bidang reseptif dari neuron yang terhubung kedalam daerah lokal (*local connectivity*) pada *input* gambar. Bentuk *layer* direpresentasikan sebagai volume $B \times K \times L$ atau *layer* ukuran $B \times K$ dengan jumlah sebanyak L . Perhitungan *Depth* pada *layer* konvolusi berdasarkan banyaknya filter atau neuron seperti yang ditunjukkan Gambar 2.8, sedangkan untuk perhitungan matriks keluaran hasil konvolusi ditunjukkan persamaan 2.1. Convolutional *layer* memiliki hyperparameter dan parameter yang ditunjukkan Tabel 2.2.

Tabel 2.2 Hyperparameter pada Convolutional Layer

No	Hyperparameter	Keterangan
1	Depth	Kedalaman <i>layer</i> atau jumlah <i>layer</i> konvolusi
2	Stride	Jumlah pergeseran <i>filter</i> pada proses konvolusi
3	Zero-padding	Jumlah penambahan nilai intensitas nol di daerah sekitar <i>input</i> gambar

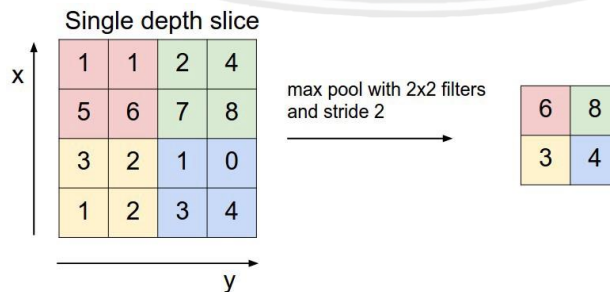
$$matriks_{output} = \frac{image_{size} - filter}{stride} + 1 \quad (2.1)$$



Gambar 2.8 Layer Konvolusi

2. Pooling

Pooling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *pooling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur serta mempercepat komputasi dan mengontrol terjadinya *overfitting*. *Pooling layer* memiliki beberapa macam tipe antara lain *average pooling*, *max pooling*, dan *Lp Pooling*. *Max pooling* membagi output dari convolution layer menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.9. Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami *translasi* (pergeseran). Perhitungan besar matriks keluaran dari proses *pooling* seperti perhitungan konvolusi pada persamaan 2.1, karena pada dasarnya sama yaitu dipengaruhi oleh besarnya filter dan *stride*.



Gambar 2.9 Max Pooling

3. Fungsi aktivasi

Fungsi aktivasi atau fungsi transfer merupakan fungsi non-linier yang memungkinkan sebuah jaringan untuk dapat menyelesaikan permasalahan

permasalahan non-trivial. Setiap fungsi aktivasi mengambil sebuah nilai dan melakukan operasi matematika. Pada arsitektur CNN, fungsi aktivasi terletak pada perhitungan akhir keluaran feature map atau sesudah proses perhitungan konvolusi atau *pooling* untuk menghasilkan suatu pola fitur. Beberapa macam fungsi aktivasi yang sering digunakan dalam penelitian antara lain fungsi *sigmoid*, *tanh*, *Rectified Linear Unit (ReLU)*, *Leaky ReLU (LReLU)* dan *Parametric ReLU*.

2.2.5 Confusion Matrix

Confusion matrix adalah salah satu metode yang digunakan untuk mengevaluasi metode-metode klasifikasi. Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dibagi menjadi 4 (empat) jenis yaitu klasifikasi *binary*, *multi-class*, *multi-label* dan *hierarchical* (Sokolova & Lapalme, 2009). *Binary* digunakan apabila keluaran berupa *Boolean* atau dua kelas. Sedangkan *multi-class* digunakan apabila keluaran terdiri dari beberapa kelas. Jenis *multi-label* hampir sama dengan *multi-class* namun *multi-label* dapat melabelkan beberapa kelas sekaligus. Jenis klasifikasi yang terakhir adalah *hierarchical*, penggunaan jenis *hierarchical* apabila keluaran berupa kelas-kelas yang digolongkan kembali menjadi beberapa kelas secara hirarki. Pada penelitian ini menggunakan jenis *multi-class* karena keluaran terdiri dari beberapa jenis makanan tradisional.

Dalam mengevaluasi hasil klasifikasi, *confusion matrix* memiliki 4 istilah umum sebagai representasi proses keluaran. Keempat istilah itu terdiri dari *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)* dan *False Negative (FN)*. Nilai *True Negative (TN)* adalah data yang di klasifikasi dengan tepat sebagai keluaran negatif atau salah. *True Positive (TP)* adalah data yang diklasifikasi dengan tepat sebagai keluaran positif atau benar. *False Positive (FP)* adalah data yang diklasifikasi dengan kurang tepat apabila keluaran berupa positif atau benar. *False Negative (FN)* adalah data yang diklasifikasi dengan kurang tepat apabila keluaran berupa negatif atau salah. Tabel 2.3 menunjukkan gambaran sederhana untuk mempermudah pemahaman tentang istilah *confusion matrix* dalam keluaran klasifikasi.

Tabel 2.3 Istilah *Confusion Matrix*

		Kelas Prediksi (<i>Observation</i>)	
		Positif	Negatif
Kelas Aktual (<i>Expextation</i>)	Positif	TP	FN
	Negatif	FP	TN

Berdasarkan istilah *confusion matrix* seperti yang ditunjukkan Tabel 2.3 ada beberapa cara mengevaluasi keluaran diantaranya yaitu *Accuracy*, *Recall*, *Precision*, *F-score*.

$$\text{Precision} = \frac{\sum_i^n \frac{TP_i}{TP_i + FP_i}}{n} \quad (2.2)$$

Persamaan 2.2 merupakan perhitungan rata-rata nilai *precision* yaitu dari data hasil klasifikasi seberapa banyak data yang benar. antara nilai sebenarnya dengan prediksi yang diberikan oleh sistem. Evaluasi presisi digunakan apabila menginginkan keluaran yang mengutamakan nilai *True Positif* (Ghoneim, 2019), dan mengesampingkan nilai *False Positif*. Seperti pada klasifikasi *email spam*, bahwa email yang sebenarnya *spam* namun diprediksi tidak *spam* lebih dipilih daripada email yang sebenarnya bukan *spam* tapi diprediksi *spam*.

$$\text{Recall} = \frac{\sum_i^n \frac{TP_i}{TP_i + FN_i}}{n} \quad (2.3)$$

Persamaan 2.3 merupakan perhitungan rata-rata nilai *recall* yaitu dari seluruh data benar seberapa banyak data yang keluar dalam hasil klasifikasi. Evaluasi *recall* digunakan apabila lebih memilih nilai *False Positif* daripada *False Negatif* (Ghoneim, 2019). Seperti pada memprediksikan seseorang terkena HIV atau tidak, bahwa memprediksikan seseorang yang sehat berlabel HIV lebih dipilih daripada melabelkan sehat kepada seseorang yang terkena HIV.

$$\text{Akurasi} = \frac{\sum_i^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} \quad (2.4)$$

$$\text{F-score} = \frac{\sum_i^n \frac{\text{recall}_i \times \text{presisi}_i}{\beta(\text{recall}_i) + (1-\beta)(\text{presisi}_i)}}{n} \quad (2.5)$$

Persamaan 2.4 merupakan perhitungan rata-rata nilai akurasi untuk menunjukkan tingkat efektifitas per kelas dari sebuah klasifikasi (Sokolova & Lapalme, 2009). Sedangkan persamaan 2.5 merupakan perhitungan rata-rata nilai *F-score* yang merupakan nilai kombinasi dari perhitungan *recall* dan presisi. Akurasi merupakan acuan bagus apabila data keluaran simetris. Seperti contoh berikut apabila TP = 10, FP = 9990, FN = 0, TN = 0. Maka presisinya 0.001, recallnya 1.0, dan akurasinya 0.1. Akurasinya rendah sedangkan semua data sebenarnya telah terprediksi. Jadi penggunaan *F-score* dibutuhkan dari pada akurasi pada permasalahan tersebut.

2.2.6 Normalisasi

Normalisasi atau *scaling* adalah prosedur mengubah data sehingga berada dalam skala tertentu (Santosa & Umam, 2018). Skala penentuannya tergantung kebutuhan setiap sistem contoh skala [0, 1], [-1, 1], atau skala lainnya. Normalisasi memiliki berbagai cara merubah data diantaranya normalisasi *scaling decimal*, *sigmoid*, *softmax*, *min-max*, *Statiscal column*, dan *Z-score*. Dalam penelitian (Chamidah, Wiharto, & Salamah, 2012) metode normalisasi yang memberikan efektifitas terbaik dalam hal akurasi dan kecepatan konvergensi adalah metode Minmax yang mencapai akurasi rata-rata hingga 96.86% dan memerlukan iterasi

yang paling sedikit dengan rata-rata sebanyak 21 *epoch*. Perhitungan min-max ditunjukkan pada persamaan 2.6.

$$s' = \frac{s - \min\{s_k\}}{\max\{s_k\} - \min\{s_k\}} \quad (2.6)$$

Berdasarkan persamaan 2.6 jika dihubungkan dengan nilai citra, maka variabel s_k merupakan himpunan nilai piksel pada citra dan s merupakan nilai per piksel. Nilai minimal pada citra RGB adalah 0 (nilai hitam) dan nilai maksimalnya adalah 255 (nilai putih). Perhitungan yang digunakan berdasarkan informasi yang ada pada citra menggunakan persamaan (2.7).

$$s' = \frac{s - 0}{255 - 0}$$
$$s' = \frac{s}{255} \quad (2.7)$$



BAB 3 METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah-langkah yang dilakukan dalam membuat analisis kebutuhan, perancangan, implementasi, dan pengujian dari sistem yang telah dibuat. Penarikan kesimpulan dan saran akan disertakan sebagai catatan atas sistem dan kemungkinan arah pengembangan dimasa yang akan datang.

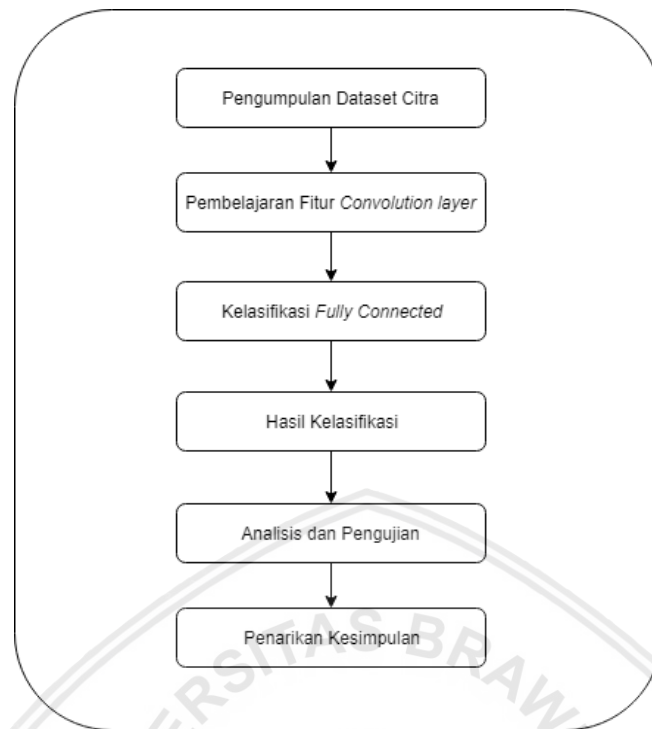
3.1 Tipe Penelitian

Penelitian non-implementatif merupakan penelitian yang menitikberatkan pada investigasi terhadap keadaan atau fenomena tertentu. Teknik yang digunakan peneliti adalah hasil dari studi kasus serta proses wawancara tentang objek penelitian yang digunakan. Sedangkan penelitian analitik adalah kegiatan penelitian yang memaparkan sebuah hasil analisis yang dihasilkan dari penjelasan hubungan antar elemen yang terdapat pada objek penelitian dengan kondisi tertentu yang diteliti. Secara garis besar, tipe non-implementatif analitik ini merupakan sebuah proses yang berkonsentrasi pada penggalian informasi dari suatu keadaan yang mampu menjawab pertanyaan-pertanyaan yang muncul pada awal penelitian.

Tipe penelitian yang dilakukan pada penelitian ini adalah Non-implementatif Analitik, yang mana tipe penelitian ini akan mengkaji Klasifikasi citra makanan, pengaruh penggunaan tipe *pooling* CNN dan algoritme perhitungan mempelajari fitur dengan otomatis. Produk utama yang dihasilkan berupa hasil analisis ilmiah dari hasil pengujian.

3.2 Strategi Penelitian

Strategi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya. Metodologi penelitian yang digunakan dalam penelitian ini akan melalui beberapa tahapan yang membentuk alur yang sistematis, seperti ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Berdasarkan Gambar 3.1 Alur pengerjaan penelitian ini dimulai dari pengumpulan data sampel yang digunakan sebagai *input* untuk data training, validasi dan pengujian/tes. Kemudian merancang jaringan dengan metode CNN untuk melakukan klasifikasi pada citra makanan. Rancangan jaringan CNN diaplikasikan dengan data training, agar komputer dapat belajar mengenali objek. Untuk mengetahui pembelajaran jaringan diperoleh hasil yang baik dalam membedakan jenis objek tanaman, maka jaringan tersebut kemudian dilakukan uji coba terhadap data validasi. Apabila data validasi juga menunjukkan hasil yang baik maka jaringan dapat digunakan untuk klasifikasi pada data tes

3.2.1 Lokasi Penelitian

Lokasi penelitian yang dipilih adalah di lingkungan Universitas Brawijaya, yaitu di Fakultas Ilmu Komputer, Perpustakaan Pusat Universitas Brawijaya dan lingkungan sekitar Universitas Brawijaya. Pemilihan lokasi penelitian ini, didasarkan atas fasilitas jaringan internet yang memadai dan juga lingkungan yang aman, tenang dan kondusif.

3.2.2 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer yang berupa citra yang diambil menggunakan teknik sebagai berikut.

1. Citra berwarna berekstensi *.jpg
2. Citra diambil langsung menggunakan kamera smartphone Meizu M4.

3. Citra diambil di gedung F Lantai 9 di depan ruang F9.3-Grup Riset Computer Vision dengan pencahayaan matahari pada pukul 10.00-13.00 WIB hari Selasa, 28 Agustus 2018.
4. Citra diambil citra makanan tradisional sebanyak 1 buah, 2 buah, dan saat makanan terpotong, dengan teknis pengambilan citra seperti berikut.
 - a. Tegak lurus dengan pengambilan 3 level ketinggian, dengan ketinggian menyesuaikan dengan makanannya dan tiap ketinggian dilakukan 3 kali pengambilan citra.
 - b. 3 jenis kemiringan secara random, tiap kemiringan dilakukan 3 kali pengambilan citra.
5. Gambar 3.2 dan Gambar 3.3 merupakan contoh pengambilan citra





Gambar 3.3 Contoh Pengambilan Citra 1







Gambar 3.2 Contoh Pengambilan Citra 2

6. Tabel 3.1 merupakan informasi tentang jumlah data dalam penelitian ini, yang nantinya akan dibagi menjadi dua untuk data latih dan data uji. Data uji akan diambil 5 citra setiap kelasnya.







Tabel 3.1 Data Citra

Kode	Nama Makanan	Jumlah	Gambar Citra
0	Wingko babat	16	
1	Gethuk	16	







Tabel 3.1 Data Citra (Lanjutan)

2	Roti Bakar	23	
3	Lumpia	21	
4	Bakso Bakar	21	
5	Lemper	21	
6	Bugis	23	
7	Onde-onde	24	

Tabel 3.1 Data Citra (Lanjutan)

8	Klepon	17	
9	Tetel	18	
10	Ketela Goreng	17	
11	Kue Kucur	19	
12	Putu Ayu	17	
13	Molen	22	

Tabel 3.1 Data Citra (Lanjutan)

14	Bikang	17	
15	Kue Sus	13	
16	Pukis	14	
17	Serabi	22	
18	Pisang Ijo	15	
19	Risoles	24	

3.2.3 Peralatan Pendukung

Peralatan pendukung diperlukan untuk menunjang proses penelitian agar berjalan dengan yang diharapkan. Oleh karena itu, pada penelitian ini diperlukan peralatan pendukung berupa perangkat keras dan perangkat lunak yang ditunjukkan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.2 Perangkat Keras Pendukung

Nama Komponen	Keterangan
Processor	Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz
RAM	10.0GB RAM DDR3
Storage	500GB HDD
Smartphone	Meizu M4

Tabel 3.3 Perangkat Lunak Pendukung

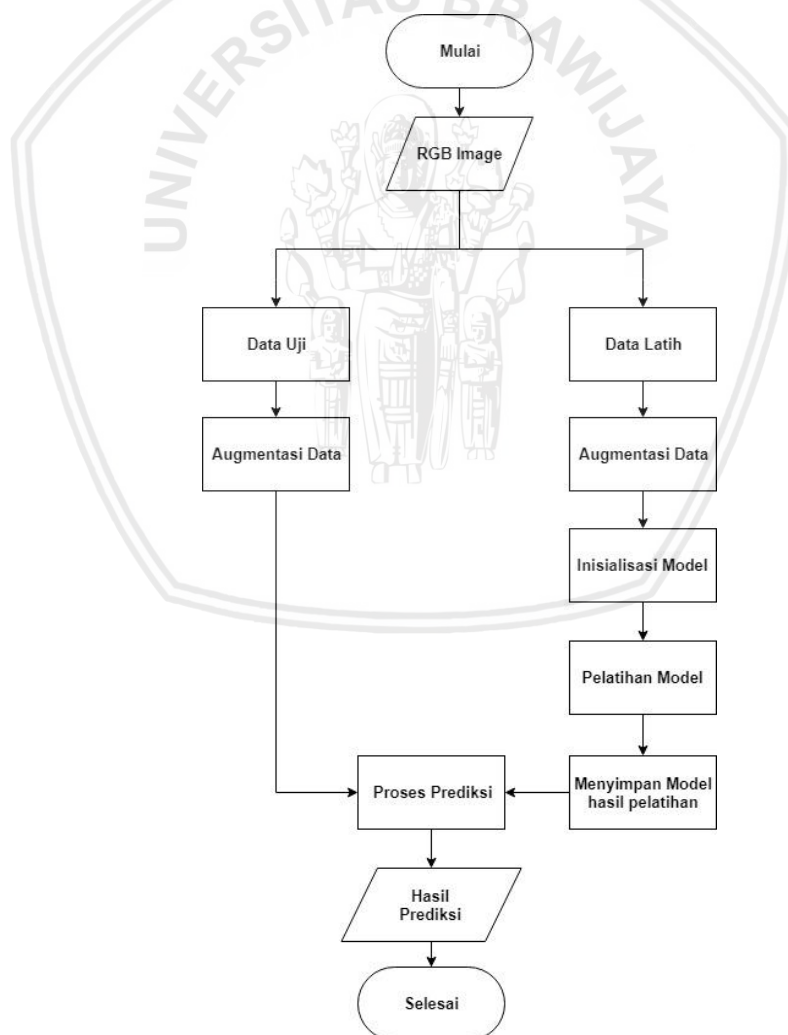
Nama Komponen	Keterangan
Sistem Operasi	Windows 10 Pro 64-bit
Bahasa Pemrograman	Python v3.6
Pengolah kode sumber	JetBrains PyCharm Community Edition 2018.1.4 x64
Pengolah dokumen	Microsoft Word 2016

BAB 4 PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan yang akan digunakan sebagai acuan dalam pengimplementasian metode dalam penelitian ini. Perancangan terdiri dari perancangan algoritme, perhitungan manualisasi dan perancangan skenario pengujian.

4.1 Perancangan Proses CNN

Perancangan proses digambarkan dalam bentuk *flowchart* (diagram alir). Hal ini bertujuan untuk lebih memahami alur suatu proses. Perancangan proses yang dibuat dalam penelitian ini secara umum dibagi menjadi 2 bagian, bagian pertama adalah proses pelatihan atau pembelajaran sistem terhadap data latih, dan kedua proses pengklasifikasian data uji. Perancangan keseluruhan sistem disusun dalam diagram alir yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Sistem Secara Umum

Seperti yang terlihat pada Gambar 4.1, sistem menerima masukan berupa citra makanan. Citra makanan tersebut dibagi menjadi dua yaitu data untuk pelatihan dan data untuk pengujian hasil pelatihan. Data citra pelatihan akan dinormalisasi yang kemudian dilakukan proses pembelajaran dengan suatu model yang telah diinisialisasikan atau sudah ditentukan arsitekturnya. Setelah proses pembelajaran selesai maka bobot, bias, dan arsitekturnya akan disimpan agar saat ingin memprediksi citra baru tidak perlu untuk melakukan proses pembelajaran kembali. Proses pembelajaran sudah selesai maka dilakukan proses prediksi dengan data uji yang telah dinormalisasikan.

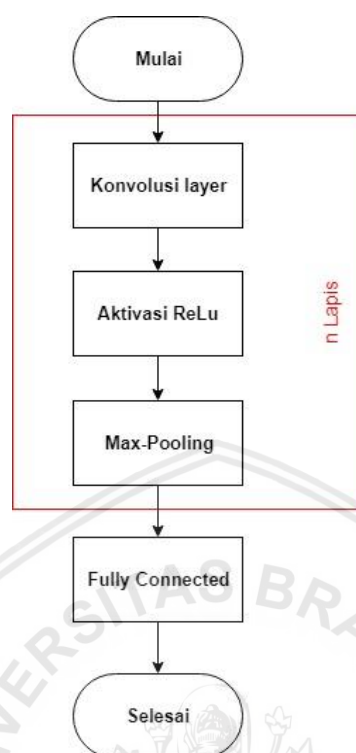
4.1.1 Augmentasi Data

Augmentasi data adalah pemrosesan data sebelum dilakukannya proses utama atau mengubah data mentah menjadi data berkualitas yang biasa disebut dengan *preprocessing*. Berbagai macam cara melakukan augmentasi data tergantung kebutuhan sistem, diantaranya yang sering digunakan dalam memproses sebuah citra adalah *segmentasi*, *Cropping* citra, *Resize* citra, *Edge detection*, *edge enhancement*, *Thinning*, *flip* citra, *rotate* citra, dll. Akan tetapi dalam penelitian ini penggunaan augmentasi citra hanya untuk menormalisasi citra, atau mengubah nilai perpixel citra menjadi nilai diantara 0 dan 1. Menggunakan hanya normalisasi bertujuan untuk mengetahui kemampuan CNN dalam memahami citra.

Tahap normalisasi data dilakukan untuk menghindari dominasi dari variabel yang bernilai besar terhadap variabel bernilai kecil. Normalisasi dilakukan dengan mengubah nilai dari data ke dalam rentang nilai nol sampai satu. Tujuan dilakukan normalisasi adalah untuk skema penghambatan atau dalam Bahasa neurobiology (ilmu saraf dalam biologi) biasa disebut dengan *inhibition lateral*. Fungsi *inhibition lateral* adalah proses stimulasi neuron menghambat aktivitas neuron terdekat. Penghambatan *lateral* memungkinkan otak untuk mengelola input lingkungan dan menghindari informasi yang berlebihan. Dengan meredam aksi dari beberapa input indera dan meningkatkan aksi dari yang lain, hambatan lateral membantu mempertajam persepsi indra tentang penglihatan, suara, sentuhan, dan bau.

4.1.2 Inisialisasi Model

Inisialisasi model adalah menentukan lapisan yang akan digunakan dan kedalaman dari metode CNN. Metode CNN berfokus pada pembelajaran fitur pada suatu gambar maka yang perlu untuk ditentukan adalah *layer convolutional* yaitu lapisan konvolusi dan *pooling*, sedangkan *fully Connected* adalah metode klasifikasinya maka dalam penelitian ini *fully Connected layer* cukup dengan 2 lapisan. Gambar 4.2 merupakan ilustrasi model yang akan digunakan.



Gambar 4.2 Perancangan model

Berdasarkan gambar 4.2 lapisan konvolusi dan *pooling* sangat berpengaruh dalam menentukan hasil klasifikasi. Perlu dianalisis jumlah minimal dan maksimalnya layer-layer yang akan digunakan sebelum diimplementasikan. Lapisan konvolusi dan *pooling* menghasilkan keluaran dimensi yang lebih kecil atau sama dengan masukannya. Persamaan 2.1 merupakan persamaan untuk matriks keluaran setelah melakukan proses konvolusi dan *pooling*.

Berdasarkan persamaan 2.1 matriks keluaran ditentukan dengan matriks dari filter dan besarnya *stride* yang digunakan. Dalam penelitian ini filter yang digunakan adalah filter 3x3 dengan *stride* samadengan 1 untuk proses konvolusi. Semakin kecil ukuran filter semakin detail sistem akan mendapatkan informasi dari suatu citra masukan, akan tetapi komputasi akan semakin besar pula. Proses *pooling* menggunakan filter 2x2 dengan *stride* 2. Pada Tabel 4.1 merupakan hasil perhitungan citra dan kedalaman lapisan konvolusi dan *pooling* yang dapat digunakan dalam penelitian ini.

Tabel 4.1 Perancangan Kedalaman Model

No	Lapis n	Proses (lapisan)	Matriks Keluaran
1		Citra	1064x614
2	1	konvolusi	1062x612
3		<i>Pooling</i>	511x306
4	2	konvolusi	509x304
5		<i>Pooling</i>	254x152

Tabel 4.1 Perancangan Kedalaman Model (Lanjutan)

No	Lapis n	Proses (lapisan)	Matriks Keluaran
6	3	konvolusi	252x150
7		Pooling	126x75
8	4	konvolusi	124x73
9		Pooling	62x36
10	5	konvolusi	60x34
11		Pooling	30x17
12	6	konvolusi	28x15
13		Pooling	14x7
14	7	konvolusi	12x5
15		Pooling	6x2

Berdasarkan Tabel 4.1 kedalam model yang dapat digunakan untuk citra 1064x614 adalah 7 kedalaman *layer convolutional* (konvolusi dan *pooling*). Dalam melakukan proses konvolusi, besar matriks masukan minimal harus lebih atau samadengan besar matriks filternya. Penggunaan filter 3x3 untuk konvolusi tidak dapat lebih dari tujuh kedalaman *layer convolutional* karena keluaran pada lapisan 7 berupa matriks 6x2.

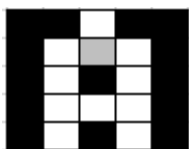
4.1.3 Pelatihan Model

Pelatihan Model merupakan proses pembelajaran model hasil inialisasi sebelumnya dengan data yang sudah dinormalisasikan. Dalam pelatihan model metode optimasi sangat menentukan kecepatan pembelajaran sistem. Ada berbagai metode optimasi stokastik diantaranya adalah *Adagrad*, *RMSprop*, *SGD Nesterov*, *Adadelta*, dan *Adam*. Menurut penelitian (Kingma & Ba, 2015) optimasi adam merupakan optimasi terbaik karena adam menggabungkan keunggulan dari metode optimasi populer sebelumnya yaitu *Adagrad* dengan kemampuan menangani data *sparse*, dan metode *RMSprop* dengan kemampuan menangani data *non-stasioner* dengan memori labih sedikit.

4.2 Perhitungan Manualisasi

Perhitungan metode CNN masukan berupa citra seperti Gambar 4.3 yang kemudian dikonversi kebentuk angka seperti Gambar 4.4 agar dapat diproses :

Langkah 1



0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0
0	0	2	1	2	0	0	0
0	0	2	0	2	0	0	0
0	0	2	2	2	0	0	0
0	0	2	0	2	0	0	0
0	0	0	0	0	0	0	0

Gambar 4.4 Contoh Citra Perhitungan Manual

Gambar 4.3 Contoh Citra Bentuk Angka

Inisialisasi nilai bobot dengan bilangan acak range -1 sampai dengan 1. Berikut merupakan inisialisasi semua bobot seperti pada Tabel 4.2 dan Tabel 4.3

Tabel 4.2 Manualisasi Inisialisasi filter

	H1	H2	H3	H4																																				
I1	<table border="1"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	<table border="1"> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> </table>	-1	0	1	-1	0	1	-1	0	1	<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	0	-1	-1	-1	<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	-1	-1	-1	0	0	0	1	1	1
1	0	-1																																						
1	0	-1																																						
1	0	-1																																						
-1	0	1																																						
-1	0	1																																						
-1	0	1																																						
1	1	1																																						
0	0	0																																						
-1	-1	-1																																						
-1	-1	-1																																						
0	0	0																																						
1	1	1																																						

Tabel 4.3 Manualisasi Inisialisasi bobot

	O1	O2
X1	-0,17	0,263
X2	0,311	0,424
X3	0,127	0,823
X4	0,99	0,691
X5	0,563	-0,81
X6	0,924	0,32
X7	-0,58	-0,62
X8	-0,91	0,611
X9	0,228	0,545
X10	-0,59	0,28
X11	0,893	-0,45
X12	0,76	0,696
X13	-0,13	-0,83
X14	0,923	0,325
X15	0,087	0,989
X16	0,678	0,817

Langkah 2

Hitung semua keluaran di unit tersembunyi (H_{-netj})

$$H_{-netk} = \sum_{k=1}^n x_k * w_{kl}$$

$$(x_k * w_{kl})_{ij} = \sum_m \sum_n x_{i+m, j+n} \cdot w_{m,n}$$

$$H_{-netk} = \begin{bmatrix} (x_k * w_{kl})_{11} & \cdots & (x_k * w_{kl})_{1j} \\ \vdots & \ddots & \vdots \\ (x_k * w_{kl})_{i1} & \cdots & (x_k * w_{kl})_{ij} \end{bmatrix}$$

$$\begin{aligned} H_{-net1_{11}} &= \sum_{i=1}^1 x_{i+m, j+n} * w_{mn} = I_1 * w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+0, 0+0, -1+0, 1+0, 0+2, -1 \\ &= -2 \end{aligned}$$

$$\begin{aligned} H_{-net1_{12}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+0, 0+2, -1+0, 1+2, 0+1, -1 \\ &= -3 \end{aligned}$$

$$\begin{aligned} H_{-net1_{13}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+2, 0+0, -1+2, 1+1, 0+2, -1 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_{-net1_{14}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+2, 1+0, 0+0, -1+1, 1+2, 0+0, -1 \\ &= 3 \end{aligned}$$

$$\begin{aligned} H_{-net1_{15}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+0, 0+0, -1+2, 1+0, 0+0, -1 \\ &= 2 \end{aligned}$$

$$\begin{aligned} H_{-net1_{21}} &= \sum_{i=1}^1 x_{i+m, j+n} * w_{mn} = I_1 * w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+0, 0+2, -1+0, 1+0, 0+2, -1 \\ &= -4 \end{aligned}$$

$$\begin{aligned} H_{-net1_{22}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+0, 1+2, 0+2, -1+0, 1+2, 0+1, -1 \end{aligned}$$



$$= -3$$

$$\begin{aligned} H_{-net123} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 0, 1+2, 0+0, -1+2, 1+1, 0+2, -1+2, 1+1, 0+2, -1 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_{-net124} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 2, 1+0, 0+0, -1+1, 1+2, 0+0, -1+0, 1+2, 0+0, -1 \\ &= 3 \end{aligned}$$

$$\begin{aligned} H_{-net125} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 0, 1+0, 0+0, -1+2, 1+0, 0+0, -1+2, 1+0, 0+0, -1 \\ &= 4 \end{aligned}$$

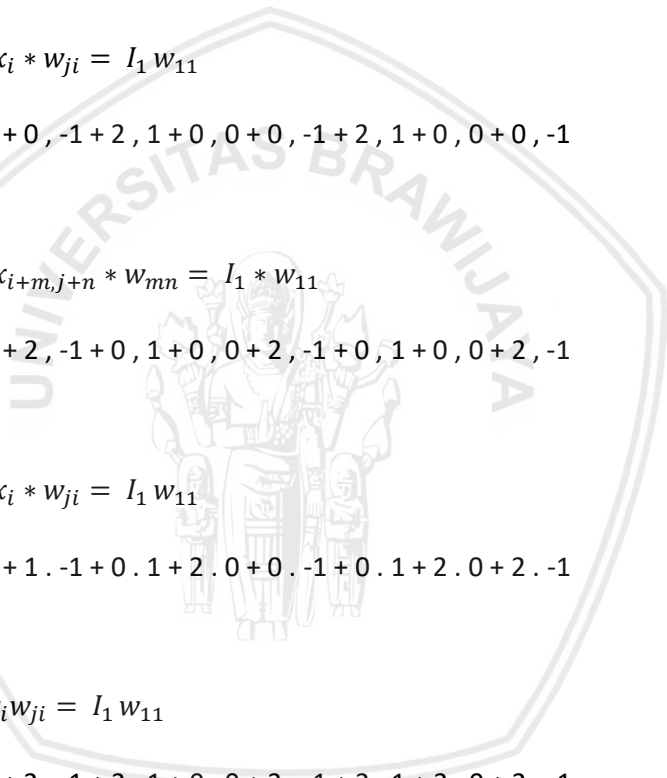
$$\begin{aligned} H_{-net131} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\ &= 0, 1+0, 0+2, -1+0, 1+0, 0+2, -1+0, 1+0, 0+2, -1 \\ &= -6 \end{aligned}$$

$$\begin{aligned} H_{-net132} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 0, 1+2, 0+1, -1+0, 1+2, 0+0, -1+0, 1+2, 0+2, -1 \\ &= -3 \end{aligned}$$

$$\begin{aligned} H_{-net133} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 2, 1+1, 0+2, -1+2, 1+0, 0+2, -1+2, 1+2, 0+2, -1 \\ &= 0 \end{aligned}$$

$$\begin{aligned} H_{-net134} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\ &= 1, 1+2, 0+0, -1+0, 1+2, 0+0, -1+2, 1+2, 0+0, -1 \\ &= 3 \end{aligned}$$

$$\begin{aligned} H_{-net135} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\ &= 2, 1+0, 0+0, -1+2, 1+0, 0+0, -1+2, 1+0, 0+0, -1 \\ &= 6 \end{aligned}$$



$$\begin{aligned}
 H_{-net141} &= \sum_{i=1}^1 x_{i+m,j+n} * W_{mn} = I_1 * W_{11} \\
 &= 0, 1+0, 0+2, -1+0, 1+0, 0+2, -1+0, 1+0, 0+2, -1 \\
 &= -6
 \end{aligned}$$

$$\begin{aligned}
 H_{-net142} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0, 1+2, 0+0, -1+0, 1+2, 0+2, -1+0, 1+2, 0+0, -1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{-net143} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2, 1+0, 0+2, -1+2, 1+2, 0+2, -1+2, 1+0, 0+2, -1 \\
 &= 0
 \end{aligned}$$

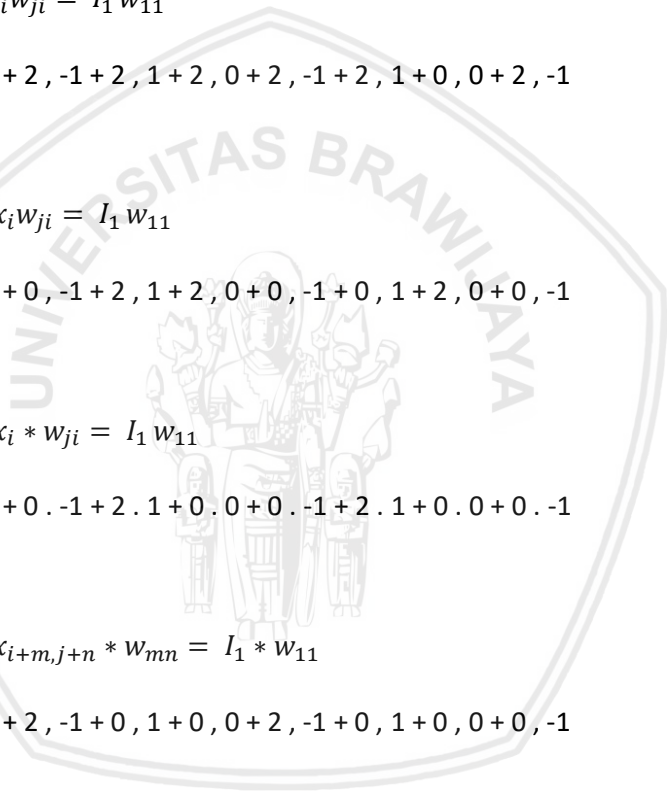
$$\begin{aligned}
 H_{-net144} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0, 1+2, 0+0, -1+2, 1+2, 0+0, -1+0, 1+2, 0+0, -1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{-net145} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2.1+0.0+0.-1+2.1+0.0+0.-1+2.1+0.0+0.-1 \\
 &= 6
 \end{aligned}$$

$$\begin{aligned}
 H_{-net151} &= \sum_{i=1}^1 x_{i+m,j+n} * W_{mn} = I_1 * W_{11} \\
 &= 0, 1+0, 0+2, -1+0, 1+0, 0+2, -1+0, 1+0, 0+0, -1 \\
 &= -4
 \end{aligned}$$

$$\begin{aligned}
 H_{-net152} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1+2.0+2.-1+0.1+2.0+0.-1+0.1+0.0+0.-1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{-net153} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1+2.0+2.-1+2.1+0.0+2.-1+0.1+0.0+0.-1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{net154}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 2.0 + 0.1 + 0.1 + 2.0 + 0.1 + 0.1 + 0.0 + 0.1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{net155}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.0 + 0.1 + 2.1 + 0.0 + 0.1 + 0.1 + 0.0 + 0.1 \\
 &= 4
 \end{aligned}$$

$$H_{\text{netk1}} = \begin{bmatrix} -2 & -3 & 0 & 3 & 2 \\ -4 & -3 & 0 & 3 & 4 \\ -6 & -3 & 0 & 3 & 6 \\ -6 & -2 & 0 & 2 & 6 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix}$$

$$\begin{aligned}
 H_{\text{net211}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.1 + 0.0 + 0.1 + 0.1 + 0.0 + 0.1 + 0.1 + 0.0 + 2.1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{net212}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.0 + 0.1 + 0.1 + 0.0 + 2.1 + 0.1 + 2.0 + 1.1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{net213}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.0 + 0.1 + 0.1 + 2.0 + 0.1 + 2.1 + 1.0 + 2.1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{net214}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.0 + 0.1 + 2.1 + 0.0 + 0.1 + 1.1 + 2.0 + 0.1 \\
 &= -3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{net215}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.0 + 0.1 + 0.1 + 0.0 + 0.1 + 2.1 + 0.0 + 0.1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{21}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{22}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 1 \cdot 1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{23}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 1 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 1 \cdot 0 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$

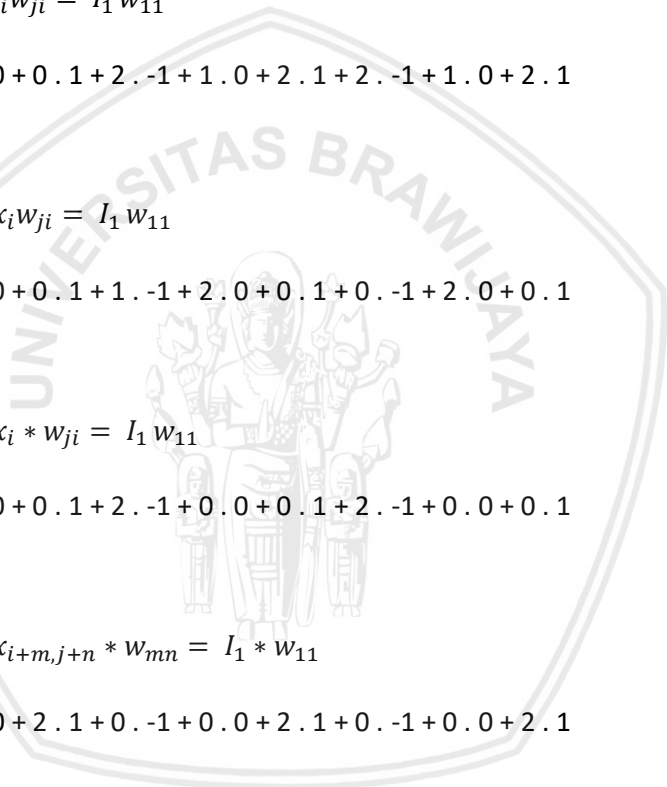
$$\begin{aligned}
 H_{\text{-net}2_{24}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 \\
 &= -3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{25}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= -4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{31}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 \\
 &= 6
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{32}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot 0 + 1 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net}2_{33}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 1 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net2}_{34}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 1 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 \\
 &= -3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{35}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= -6
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{41}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 \\
 &= 6
 \end{aligned}$$

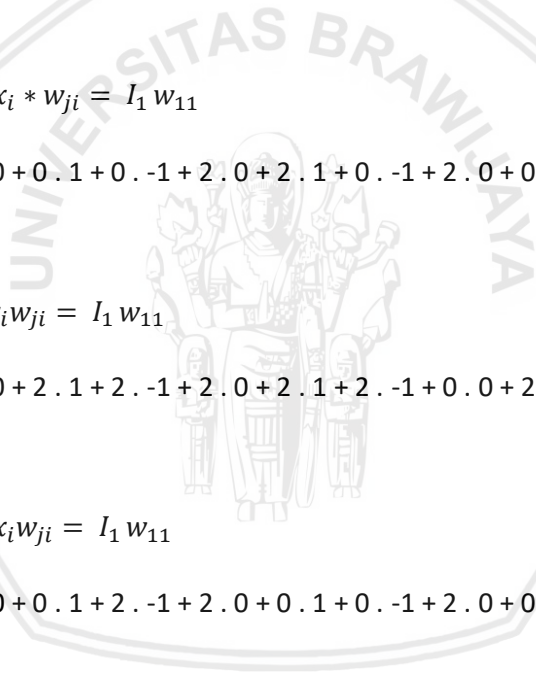
$$\begin{aligned}
 H_{\text{-net2}_{42}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{43}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{44}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{45}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= -6
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net2}_{51}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= 4
 \end{aligned}$$

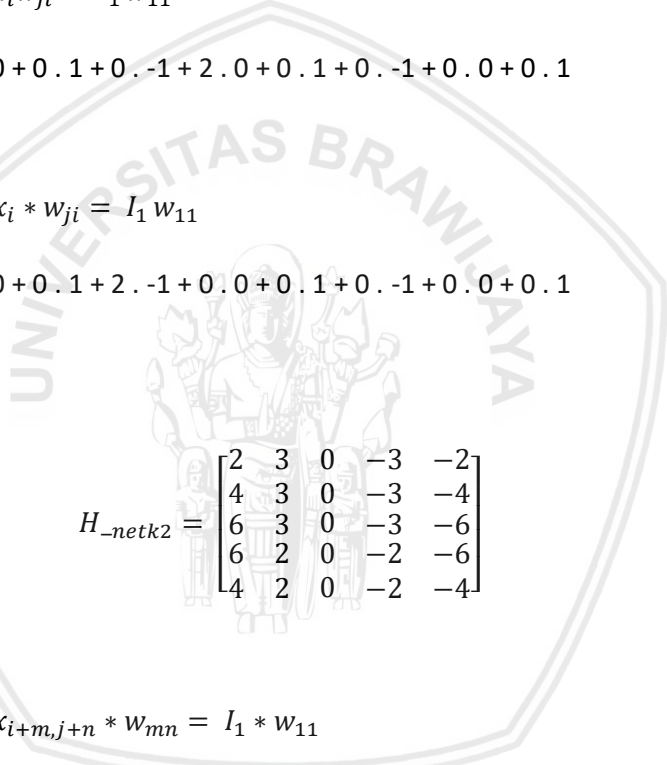


$$\begin{aligned}
 H_{\text{-net252}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net253}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 2 \cdot 0 + 2 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net254}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net255}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
 &= -4
 \end{aligned}$$



$$H_{\text{-netk2}} = \begin{bmatrix} 2 & 3 & 0 & -3 & -2 \\ 4 & 3 & 0 & -3 & -4 \\ 6 & 3 & 0 & -3 & -6 \\ 6 & 2 & 0 & -2 & -6 \\ 4 & 2 & 0 & -2 & -4 \end{bmatrix}$$

$$\begin{aligned}
 H_{\text{-net311}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net312}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 1 \cdot -1 \\
 &= -3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net313}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 2 \cdot -1 + 1 \cdot -1 + 2 \cdot -1 \\
 &= -5
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net314}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.1 + 0.1 + 2.0 + 0.0 + 0.0 + 1. \cdot -1 + 2. \cdot -1 + 0. \cdot -1 \\
 &= -3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net315}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.1 + 0.1 + 0.0 + 0.0 + 0.0 + 2. \cdot -1 + 0. \cdot -1 + 0. \cdot -1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net321}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.1 + 0.1 + 0.1 + 0.0 + 0.0 + 2.0 + 0. \cdot -1 + 0. \cdot -1 + 2. \cdot -1 \\
 &= -2
 \end{aligned}$$

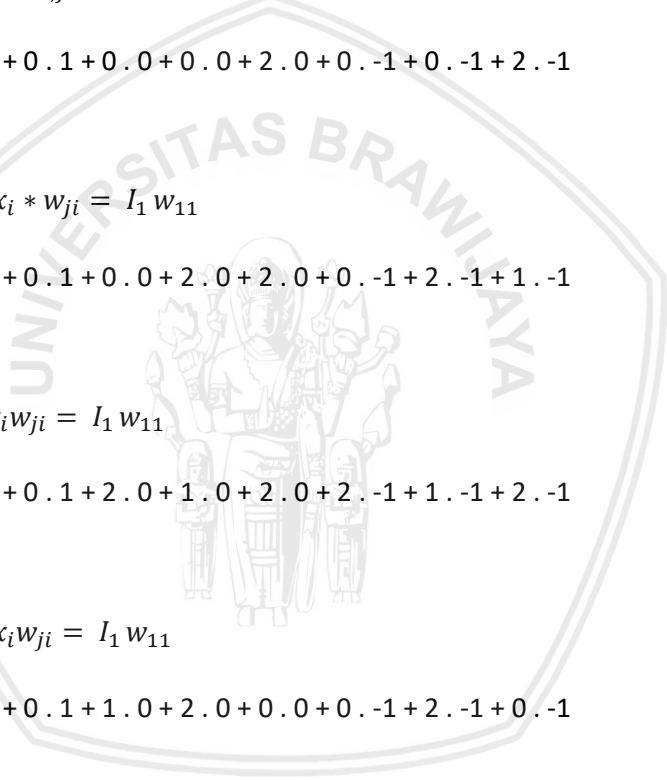
$$\begin{aligned}
 H_{\text{-net322}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.1 + 0.1 + 0.0 + 2.0 + 2.0 + 0. \cdot -1 + 2. \cdot -1 + 1. \cdot -1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net323}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0.1 + 2.1 + 0.1 + 2.0 + 1.0 + 2.0 + 2. \cdot -1 + 1. \cdot -1 + 2. \cdot -1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net324}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.1 + 0.1 + 1.0 + 2.0 + 0.0 + 0. \cdot -1 + 2. \cdot -1 + 0. \cdot -1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net325}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 0.1 + 0.1 + 2.0 + 0.0 + 0.0 + 2. \cdot -1 + 0. \cdot -1 + 0. \cdot -1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net331}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.1 + 0.1 + 2.1 + 0.0 + 0.0 + 2.0 + 0. \cdot -1 + 0. \cdot -1 + 2. \cdot -1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net332}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 2.1 + 1.1 + 0.0 + 2.0 + 0.0 + 0.0 + 0.1 + 2.1 + 2.1 \\
 &= -1
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net333}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 1.1 + 2.1 + 2.0 + 0.0 + 2.0 + 2.1 + 2.1 + 2.1 \\
 &= -1
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net334}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 1.1 + 2.1 + 0.1 + 0.0 + 2.0 + 0.0 + 2.1 + 2.1 + 0.1 \\
 &= -1
 \end{aligned}$$

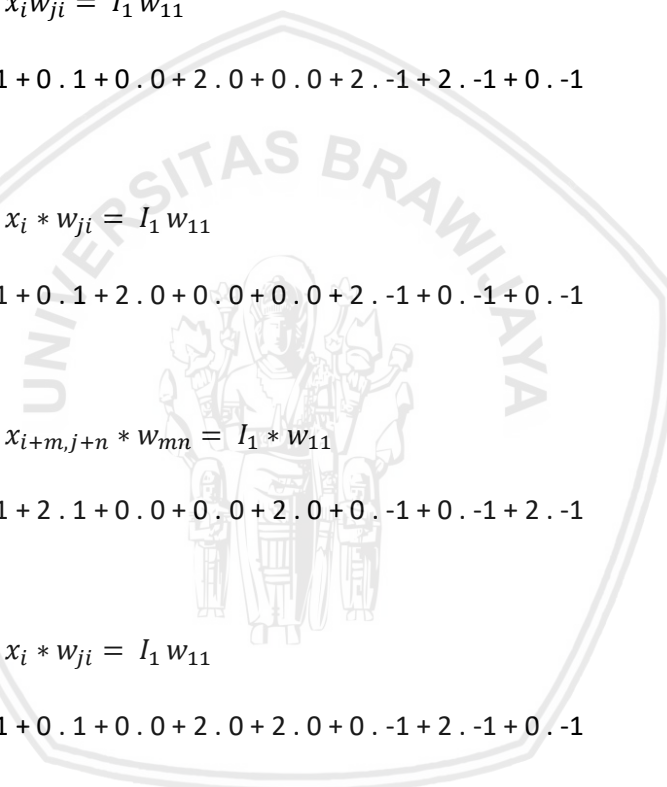
$$\begin{aligned}
 H_{\text{-net335}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.1 + 0.1 + 2.0 + 0.0 + 0.0 + 2.1 + 0.1 + 0.1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net341}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.1 + 0.1 + 2.1 + 0.0 + 0.0 + 2.0 + 0.1 + 0.1 + 2.1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net342}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 2.1 + 0.1 + 0.0 + 2.0 + 2.0 + 0.1 + 2.1 + 0.1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net343}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.1 + 2.1 + 2.0 + 2.0 + 2.0 + 2.1 + 0.1 + 2.1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net344}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0.1 + 2.1 + 0.1 + 2.0 + 2.0 + 0.0 + 0.1 + 2.1 + 0.1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net3}_{45}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.1 + 0.1 + 2.0 + 0.0 + 0.0 + 2.-1 + 0.-1 + 0.-1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net3}_{51}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.1 + 0.1 + 2.1 + 0.0 + 0.0 + 2.0 + 0.-1 + 0.-1 + 0.-1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net3}_{52}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0.1 + 2.1 + 2.1 + 0.0 + 2.0 + 0.0 + 0.-1 + 0.-1 + 0.-1 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net3}_{53}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 2.1 + 2.1 + 2.0 + 0.0 + 2.0 + 0.-1 + 0.-1 + 0.-1 \\
 &= 6
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net3}_{54}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2.1 + 2.1 + 0.1 + 0.0 + 2.0 + 0.0 + 0.-1 + 0.-1 + 0.-1 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net3}_{55}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2.1 + 0.1 + 0.1 + 2.0 + 0.0 + 0.0 + 0.1 + 0.1 + 0.1 \\
 &= 2
 \end{aligned}$$

$$H_{\text{-netk3}} = \begin{bmatrix} -2 & -3 & -5 & -3 & -2 \\ -2 & 0 & -2 & 0 & -2 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 6 & 4 & 2 \end{bmatrix}$$

$$\begin{aligned}
 H_{\text{-net4}_{11}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0.-1 + 0.-1 + 0.-1 + 0.0 + 0.0 + 0.0 + 0.1 + 0.1 + 2.1 \\
 &= 2
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net4}_{12}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 1 \cdot -1 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{13}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 \\
 &= 5
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{14}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 2 \cdot 1 + 0 \cdot 1 \\
 &= 3
 \end{aligned}$$

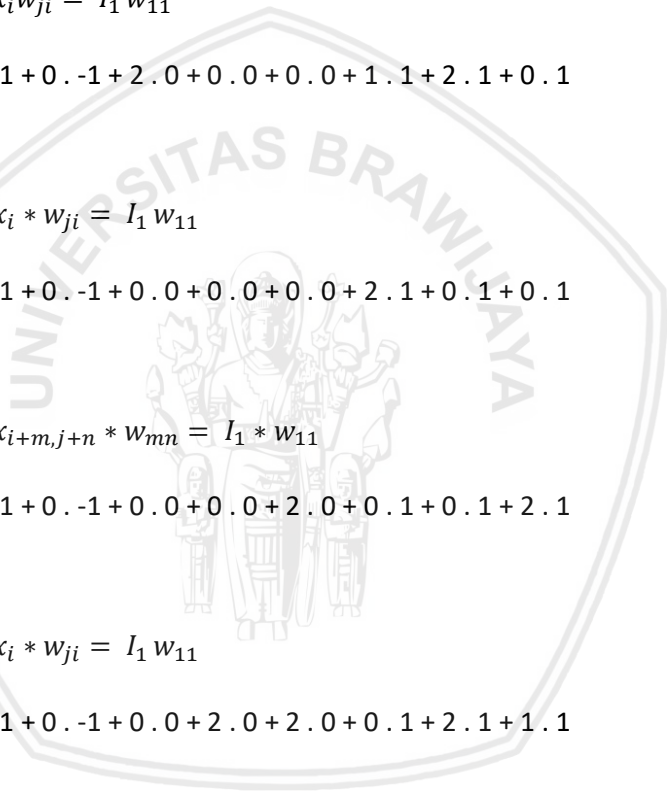
$$\begin{aligned}
 H_{\text{-net4}_{15}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{21}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 2 \cdot 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{22}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 2 \cdot 1 + 1 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{23}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 2 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{24}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot 1 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net4}_{25}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= 2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{31}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{32}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot -1 + 1 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot 1 + 2 \cdot 1 \\
 &= 1
 \end{aligned}$$

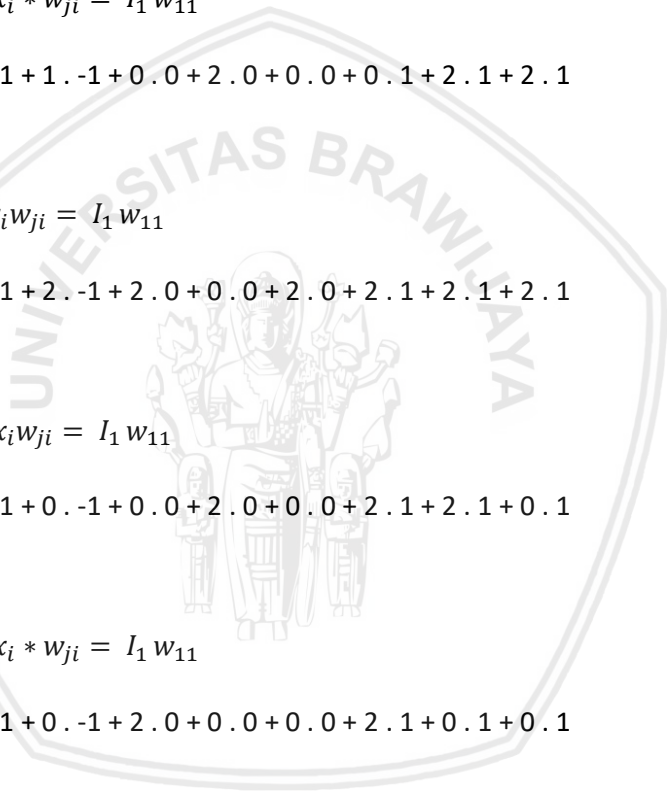
$$\begin{aligned}
 H_{\text{-net4}_{33}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 1 \cdot -1 + 2 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 2 \cdot 1 + 2 \cdot 1 + 2 \cdot 1 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{34}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 1 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 2 \cdot 1 + 0 \cdot 1 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{35}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{41}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{42}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 2 \cdot 1 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 H_{\text{-net4}_{43}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 2 \cdot 0 + 2 \cdot 0 + 2 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 2 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{44}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 2 \cdot 1 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{45}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= 0
 \end{aligned}$$

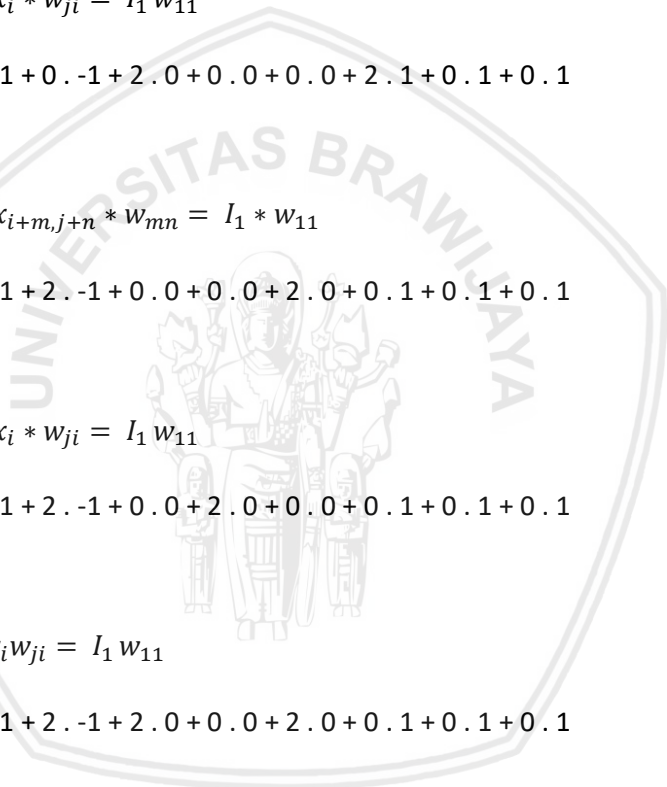
$$\begin{aligned}
 H_{\text{-net4}_{51}} &= \sum_{i=1}^1 x_{i+m,j+n} * w_{mn} = I_1 * w_{11} \\
 &= 0 \cdot -1 + 0 \cdot -1 + 2 \cdot -1 + 0 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= -2
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{52}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 0 \cdot -1 + 2 \cdot -1 + 2 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= -4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{53}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 2 \cdot -1 + 2 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= -6
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{54}} &= \sum_{i=1}^1 x_i w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 2 \cdot -1 + 0 \cdot -1 + 0 \cdot 0 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= -4
 \end{aligned}$$

$$\begin{aligned}
 H_{\text{-net4}_{55}} &= \sum_{i=1}^1 x_i * w_{ji} = I_1 w_{11} \\
 &= 2 \cdot -1 + 0 \cdot -1 + 0 \cdot -1 + 2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 \\
 &= -2
 \end{aligned}$$



$$H_{net4} = \begin{bmatrix} 2 & 3 & 5 & 3 & 2 \\ 2 & 0 & 2 & 0 & 2 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4 & -6 & -4 & -2 \end{bmatrix}$$

Langkah 3

Menambahkan hasil dengan bias

Bias = b = 1.

$$H_{net1} = \begin{bmatrix} -2 & -3 & 0 & 3 & 2 \\ -4 & -3 & 0 & 3 & 4 \\ -6 & -3 & 0 & 3 & 6 \\ -6 & -2 & 0 & 2 & 6 \\ -4 & -2 & 0 & 2 & 4 \end{bmatrix} + b = \begin{bmatrix} -1 & -2 & 1 & 4 & 3 \\ -3 & -2 & 1 & 4 & 5 \\ -5 & -2 & 1 & 4 & 7 \\ -5 & -1 & 1 & 3 & 7 \\ -3 & -1 & 1 & 3 & 5 \end{bmatrix}$$

$$H_{net2} = \begin{bmatrix} 2 & 3 & 0 & -3 & -2 \\ 4 & 3 & 0 & -3 & -4 \\ 6 & 3 & 0 & -3 & -6 \\ 6 & 2 & 0 & -2 & -6 \\ 4 & 2 & 0 & -2 & -4 \end{bmatrix} + b = \begin{bmatrix} 3 & 4 & 1 & -2 & -1 \\ 5 & 4 & 1 & -2 & -3 \\ 7 & 4 & 1 & -2 & -5 \\ 7 & 3 & 1 & -1 & -5 \\ 5 & 3 & 1 & -1 & -3 \end{bmatrix}$$

$$H_{net3} = \begin{bmatrix} -2 & -3 & -5 & -3 & -2 \\ -2 & 0 & -2 & 0 & -2 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 6 & 4 & 2 \end{bmatrix} + b = \begin{bmatrix} -1 & -2 & -4 & -2 & -1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 5 & 7 & 5 & 3 \end{bmatrix}$$

$$H_{net4} = \begin{bmatrix} 2 & 3 & 5 & 3 & 2 \\ 2 & 0 & 2 & 0 & 2 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -4 & -6 & -4 & -2 \end{bmatrix} + b = \begin{bmatrix} 3 & 4 & 6 & 4 & 3 \\ 3 & 1 & 3 & 1 & 3 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ -1 & -3 & -5 & -3 & -1 \end{bmatrix}$$

Langkah 4

Hitung semua keluaran di unit tersembunyi dengan fungsi *ReLU* (H_j)

$$H_j = f(H_{netj}) = \begin{cases} f(x)=x & \text{jika } x \geq 0 \\ f(x)=0 & \text{jika } x < 0 \end{cases}$$

$$H_1 = f(H_{net1}) = f \left(\begin{bmatrix} -1 & -2 & 1 & 4 & 3 \\ -3 & -2 & 1 & 4 & 5 \\ -5 & -2 & 1 & 4 & 7 \\ -5 & -1 & 1 & 3 & 7 \\ -3 & -1 & 1 & 3 & 5 \end{bmatrix} \right) = \begin{bmatrix} 0 & 0 & 1 & 4 & 3 \\ 0 & 0 & 1 & 4 & 5 \\ 0 & 0 & 1 & 4 & 7 \\ 0 & 0 & 1 & 3 & 7 \\ 0 & 0 & 1 & 3 & 5 \end{bmatrix}$$



$$H_2 = f(H_{net2}) = f\left(\begin{bmatrix} 3 & 4 & 1 & -2 & -1 \\ 5 & 4 & 1 & -2 & -3 \\ 7 & 4 & 1 & -2 & -5 \\ 7 & 3 & 1 & -1 & -5 \\ 5 & 3 & 1 & -1 & -3 \end{bmatrix}\right) = \begin{bmatrix} 3 & 4 & 1 & 0 & 0 \\ 5 & 4 & 1 & 0 & 0 \\ 7 & 4 & 1 & 0 & 0 \\ 7 & 3 & 1 & 0 & 0 \\ 5 & 3 & 1 & 0 & 0 \end{bmatrix}$$

$$H_3 = f(H_{net3}) = f\left(\begin{bmatrix} -1 & -2 & -4 & -2 & -1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 5 & 7 & 5 & 3 \end{bmatrix}\right) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 3 & 5 & 7 & 5 & 3 \end{bmatrix}$$

$$H_4 = f(H_{net4}) = f\left(\begin{bmatrix} 3 & 4 & 6 & 4 & 3 \\ 3 & 1 & 3 & 1 & 3 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ -1 & -3 & -5 & -3 & -1 \end{bmatrix}\right) = \begin{bmatrix} 3 & 4 & 6 & 4 & 3 \\ 3 & 1 & 3 & 1 & 3 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Langkah 5

Menyederhanakan keluaran dengan *max_pooling*

$$P_{j,m,n} = \max((H_{j,m,n}), (H_{j,m+1,n}), (H_{j,m,n+1}), (H_{j,m+1,n+1}))$$

m = panjang baris matriks atau gambar dari H_j yang dikurangi *stride*

n = panjang kolom matriks atau gambar dari H_j yang dikurangi *stride*

Stride = pergeseran untuk *pooling*

Dalam contoh :

$$m = 5 - 1 = 4$$

$$n = 5 - 1 = 4$$

$$P_{1,1,1} = \max(H_{1,1,1}, H_{1,2,1}, H_{1,1,2}, H_{1,2,2})$$

$$P_{1,1,1} = \max(0, 0, 0, 0) = 0$$

$$P_{1,1,2} = \max(0, 1, 0, 1) = 1$$

$$P_{1,1,3} = \max(1, 4, 1, 4) = 4$$

$$P_{1,1,4} = \max(4, 3, 4, 5) = 5$$

$$P_{1,2,1} = \max(0, 0, 0, 0) = 0$$

$$P_{1,2,2} = \max(0, 1, 0, 1) = 1$$

$$P_{1,2,3} = \max(1, 4, 1, 4) = 4$$

$$P_{1,2,4} = \max(4, 5, 4, 7) = 7$$

$$P_{1,3,1} = \max(0, 0, 0, 0) = 0$$

$$P_{1,3,2} = \max(0, 1, 0, 1) = 1$$

$$P_{1,3,3} = \max(1, 4, 1, 3) = 4$$

$$P_{1,3,4} = \max(4, 7, 3, 7) = 7$$

$$P_{1,4,1} = \max(0, 0, 0, 0) = 0$$

$$P_{1,4,2} = \max(0, 1, 0, 1) = 1$$

$$P_{1,4,3} = \max(1, 3, 1, 3) = 3$$

$$P_{1,4,4} = \max(3, 7, 3, 5) = 7$$



$$P_1 = \begin{bmatrix} 0 & 1 & 4 & 5 \\ 0 & 1 & 4 & 7 \\ 0 & 1 & 4 & 7 \\ 0 & 1 & 3 & 7 \end{bmatrix}$$

$$P_{2_{1,1}} = \max(3, 4, 5, 4) = 5$$

$$P_{2_{1,2}} = \max(4, 1, 4, 1) = 4$$

$$P_{2_{1,3}} = \max(1, 0, 1, 0) = 1$$

$$P_{2_{1,4}} = \max(0, 0, 0, 0) = 0$$

$$P_{2_{2,1}} = \max(5, 4, 7, 4) = 7$$

$$P_{2_{2,2}} = \max(4, 1, 4, 1) = 4$$

$$P_{2_{2,3}} = \max(1, 0, 1, 0) = 1$$

$$P_{2_{2,4}} = \max(0, 0, 0, 0) = 0$$

$$P_{2_{3,1}} = \max(7, 4, 7, 3) = 7$$

$$P_{2_{3,2}} = \max(4, 1, 3, 1) = 4$$

$$P_{2_{3,3}} = \max(1, 0, 1, 0) = 1$$

$$P_{2_{3,4}} = \max(0, 0, 0, 0) = 0$$

$$P_{2_{4,1}} = \max(7, 3, 5, 3) = 7$$

$$P_{2_{4,2}} = \max(3, 1, 3, 1) = 3$$

$$P_{2_{4,3}} = \max(1, 0, 1, 0) = 1$$

$$P_{2_{4,4}} = \max(0, 0, 0, 0) = 0$$

$$P_2 = \begin{bmatrix} 5 & 4 & 1 & 0 \\ 7 & 4 & 1 & 0 \\ 7 & 4 & 1 & 0 \\ 7 & 3 & 1 & 0 \end{bmatrix}$$

$$P_{3_{1,1}} = \max(0, 0, 0, 0) = 0$$

$$P_{3_{1,2}} = \max(0, 0, 0, 0) = 0$$

$$P_{3_{1,3}} = \max(0, 0, 0, 1) = 1$$

$$P_{3_{1,4}} = \max(0, 0, 1, 0) = 1$$

$$P_{3_{2,1}} = \max(0, 0, 1, 0) = 1$$

$$P_{3_{2,2}} = \max(0, 0, 0, 0) = 0$$

$$P_{3_{2,3}} = \max(0, 1, 0, 0) = 1$$

$$P_{3_{2,4}} = \max(1, 0, 0, 1) = 1$$

$$P_{3_{3,1}} = \max(1, 0, 1, 1) = 1$$

$$P_{3_{3,2}} = \max(0, 0, 1, 1) = 1$$

$$P_{3_{3,3}} = \max(0, 0, 1, 1) = 1$$

$$P_{3_{3,4}} = \max(0, 1, 1, 1) = 1$$

$$P_{3_{4,1}} = \max(1, 1, 3, 5) = 5$$

$$P_{3_{4,2}} = \max(1, 1, 5, 7) = 7$$

$$P_{3_{4,3}} = \max(1, 1, 7, 5) = 7$$

$$P_{3_{4,4}} = \max(1, 1, 5, 3) = 5$$

$$P_3 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 5 & 7 & 7 & 5 \end{bmatrix}$$

$$P_{4_{1,1}} = \max(3, 4, 3, 1) = 4$$

$$P_{4_{1,2}} = \max(4, 6, 1, 3) = 6$$

$$P_{4_{1,3}} = \max(6, 4, 3, 1) = 6$$

$$P_{4_{1,4}} = \max(4, 3, 1, 3) = 4$$

$$P_{4_{2,1}} = \max(3, 1, 1, 2) = 3$$

$$P_{4_{2,2}} = \max(1, 3, 2, 2) = 3$$

$$P_{4_{2,3}} = \max(3, 1, 2, 2) = 3$$

$$P_{4_{3,1}} = \max(1, 2, 1, 1) = 2$$

$$P_{4_{3,2}} = \max(2, 2, 1, 1) = 2$$

$$P_{4_{3,3}} = \max(2, 2, 1, 1) = 2$$

$$P_{4_{3,4}} = \max(2, 1, 1, 1) = 2$$

$$P_{4_{4,1}} = \max(1, 1, 0, 0) = 1$$

$$P_{4_{4,2}} = \max(1, 1, 0, 0) = 1$$

$$P_{4_{4,3}} = \max(1, 1, 0, 0) = 1$$



$$P_{4_2,4} = \max(1, 3, 2, 1) = 3$$

$$P_{4_4,4} = \max(1, 1, 0, 0) = 1$$

$$P_4 = \begin{bmatrix} 4 & 6 & 6 & 4 \\ 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Langkah 6

Memecah matriks konvolusi menjadi satuan individu dengan cara *flattening*($f(x)$)

$$x_{(1, 2, \dots, n)} = f(P_1 + P_2 + P_3 + P_4)$$

$$P_1 + P_2 + P_3 + P_4 = \begin{bmatrix} 10 & 12 & 12 & 10 \\ 11 & 9 & 9 & 11 \\ 10 & 8 & 8 & 10 \\ 13 & 1 & 3 & 7 \end{bmatrix}$$

$x_1 = 10$	$x_9 = 10$
$x_2 = 12$	$x_{10} = 8$
$x_3 = 12$	$x_{11} = 8$
$x_4 = 10$	$x_{12} = 10$
$x_5 = 11$	$x_{13} = 13$
$x_6 = 9$	$x_{14} = 1$
$x_7 = 9$	$x_{15} = 3$
$x_8 = 11$	$x_{16} = 7$

Langkah 7

Hitung semua keluaran di unit tersembunyi (O_{-netj})

$$O_{-netj} = \sum_{i=1}^n x_i w_{ji}$$

$$O_{-net1} = \sum_{i=1}^{16} x_i w_{ji}$$

$$= I_1 w_{11} + I_2 w_{21} + I_3 w_{31} + I_4 w_{41} + I_5 w_{51} + I_6 w_{61} + I_7 w_{71} + I_8 w_{81} \\ + I_9 w_{91} + I_{10} w_{101} + I_{11} w_{111} + I_{12} w_{121} + I_{13} w_{131} + I_{14} w_{141} \\ + I_{15} w_{151} + I_{16} w_{161}$$

$$= 10 \cdot -0,17 + 12 \cdot 0,31 + 12 \cdot 0,13 + 10 \cdot 0,99 + 11 \cdot 0,56 + 9 \cdot 0,92 + 9 \cdot -0,576 + \\ 11 \cdot -0,91 + 10 \cdot 0,23 + 8 \cdot -0,59 + 8 \cdot 0,893 + 10 \cdot 0,76 + 13 \cdot -0,134 + 1 \cdot 0,92 \\ + 3 \cdot 0,087 + 7 \cdot 0,68$$

$$= 29,31$$



$$\begin{aligned}
 O_{-net2} &= \sum_{i=1}^{16} x_i w_{ji} \\
 &= I_1 w_{12} + I_2 w_{22} + I_3 w_{32} + I_4 w_{42} + I_5 w_{52} + I_6 w_{62} + I_7 w_{72} + I_8 w_{82} + \\
 &\quad I_9 w_{92} + I_{10} w_{102} + I_{11} w_{112} + I_{12} w_{122} + I_{13} w_{132} + I_{14} w_{142} + \\
 &\quad I_{15} w_{152} + I_{16} w_{162} \\
 &= 10 \cdot 0.26 + 12 \cdot 0.42 + 12 \cdot 0.82 + 10 \cdot 0.69 + 11 \cdot -0.81 + 9 \cdot 0.32 + 9 \cdot -0.62 + 11 \\
 &\quad \cdot 0.61 + 10 \cdot 0.55 + 8 \cdot 0.28 + 8 \cdot -0.45 + 10 \cdot 0.696 + 13 \cdot -0.83 + 1 \cdot 0.33 + 3 \cdot \\
 &\quad 0.99 + 7 \cdot 0.82 \\
 &= 28.96
 \end{aligned}$$

Langkah 8

Hitung semua keluaran di unit keluaran dengan fungsi *Softmax sigmoid* (O_k)

$$O_k = \frac{e^{O_{-netk}}}{\sum e^{O_{-neti}}}$$

$$O_1 = \frac{e^{O_{-net1}}}{\sum e^{O_{-neti}}} = \frac{e^{29,31}}{e^{29,31} + e^{28,96}} = 0,5855$$

$$O_2 = \frac{e^{O_{-net2}}}{\sum e^{O_{-neti}}} = \frac{e^{28,96}}{e^{29,31} + e^{28,96}} = 0,4145$$

Langkah 9

Fase propagasi mundur

$$\delta_k = (T_k - y_k)(y_k - (T_k - y_k))$$

$$\begin{aligned}
 \delta_1 &= (T_1 - y_1)(y_1 - (T_1 - y_1)) \\
 &= (1 - 0,5855)(0,5855 - (1 - 0,5855)) \\
 &= -0.071
 \end{aligned}$$

$$\begin{aligned}
 \delta_2 &= (T_2 - y_2)(y_2 - (T_2 - y_2)) \\
 &= (0 - 0,4145)(0,4145 - (0 - 0,4145)) \\
 &= 0.344
 \end{aligned}$$

$$\Delta w_{Njk} = \alpha \delta_k H_j$$

$$\Delta w_{N11} = w_{N11} - \alpha \delta_1 \cdot x_1 = 0,005 \cdot -0.071 \cdot 10 = -0.162675868$$

$$\Delta w_{N12} = w_{N12} - \alpha \delta_1 \cdot x_2 = 0,005 \cdot -0.071 \cdot 12 = 0.315573667$$

$$\begin{aligned}
\Delta w_{N13} &= w_{N13} - \alpha \delta_1 \cdot x_3 = 0,005 \cdot -0,071 \cdot 12 = 0.13151074 \\
\Delta w_{N14} &= w_{N14} - \alpha \delta_1 \cdot x_4 = 0,005 \cdot -0,071 \cdot 10 = 0.993813694 \\
\Delta w_{N15} &= w_{N15} - \alpha \delta_1 \cdot x_5 = 0,005 \cdot -0,071 \cdot 11 = 0.567267325 \\
\Delta w_{N16} &= w_{N16} - \alpha \delta_1 \cdot x_6 = 0,005 \cdot -0,071 \cdot 9 = 0.927319899 \\
\Delta w_{N17} &= w_{N17} - \alpha \delta_1 \cdot x_7 = 0,005 \cdot -0,071 \cdot 9 = -0.572903 \\
\Delta w_{N18} &= w_{N18} - \alpha \delta_1 \cdot x_8 = 0,005 \cdot -0,071 \cdot 11 = -0.909681604 \\
\Delta w_{N19} &= w_{N19} - \alpha \delta_1 \cdot x_9 = 0,005 \cdot -0,071 \cdot 10 = 0.231490364 \\
\Delta w_{N110} &= w_{N110} - \alpha \delta_1 \cdot x_{10} = 0,005 \cdot -0,071 \cdot 8 = -0.583536641 \\
\Delta w_{N111} &= w_{N111} - \alpha \delta_1 \cdot x_{11} = 0,005 \cdot -0,071 \cdot 8 = 0.896301679 \\
\Delta w_{N12} &= w_{N112} - \alpha \delta_1 \cdot x_{12} = 0,005 \cdot -0,071 \cdot 10 = 0.764019023 \\
\Delta w_{N13} &= \alpha w_{N113} - \delta_1 \cdot x_{13} = 0,005 \cdot -0,071 \cdot 13 = -0.129457688 \\
\Delta w_{N14} &= w_{N114} - \alpha \delta_1 \cdot x_{14} = 0,005 \cdot -0,071 \cdot 1 = 0.923062794 \\
\Delta w_{N15} &= w_{N115} - \alpha \delta_1 \cdot x_{15} = 0,005 \cdot -0,071 \cdot 3 = 0.087655697 \\
\Delta w_{N16} &= w_{N116} - \alpha \delta_1 \cdot x_{16} = 0,005 \cdot -0,071 \cdot 7 = 0.680132193 \\
\\
\Delta w_{N21} &= w_{N21} - \alpha \delta_1 \cdot x_1 = 0,005 \cdot 0,344 \cdot 10 = -0.162675868 \\
\Delta w_{N22} &= w_{N21} - \alpha \delta_1 \cdot x_2 = 0,005 \cdot 0,344 \cdot 12 = 0.315573667 \\
\Delta w_{N23} &= w_{N21} - \alpha \delta_1 \cdot x_3 = 0,005 \cdot 0,344 \cdot 12 = 0.13151074 \\
\Delta w_{N24} &= w_{N21} - \alpha \delta_1 \cdot x_4 = 0,005 \cdot 0,344 \cdot 10 = 0.993813694 \\
\Delta w_{N25} &= w_{N21} - \alpha \delta_1 \cdot x_5 = 0,005 \cdot 0,344 \cdot 11 = 0.567267325 \\
\Delta w_{N26} &= w_{N21} - \alpha \delta_1 \cdot x_6 = 0,005 \cdot 0,344 \cdot 9 = 0.927319899 \\
\Delta w_{N27} &= w_{N21} - \alpha \delta_1 \cdot x_7 = 0,005 \cdot 0,344 \cdot 9 = -0.572903 \\
\Delta w_{N28} &= w_{N21} - \alpha \delta_1 \cdot x_8 = 0,005 \cdot 0,344 \cdot 11 = -0.909681604 \\
\Delta w_{N29} &= w_{N21} - \alpha \delta_1 \cdot x_9 = 0,005 \cdot 0,344 \cdot 10 = 0.231490364 \\
\Delta w_{N210} &= w_{N21} - \alpha \delta_1 \cdot x_{10} = 0,005 \cdot 0,344 \cdot 8 = -0.583536641 \\
\Delta w_{N211} &= w_{N21} - \alpha \delta_1 \cdot x_{11} = 0,005 \cdot 0,344 \cdot 8 = 0.896301679 \\
\Delta w_{N212} &= w_{N21} - \alpha \delta_1 \cdot x_{12} = 0,005 \cdot 0,344 \cdot 10 = 0.764019023 \\
\Delta w_{N213} &= w_{N21} - \alpha \delta_1 \cdot x_{13} = 0,005 \cdot 0,344 \cdot 13 = -0.129457688 \\
\Delta w_{N214} &= w_{N21} - \alpha \delta_1 \cdot x_{14} = 0,005 \cdot 0,344 \cdot 1 = 0.923062794 \\
\Delta w_{N215} &= w_{N21} - \alpha \delta_1 \cdot x_{15} = 0,005 \cdot 0,344 \cdot 3 = 0.087655697 \\
\Delta w_{N216} &= w_{N21} - \alpha \delta_1 \cdot x_{16} = 0,005 \cdot 0,344 \cdot 7 = 0.680132193
\end{aligned}$$



Langkah 10

$$\delta_{flatk} = \sum \delta_k \cdot w_{Njk}$$

$$\delta_{flat1} = \delta_1 \cdot w_{N11} + \delta_2 \cdot w_{N21} = -0.071 \cdot -0.166 + 0,344 \cdot 0.26 = 0.102179077$$

$$\delta_{flat2} = \delta_1 \cdot w_{N12} + \delta_2 \cdot w_{N22} = -0.071 \cdot 0.31 + 0,344 \cdot 0.424 = 0.12344342$$

$$\delta_{flat3} = \delta_1 \cdot w_{N13} + \delta_2 \cdot w_{N23} = -0.071 \cdot 0.127 + 0,344 \cdot 0.82 = 0.273838775$$

$$\delta_{flat4} = \delta_1 \cdot w_{N14} + \delta_2 \cdot w_{N24} = -0.071 \cdot 0.99 + 0,344 \cdot 0.69 = 0.167133977$$

$$\delta_{flat5} = \delta_1 \cdot w_{N15} + \delta_2 \cdot w_{N25} = -0.071 \cdot 0.56 + 0,344 \cdot -0.81 = -0.31660113$$

$$\delta_{flat6} = \delta_1 \cdot w_{N16} + \delta_2 \cdot w_{N26} = -0.071 \cdot 0.92 + 0,344 \cdot 0.32 = 0.044440792$$

$$\delta_{flat7} = \delta_1 \cdot w_{N17} + \delta_2 \cdot w_{N27} = -0.071 \cdot 0.58 + 0,344 \cdot 0.62 = -0.171381277$$

$$\delta_{flat8} = \delta_1 \cdot w_{N18} + \delta_2 \cdot w_{N28} = -0.071 \cdot -0.91 + 0,344 \cdot 0.61 = 0.274772067$$

$$\delta_{flat9} = \delta_1 \cdot w_{N19} + \delta_2 \cdot w_{N29} = -0.071 \cdot 0.23 + 0,344 \cdot 0.55 = 0.171099884$$

$$\delta_{flat10} = \delta_1 \cdot w_{N110} + \delta_2 \cdot w_{N210} = -0.071 \cdot -0.59 + 0,344 \cdot 0.28 = 0.13780983$$

$$\delta_{flat11} = \delta_1 \cdot w_{N111} + \delta_2 \cdot w_{N211} = -0.071 \cdot 0.89 + 0,344 \cdot -0.45 = -0.2192350$$

$$\delta_{flat12} = \delta_1 \cdot w_{N112} + \delta_2 \cdot w_{N212} = -0.071 \cdot 0.76 + 0,344 \cdot 0.69 = 0.185096281$$

$$\delta_{flat13} = \delta_1 \cdot w_{N113} + \delta_2 \cdot w_{N213} = -0.071 \cdot -0.13 + 0,344 \cdot -0.827 = -0.27464$$

$$\delta_{flat14} = \delta_1 \cdot w_{N114} + \delta_2 \cdot w_{N214} = -0.071 \cdot 0.92 + 0,344 \cdot 0.325 = 0.0462878$$

$$\delta_{flat15} = \delta_1 \cdot w_{N115} + \delta_2 \cdot w_{N215} = -0.071 \cdot 0.086 + 0,344 \cdot 0.988 = 0.333478$$

$$\delta_{flat16} = \delta_1 \cdot w_{N116} + \delta_2 \cdot w_{N216} = -0.071 \cdot 0.68 + 0,344 \cdot 0.816 = 0.2325076$$

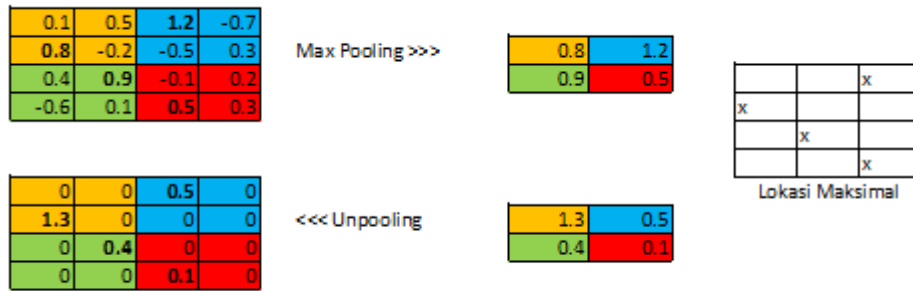
$\delta_{flat} =$

0.1022	0.1234	0.2738	0.1671
-0.317	0.0444	-0.171	0.2748
0.1711	0.1378	-0.219	0.1851
-0.275	0.0463	0.3335	0.2325

Langkah 11

Melakukan *unpooling* δ_{flatk} atau membuat gradient δ_{poolk} dari proses *pooling*

Pada waktu proses convolution ada empat neuron maka membuat gradient setiap neuronnya dari gradient proses *pooling* dengan rumus :



$$\delta_{pool1} =$$

δ_{flat11}		δ_{flat12}	δ_{flat13}	
δ_{flat21}		δ_{flat22}	δ_{flat23}	δ_{flat14}
δ_{flat31}		δ_{flat32}	δ_{flat33}	$(\delta_{flat24} + \delta_{flat34}) / 2$
δ_{flat41}		δ_{flat42}	δ_{flat43}	δ_{flat44}

$$\delta_{pool1} =$$

0.1022	0	0.1234	0.2738	0
-0.317	0	0.0444	-0.171	0.1671
0.1711	0	0.1378	-0.219	0.2299
-0.275	0	0.0463	0.3335	0.2325
0	0	0	0	0

$$\delta_{pool2} =$$

	δ_{flat12}	δ_{flat13}	δ_{flat14}	
δ_{flat11}	δ_{flat22}	δ_{flat23}	δ_{flat24}	
$(\delta_{flat21} + \delta_{flat31}) / 2$	δ_{flat32}	δ_{flat33}	δ_{flat34}	
δ_{flat41}	δ_{flat42}	δ_{flat43}	δ_{flat44}	

$$\delta_{pool2} =$$

0	0.1234	0.2738	0.1671	0
0.1022	0.0444	-0.171	0.2748	0
-0.073	0.1378	-0.219	0.1851	0
-0.275	0.0463	0.3335	0.2325	0
0	0	0	0	0

$$\delta_{pool3} =$$

	$(\delta_{flat11} + \delta_{flat22} + \delta_{flat12} + \delta_{flat21})/4$		$\delta_{flat13} + \delta_{flat14} + \delta_{flat23} + \delta_{flat24})/4$	
δ_{flat31}				
	δ_{flat32}	δ_{flat33}	δ_{flat34}	
	δ_{flat41}	$(\delta_{flat42} + \delta_{flat43})/2$	δ_{flat44}	

$\delta_{pool3} =$

0	0	0	0	0
0	-0.012	0	0.1361	0
0.1711	0	0	0	0
0	0.1378	-0.219	0.1851	0
0	-0.275	0.1899	0.2325	0

$\delta_{pool4} =$

	δ_{flat11}	$(\delta_{flat12} + \delta_{flat13})/2$	δ_{flat14}	
δ_{flat21}		$(\delta_{flat22} + \delta_{flat23})/2$		δ_{flat24}
	$(\delta_{flat31} + \delta_{flat32})/2$	δ_{flat33}	δ_{flat34}	
δ_{flat41}	δ_{flat42}	δ_{flat43}	δ_{flat44}	

$\delta_{pool4} =$

0	0.1022	0.1986	0.1671	0
-0.317	0	-0.063		0.2748
0	0.1545	-0.219	0.1851	0
-0.275	0.0463	0.3335	0.2325	0
0	0	0	0	0

Langkah 12

Mengkalikan gradient *pooling* dengan turunan dari fungsi aktivasi (ReLU Activation) untuk menjadi gradient konvolusi

$$\delta_{conv} = \delta_{pool} \cdot f'(x)$$

$f'(x)_1 =$

0	0	0	1	1
0	0	0	1	1
0	0	0	1	1
0	0	0	1	1
0	0	0	1	1

$\delta_{conv1} =$



-0.275	0	0.0463	0.3335	0.2325
-0.275	0	0.0463	0.3335	0.2325
-0.275	0	0.0463	0.3335	0.2325
-0.275	0	0.0463	0.3335	0.2325
-0.275	0	0.0463	0.3335	0.2325

$$f'(x)_2 =$$

1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

$$\delta_{conv2} =$$

0.1022	0.1679	0.1025	0.4419	0
0.1022	0.1679	0.1025	0.4419	0
0.1022	0.1679	0.1025	0.4419	0
0.1022	0.1679	0.1025	0.4419	0
0.1022	0.1679	0.1025	0.4419	0

$$f'(x)_3 =$$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	1	1

$$\delta_{conv3} =$$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0.1711	-0.148	-0.029	0.5537	0

$$f'(x)_4 =$$

1	1	1	1	1
1	0	1	0	1
0	1	1	1	0

0	0	0	0	0
0	0	0	0	0

$\delta_{conv4} =$

-0.591	0.3029	0.2494	0.5847	0.2748
0	0.2566	-0.021	0.3522	0
-0.591	0.2007	0.0508	0.4176	0.2748
0	0	0	0	0
0	0	0	0	0

Langkah 13

$$\Delta w_{Njk} = w - \alpha(\delta_k * I_j)$$

	H1	H2	H3	H4
1	0.983 -0.0112 -0.99315	-1.014 -0.0209 0.989664	0.9947566 0.9962 1.000067	-1.009 -1.0062 -0.99817
1	0.981 -0.0145 -0.99087	-1.015 -0.027 0.987617	0.0002935 -0.0041 -0.00142	-0.011 -0.0147 0.001298
	0.984 -0.014 -0.99087	-1.011 -0.0259 0.989296	-1 -1 -1	0.985 0.9771 1.005507

4.3 Skenario Pengujian

Skenario pengujian membahas langkah dan alur pengujian yang akan digunakan. Pengujian ditujukan untuk mengevaluasi sistem untuk menemukan model terbaik. Dalam penelitian ini akan ada dua bentuk pengujian. Pengujian pertama adalah pengujian *loss value* sistem dan akurasi sistem, pengujian kedua yaitu dengan pengujian *confusion matrix*. Berikut pengujian yang akan dilakukan pada penelitian ini.

4.3.1 Skenario pengujian Loss dan Akurasi CNN

Dalam pengujian *loss* dan akurasi akan membandingkan beberapa model CNN. Dari beberapa model CNN akan diambil 3 sampai 5 terbaik dari hasil *loss* dan akurasi. *Loss* dan akurasi didapatkan dari proses pelatihan data latih setiap *epoch* nya. *Loss* adalah nilai error dalam pelatihan data latih, sedangkan akurasi adalah nilai validasi setiap *epoch* yang dilakukan sistem saat pelatihan data latih. Tabel 4.4 merupakan tabel skenario pengujian *loss* dan akurasi yang nantinya juga akan menampilkan grafik pergerakan *loss* dan akurasi.

Tabel 4.4 Tabel Skenario Pengujian Loss dan Akurasi

<i>epoch</i>	model 1	model 2	model 3	model 4	model 5	model m
1							
2							
3							
4							
.....							
n							

4.3.2 Skenario Pengujian *Confusion Matrix*

Pengujian ini untuk memastikan model yang telah berhasil dalam pengujian *loss* dan akurasi akan dianalisis lebih dalam untuk menentukan dan mengevaluasi model terbaik. Dalam pengujian ini akan membutuhkan hasil prediksi data uji kesetiap model yang tersisa. Kemudian hasil prediksi akan dilakukan perhitungan akurasi, presisi, *recall*, dan *F-score*. Kemudian mengevaluasi model terpilih hasil pengujian. Tabel 4.5 adalah tabel skenario pengujian untuk melihat hasil keluaran data uji.

Tabel 4.5 Skenario Pengujian *Confusion Matrix*

		Kelas Sebenarnya			
		Kelas 1	Kelas 2	...	Kelas n
Kelas Prediksi	Kelas 1	x_{11}	x_{12}	...	x_{1n}
	Kelas 2	x_{21}	x_{22}	...	x_{2n}

	Kelas n	x_{n1}	x_{n2}	...	x_{nn}



BAB 5 IMPLEMENTASI

Pada bab ini berisi hasil implementasi berdasarkan perancangan yang sudah dibuat pada bab sebelumnya. Bab ini terdiri dari penjelasan mengenai lingkungan implementasi dan hasil implementasi kode sumber beserta penjelasannya.

5.1 Lingkungan Implementasi

Pada subbab ini akan dijelaskan lingkungan implementasi yang terdiri dari lingkungan implementasi perangkat keras dan perangkat lunak yang digunakan untuk mengimplementasikan sistem.

5.1.1 Lingkungan Perangkat Keras

Pada Tabel 5.1 ditunjukkan spesifikasi pada lingkungan perangkat keras yang digunakan untuk mengimplementasikan sistem.

Tabel 5.1 Lingkungan Perangkat Keras

Nama Komponen	Keterangan
Processor	Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz
Memory	10.00 GB RAM DDR3L
Storage	500GB HDD
VGA	nVidia GeForce
Smartphone	Meizu M4 (Kamera 13MP) Perangkat ini digunakan untuk pengambilan citra makanan.

5.1.2 Lingkungan Perangkat Lunak

Pada Tabel 5.2 ditunjukkan spesifikasi pada lingkungan perangkat lunak yang digunakan dalam pengimplementasian sistem.

Tabel 5.2 Lingkungan Perangkat Lunak

Nama Perangkat Lunak	Keterangan
Windows 10 Pro 64-bit	Merupakan sistem operasi yang digunakan pada perangkat keras laptop
Python v3.7.1	Bahasa pemrograman yang digunakan untuk mengimplementasikan sistem
Google Colab	Digunakan untuk menjalankan kode program
Microsoft Word 2016	Pengolah dokumen yang digunakan untuk membuat laporan penelitian ini
Microsoft Excel 2016	Pengolah angka yang digunakan untuk perhitungan manual
draw.io	Digunakan untuk membuat perancangan sistem berupa diagram

5.2 Implementasi Algoritme *library*

Pada subbab ini akan ditunjukkan hasil implementasi algoritme berupa kode sumber dan pembahasannya. Pada subbab ini menggunakan kode menggunakan modul *Tensorflow*. *Tensorflow* merupakan *library* untuk *machine learning* khususnya pada *deep learning* yang merupakan produk versi 2 dari Google Brain. Pada penjelasan ini akan terbagi menjadi 2 yaitu proses augmentasi data, penentuan arsitektur CNN dan proses pembelajaran.

5.2.1 Implementasi Augmentasi Data

Augmentasi data adalah proses mengolah data sebelum dilakukannya proses pembelajaran. Diataranya yaitu proses untuk memperbanyak data dengan merotasi data masukan, menormalisasikan data, menentukan *batch size* yaitu menjalankan citra sekaligus sebanyak *batch size*.

Tabel 5.3 Implementasi Augmentasi Data

no	Source code
1.	<code>img_height = 1024</code>
2.	<code>img_width = 614</code>
3.	<code>n_output = 20</code>
4.	<code>batch_size = 10</code>
5.	<code>dir_dataset = 'data_latih_resize'</code>
	<code>#data scaling to 0 -1</code>
6.	<code>train_datagen = ImageDataGenerator(</code>
7.	<code> rescale=1./255,</code>
8.	<code> validation_split = 0.2</code>
	<code>)</code>
9.	<code>test_datagen = ImageDataGenerator(rescale=1./255)</code>
10.	<code>train_generator = train_datagen.flow_from_directory(</code>
11.	<code> dir_dataset,</code>
12.	<code> target_size=(img_height,img_width),</code>
13.	<code> batch_size=batch_size,</code>
14.	<code> class_mode='sparse',</code>
17.	<code> subset = 'training'</code>
	<code>)</code>
18.	<code>validation_generator = train_datagen.flow_from_directory(</code>
19.	<code> dir_dataset,</code>
20.	<code> target_size=(img_height,img_width),</code>
21.	<code> batch_size=batch_size,</code>
22.	<code> class_mode='sparse',</code>
23.	<code> subset = 'validation'</code>
24.	<code>)</code>

Berdasarkan Tabel 5.11 No. 1-6 Inisialisasi parameter, *img_width* dan *img_height* merupakan dimensi citra yang akan digunakan, *batch size* *batch* yang digunakan untuk training, *dir_dataset* merupakan alamat direktori data latih. No. 7-11 merupakan proses normalisasi data berdasarkan hasil perancangan. *ImageGenerator* adalah sebuah kelas yang berfokus mengolah citra sebelum dilakukannya proses *deep learning*, seperti rotasi citra, membalik citra, dll. Pada

no 7-11 penggunaan *ImageGenerator* digunakan hanya untuk menormalisasi data, karena berdasarkan perancangan. No. 12-24 merupakan proses mengeksekusi citra menjadi matriks baru sesuai inisialisasi *imageGenerator* pada no 7-11.

5.2.2 Implementasi Penentuan Arsitektur CNN

Pada subbab ini akan menunjukkan kode program mengenai inisialisasi arsitektur dari CNN. Arsitektur CNN akan dibentuk dari *layer* konvolusi, *layer pooling*, dan *Fully Connected*.

Tabel 5.4 Kode Program Arsitektur

no	Source code
1.	<code>classifier = Sequential() #initialize the CNN</code>
2.	<code>#1. Convolution</code>
3.	<code>classifier.add(Convolution2D(32,3,3, input_shape=(img_height,img_width,3), activation='relu'))</code>
4.	<code>#2. Pooling</code>
5.	<code>#classifier.add(MaxPooling2D(pool_size=(2,2)))</code>
6.	<code>classifier.add(Convolution2D(64, kernel_size=(3,3),</code>
7.	<code>activation='relu'))</code>
8.	<code>classifier.add(MaxPooling2D(pool_size=(2,2)))</code>
9.	<code>#3. Flattening</code>
10.	<code>classifier.add(Flatten())</code>
11.	<code>#4. Full connection</code>
12.	<code>classifier.add(Dense(output_dim = 32, activation='relu'))</code>
13.	<code>classifier.add(Dense(output_dim =34, activation='softmax'))</code>
14.	<code>#compiling CNN</code>
15.	<code>classifier.compile(optimizer='adam', loss</code>
16.	<code>='sparse_categorical_crossentropy', metrics=['accuracy'])</code>
17.	<code>classifier.summary()</code>

Berdasarkan Tabel 5.4 No. 1 instansiasi kelas *neural network*. No. 2-8 menambahkan prosesn konvolusi dengan aktivasi relu dan diakhiri proses *pooling*. No 2-8 menentukan arsitektur pembelajaran fitur dengan kedelaman maksimal tujuh yang telah dijelaskan dalam perancangan. No. 9-10 proses *flattening* atau pemecahan matriks menjadi 1 dimensi untuk dilakukannya proses *fully connected*. No. 11-13 menambahkan proses *fully connected* dengan kedalaman 2. No. 14-16 proses mengompile dari hasil penambahan layer yang digunakan dengan *optimizer adam* dan *loss cross etropy*. No. 17 perintah untuk menampilkan arsitektur yang digunakan.

5.2.3 Implementasi proses pembelajaran CNN

Pada subbab ini akan menunjukkan kode program mengenai proses atau variabel yang digunakan untuk pelatihan.

Tabel 5.5 Kode Program Pelatihan

no	Source code
1.	<code>classifier.fit_generator(</code>
2.	<code> train_generator,</code>
3.	<code> steps_per_epoch =100,</code>
4.	<code> epochs=15,</code>
5.	<code> validation_data =validation_generator,</code>
6.	<code> validation_steps=10)</code>
7.	<code>classifier.save('model.h5')</code>

Berdasarkan Tabel 5.5 No. 1 menjalankan fungsi `fit_generator` yaitu proses pelatihan. No 2 Variabel `train_generator` adalah variabel yang berisikan citra yang sudah dinormalisasi. No 3 `step_per_epoch` berfungsi menghitung nilai *loss* dan akurasi setiap 100 langkah dalam *epoch*nya. 1 langkah mengartikan satu citra diproses satu arsitektur yang sudah di inialisasikan. No 4 *epoch* merupakan banyaknya zaman yang digunakan dalam pelatihan. Satu zaman mengartikan telah menjalankan seluruh langkah citra yang ada. No 5-6 proses memvalidasi hasil pelatihan. No 7 menyimpan model *sequential* dengan ekstensi `.h5` atau biasa disebut dengan HDF5 merupakan model penyimpanan yang disarankan keras karena dapat dengan mudah menyimpan bobot dan konfigurasi model dalam satu file.

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian sistem akan dilakukan berdasarkan skenario pengujian. Dalam skenario pengujian akan dilakukan dua bentuk pengujian yaitu membandingkan *loss* dan akurasi antar model, dan pengujian *confusion matrix*. Setiap hasil pengujian disertakan analisis berdasarkan hasil pengujian untuk menemukan kesimpulan.

6.1 Pengujian *Loss* dan Akurasi pada Data Uji

Pengujian *loss* yaitu membandingkan *loss* dan akurasi antar model CNN yang dibuat. Pengujian *loss* dan akurasi bertujuan untuk mengetahui seberapa cepat sistem mendapatkan bobot terbaik. Selain itu, pengujian dilakukan untuk mengetahui pengaruh *layer* dalam mencari bobot terbaik. Dalam pengujian ini menggunakan 280 citra yang terdiri dari 20 kelas lebih jelasnya pada subbab 3.2. Model yang digunakan dalam pengujian ini ditunjukkan Tabel 6.1.

Tabel 6.1 Model CNN

Nama Model	<i>Convolution Layer</i>	<i>Maxpooling Layer</i>	<i>Fully connected Layer</i>
Model 1	1	1	2
Model 2	2	2	2
Model 3	3	3	2
Model 4	4	4	2
Model 5	5	5	2
Model 6	6	6	2
Model 7	7	7	2

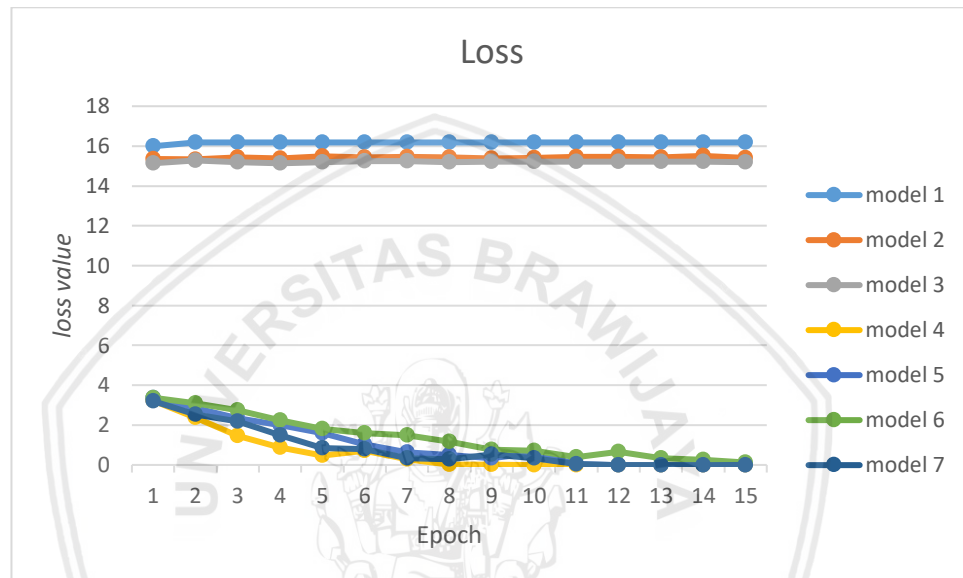
Berdasarkan Tabel 6.1 *Convolution Layer* dan *Maxpooling layer* jumlahnya sama agar mudah dipelajari polanya. Dalam prosesnya *maxpooling* selalu setelah *convolution layer*. Sedangkan *epoch* yang digunakan untuk melakukan pembelajaran sebanyak 15 kali. Tabel 6.2 menunjukkan hasil perkaman *loss value* dari proses pembelajaran sistem.

Tabel 6.2 Hasil Perekaman Nilai *Loss*

<i>Epoch</i>	Model1	Model2	Model3	Model4	Model5	Model6	Model7
1	15,99	15,351	15,137	3,2262	3,3572	3,3738	3,2043
2	16,18	15,334	15,279	2,3658	2,8615	3,0956	2,5225
3	16,18	15,419	15,178	1,474	2,3434	2,745	2,1874
4	16,18	15,387	15,135	0,8782	1,9841	2,2252	1,4925
5	16,18	15,489	15,178	0,4785	1,591	1,8083	0,8352
6	16,18	15,424	15,247	0,7213	1,0418	1,6062	0,7869
7	16,18	15,457	15,247	0,2862	0,6437	1,4824	0,3372
8	16,18	15,419	15,178	0,027	0,5104	1,1542	0,2725
9	16,18	15,382	15,205	0,0034	0,3351	0,7645	0,5308

Tabel 6.3 Hasil Perrekaman Nilai Loss (Lanjutan)

Epoch	Model1	Model2	Model3	Model4	Model5	Model6	Model7
10	16,18	15,392	15,226	0,00084	0,5923	0,7011	0,3365
11	16,18	15,457	15,231	0,00043	0,0419	0,3977	0,0348
12	16,18	15,457	15,231	0,00023	0,0022	0,6686	0,0022
13	16,18	15,424	15,226	0,00012	0,0012	0,3251	0,0011
14	16,18	15,505	15,205	0,00006	0,0005	0,2591	0,000744
15	16,18	15,408	15,178	0,00004	0,0003	0,1214	0,000481



Gambar 6.1 Grafik perekaman loss

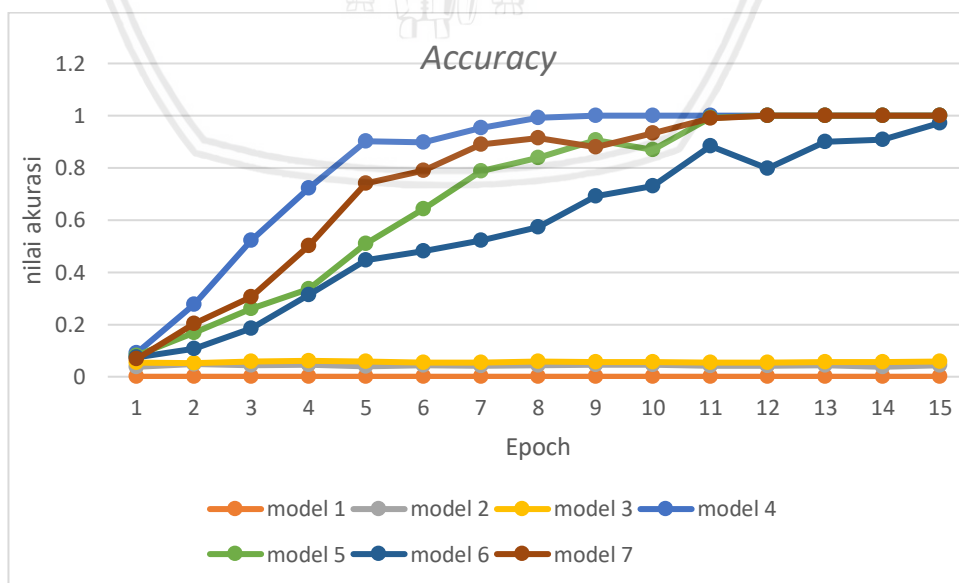
Berdasarkan Tabel 6.2, Tabel 6.3 dan Gambar 6.1 dapat dilihat bahwa penggunaan *layer* konvolusi kurang dari 4 *layer* akan mengalami *loss* yang besar dan konvergen setiap *epoch*nya yaitu sebesar 15,00 sampai 16,50. Model 1 memiliki *loss* selalu besar karena hanya terdapat 1 *layer* konvolusi didalam arsitekturnya, memiliki *layer* yang terlalu sedikit membuat sistem akan sulit untuk menemukan fitur khas pada setiap citra. Selain memiliki *layer* konvolusi yang sedikit, Model 1 juga memiliki *layer pooling* hanya 1, yang mengakibatkan data terlalu kompleks sehingga akan terjadinya *underfitting*. Maka penggunaan CNN dengan Model 1, Model 2, dan Model 3 tidak dapat mempelajari gambar apapun. Sedangkan model yang memiliki arsitektur dengan lebih dari 3 *layer* konvolusi dan 3 *layer pooling* mengalami penurunan *loss* setiap *epoch* sampai mendekati nilai nol pada akhirnya. Jadi Model 4, Model 5, Model 6, dan Model 7 dapat memahami data latih dengan baik dibuktikan dengan nilai akurasi pada gambar 6.2.

Gambar 6.2 dan Tabel 6.4 menyatakan bahwa hampir semua model yang memiliki *layer* konvolusi dan *pooling* lebih dari 3 memiliki akurasi sebesar 1. Memiliki akurasi sebesar 1 dapat dikatakan bahwa sistem mampu memisahkan seluruh data latih berdasarkan kelasnya masing-masing. Karena fungsi dari akurasi

adalah validasi proses data latih. *Layer* konvolusi dan *pooling* yang dibutuhkan yaitu minimal 4 agar sistem dapat mempelajari citra. Sedangkan semakin besar *epoch* maka nilai *loss* dan akurasi akan semakin baik akan tetapi membuat sistem berproses semakin lama. Model 4 lebih cepat dalam memahami fitur citra dibuktikan dengan nilai akurasi yang paling besar setiap *epoch*. Maka berdasarkan evaluasi nilai *loss* dan akurasi dengan 280 data latih yang terdiri dari 20 kelas, arsitektur terbaik menggunakan 4 *layer* konvolusi, 4 *layer pooling* dan 2 *layer fully connected*.

Tabel 6.4 Hasil Perekaman Akurasi Data Latih

Epoch	Model1	Model2	Model3	Model4	Model5	Model6	Model7
1	0	0,038	0,053	0,0903	0,081	0,0739	0,0673
2	0	0,0486	0,052	0,276	0,1684	0,1073	0,204
3	0	0,0433	0,0583	0,5217	0,2614	0,1857	0,306
4	0	0,0453	0,0609	0,7213	0,3366	0,3139	0,5007
5	0	0,039	0,0583	0,902	0,5097	0,4464	0,7407
6	0	0,043	0,054	0,897	0,6424	0,481	0,7896
7	0	0,041	0,054	0,953	0,7883	0,521	0,889
8	0	0,0433	0,0583	0,9927	0,8376	0,5731	0,915
9	0	0,0456	0,0566	1	0,9064	0,6917	0,8793
10	0	0,045	0,0553	1	0,8683	0,7313	0,9327
11	0	0,041	0,055	1	0,992	0,8843	0,989
12	0	0,041	0,055	1	1	0,7974	1
13	0	0,043	0,0553	1	1	0,8994	1
14	0	0,038	0,0566	1	1	0,9087	1
15	0	0,044	0,0583	1	1	0,972	1



Gambar 6.2 Grafik Perekaman Akurasi

6.2 Pengujian *Confusion Matrix*

Pengujian ini memiliki tujuan untuk mengetahui kemampuan sistem dalam memprediksi. Dalam pengujian *confusion matrix* ini menggunakan 100 data uji dari 20 kelas. Pengujian akan dilakukan sebanyak model yang lolos dalam pengujian *loss* dan akurasi. Model yang layak berdasarkan pengujian *loss* dan akurasi yaitu Model 4, Model 5, Model 6 dan Model 7.

Model 4 memiliki 4 konvolusi *layer* dan 4 *maxpooling* dan 2 *layer Fully connected*. Tabel 6.5 merupakan hasil pengujian *confusion matrix*.











Tabel 6.5 Pengujian Model 4

Kode	FN	FP	TN	TP	Presisi	Recall	acc	F-score
0	1	2	93	4	0,666667	0,8	0,97	0,727273
1	1	1	94	4	0,8	0,8	0,98	0,8
2	1	0	95	4	1	0,8	0,99	0,888889
3	4	4	91	1	0,2	0,2	0,92	0,2
4	0	1	94	5	0,833333	1	0,99	0,909091
5	1	1	94	4	0,8	0,8	0,98	0,8
6	0	1	94	5	0,833333	1	0,99	0,909091
7	1	5	90	4	0,444444	0,8	0,94	0,571429
8	3	0	95	2	1	0,4	0,97	0,571429
9	3	1	94	2	0,666667	0,4	0,96	0,5
10	3	0	95	2	1	0,4	0,97	0,571429
11	1	0	95	4	1	0,8	0,99	0,888889
12	1	1	94	4	0,8	0,8	0,98	0,8
13	2	4	91	3	0,428571	0,6	0,94	0,5
14	1	2	93	4	0,666667	0,8	0,97	0,727273
15	1	1	94	4	0,8	0,8	0,98	0,8
16	2	1	94	3	0,75	0,6	0,97	0,666667
17	2	3	92	3	0,5	0,6	0,95	0,545455
18	1	1	94	4	0,8	0,8	0,98	0,8
19	2	2	93	3	0,6	0,6	0,96	0,6
	31	31	1869	69	0,729484	0,69	0,969	0,688846

Berdasarkan Tabel 6.5 pengujian menampilkan nilai rata-rata presisi sebesar 73%, rata-rata *recall* sebesar 69%, nilai rata-rata *accuracy* sebesar 0.97% dan rata-rata *F-score* sebesar 69%. Dalam pengujiannya Model 4 memiliki nilai *F-*

score terendah sebesar 0.2 atau hanya 20% tepatnya pada kode makanan 3. Penilaian *F-score* didapat dari nilai presisi dan *recall*nya terbukti dengan nilai presisi sebesar 0,2 dan nilai *recall* sebesar 0,2. Nilai *recall* yang rendah itu dipengaruhi oleh FN yang besar, sedangkan FN didapat dengan kesalahan sistem dalam memprediksi makanan dengan no kode 3, seperti yang ditunjukkan Tabel 6.6. Berdasarkan tabel 6.6 keempat hasil prediksi yang salah, tidak ada yang memiliki kesamaan kelas. Maka bisa dikatakan sistem kurang mampu mempelajari ciri khas untuk makanan ke 3. Sedangkan Nilai presisi yang rendah itu dipengaruhi oleh FP yang besar, sedangkan FP didapat dengan kesalahan sistem dalam memprediksi makanan data uji menjadi makanan no kode 3, seperti yang ditunjukkan Tabel 6.7.











Tabel 6.6 Hasil Prediksi Makanan 3 pada Model 4

No	Nama Data Uji	Gambar Data Uji	Gambar Prediksi
1	P90124-105633		
2	P90124-104022		
3	P90124-105650		
4	P90124-104044		
5	P90124-114309		

Berdasarkan Tabel 6.6 sistem Model 4 belum bisa mempelajari ciri-ciri dari makanan kode 3. Terbukti pada data uji no 1 makanan kode 3 memiliki tekstur gradien perbedaan warna yang serupa yaitu warna hijau dengan warna coklat. Warna hijau pada daun dan warna coklat pada makanan, sistem menganggap

warna hijau untuk makanan kode 23 yang berwarna hijau dan warna coklat untuk makanan kode 23 yang berwarna coklat. Sedangkan pada data uji 2 memiliki bentuk yang hampir sama, warna sama dan tekstur yang mirip. Begitu juga pada data uji no 3 dan 4 yang memiliki masalah seperti data no 2 dengan kelas yang diprediksi. Kecuali untuk data no 5 dapat di klasifikasi terhadap kelasnya karena data uji no 5 hampir tidak ada citra dengan bentuk penataan yang mirip pada kelas lain. Maka pada kelas makanan kode 3 memiliki data latih citra yang sedikit dari segi sudut pandang yang membuat sistem *underfitting* atau kurang memahami ciri-ciri dari kelas makanan kode 3. Diharapkan akan ditambahkan jumlah citra data latih yang memiliki sudut pandang berbeda.

Tabel 6.7 Hasil Prediksi Model 4 Terhadap Makanan 3

No	Nama Data Uji	Gambar Data Uji	Gambar Prediksi
1	P90124-105304		
2	P90124-105311		
3	P90124-103511		
4	P90124-104413		
5	P90124-114309		

Berdasarkan Tabel 6.7, 4 citra data uji yang terprediksi menjadi Makanan kode 3 dan hasil prediksi itu salah, maka membuktikan bahwa data latih kode 3 mengalami *overfitting* dini atau terjadinya data redundansi. Redundansi disini bukan karena data terduplikat, akan tetapi pengambilan citra dari sudut pandang yang sama dari beberapa citra data latih untuk kode makanan 3.

Berdasarkan hasil analisis Model 4, untuk data makanan kode 3 yang mendapatkan *F-score* rendah sebesar 0.2 dengan data latih sebanyak 16 citra. Disebabkan karena data latih yang digunakan tidak mewakili keseluruhan ciri dari makanan kode 3 atau kurangnya data latih dalam segi sudut pandang pengambilan citra, serta terdapat citra redundansi pada data latih atau pengambilan citra secara berulang dalam satu sudut pandang.

Model 5 memiliki 5 konvolusi *layer*, 5 *maxpooling* dan 2 *layer Fully connected*. Tabel 6.8 merupakan hasil pengujian *confusion matrix*.

Tabel 6.8 Hasil Pengujian Model 5

	FN	FP	TN	TP	P	R	acc	<i>F-score</i>
0	2	1	94	3	0,75	0,6	0,97	0,666667
1	1	1	94	4	0,8	0,8	0,98	0,8
2	0	4	91	5	0,555556	1	0,96	0,714286
3	2	5	90	3	0,375	0,6	0,93	0,461538
4	1	3	92	4	0,571429	0,8	0,96	0,666667
5	0	14	81	5	0,263158	1	0,86	0,416667
6	0	3	92	5	0,625	1	0,97	0,769231
7	4	1	94	1	0,5	0,2	0,95	0,285714
8	3	3	92	2	0,4	0,4	0,94	0,4
9	5	1	94	0	0	0	0,94	0
10	5	0	95	0	0	0	0,95	0
11	4	1	94	1	0,5	0,2	0,95	0,285714
12	2	3	92	3	0,5	0,6	0,95	0,545455
13	1	3	92	4	0,571429	0,8	0,96	0,666667
14	5	1	94	0	0	0	0,94	0
15	5	0	95	0	0	0	0,95	0
16	2	1	94	3	0,75	0,6	0,97	0,666667
17	3	1	94	2	0,666667	0,4	0,96	0,5
18	0	1	94	5	0,833333	1	0,99	0,909091
19	4	2	93	1	0,333333	0,2	0,94	0,25
	49	49	1851	51	0,449745	0,51	0,951	0,450218

Berdasarkan Tabel 6.8 pengujian menampilkan nilai rata-rata presisi sebesar 44%, rata-rata *recall* sebesar 51%, nilai rata-rata *accuracy* sebesar 0.95% dan rata-rata *F-score* sebesar 45%. Dalam pengujiannya Model 5 memiliki 3 nilai

yang perbedaannya terlalu signifikan antara *recall* dan presisi yaitu terdapat pada makanan yang berkode 2, 5 dan 6. Perbedaan yang signifikan itu lebih besar pada nilai *recall*, jadi sistem lebih mengutamakan hasil yang memiliki ciri khas paling dominan atau yang memiliki ciri khas yang paling banyak walaupun kelas lain memilikinya juga.

Model 6 memiliki 6 konvolusi *layer* dan 6 *maxpooling* dan 2 *layer Fully connected*. Tabel 6.9 merupakan hasil pengujian *confusion matrix*.

Tabel 6.9 Hasil Pengujian Model 6

	FN	FP	TN	TP	P	R	acc	F-score
0	2	2	93	3	0,6	0,6	0,96	0,6
1	3	1	94	2	0,666667	0,4	0,96	0,5
2	1	4	91	4	0,5	0,8	0,95	0,615385
3	4	5	90	1	0,166667	0,2	0,91	0,181818
4	3	0	95	2	1	0,4	0,97	0,571429
5	0	9	86	5	0,357143	1	0,91	0,526316
6	0	0	95	5	1	1	1	1
7	2	1	94	3	0,75	0,6	0,97	0,666667
8	3	0	95	2	1	0,4	0,97	0,571429
9	2	1	94	3	0,75	0,6	0,97	0,666667
10	3	2	93	2	0,5	0,4	0,95	0,444444
11	2	1	94	3	0,75	0,6	0,97	0,666667
12	0	0	95	5	1	1	1	1
13	1	4	91	4	0,5	0,8	0,95	0,615385
14	2	3	92	3	0,5	0,6	0,95	0,545455
15	1	0	95	4	1	0,8	0,99	0,888889
16	4	0	95	1	1	0,2	0,96	0,333333
17	3	2	93	2	0,5	0,4	0,95	0,444444
18	0	0	95	5	1	1	1	1
19	1	2	93	4	0,666667	0,8	0,97	0,727273
	37	37	1863	63	0,710357	0,63	0,963	0,62828

Berdasarkan Tabel 6.9 pengujian menampilkan nilai rata-rata presisi sebesar 71%, rata-rata *recall* sebesar 63%, nilai rata-rata *accuracy* sebesar 0.96% dan rata-rata *F-score* sebesar 63%. Dalam pengujiannya Model 6 memiliki 3 nilai yang perbedaannya terlalu signifikan antara *recall* dan presisi yaitu terdapat pada makanan yang berkode 4, 8 dan 16. Perbedaan yang signifikan itu lebih besar pada

nilai presisi, jadi sistem lebih mengutamakan hasil yang memiliki ciri khas yang paling berbeda dengan kelas yang lainnya atau ciri khas yang tidak dimiliki kelas lainnya.

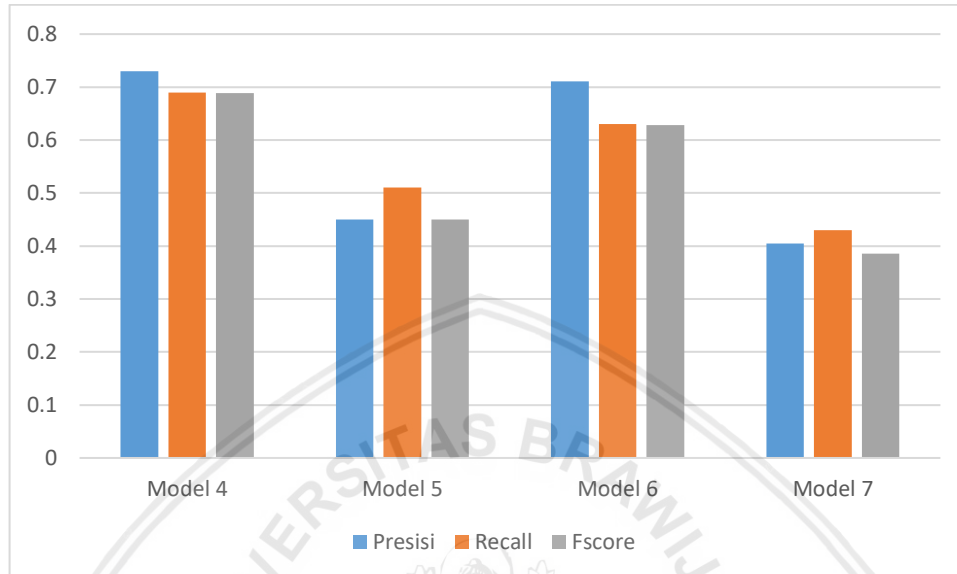
Model 7 memiliki 7 konvolusi *layer* dan 7 *maxpooling* dan 2 *layer Fully connected*. Tabel 6.10 merupakan hasil pengujian *confusion matrix*.

Tabel 6.10 Hasil Pengujian Model 7

	FN	FP	TN	TP	P	R	acc	F-score
0	5	0	95	0	0	0	0,95	0
1	4	2	93	1	0,333333	0,2	0,94	0,25
2	4	1	94	1	0,5	0,2	0,95	0,285714
3	4	2	93	1	0,333333	0,2	0,94	0,25
4	3	4	91	2	0,333333	0,4	0,93	0,363636
5	0	16	79	5	0,238095	1	0,84	0,384615
6	1	0	95	4	1	0,8	0,99	0,888889
7	2	10	85	3	0,230769	0,6	0,88	0,333333
8	5	2	93	0	0	0	0,93	0
9	0	6	89	5	0,454545	1	0,94	0,625
10	4	1	94	1	0,5	0,2	0,95	0,285714
11	5	2	93	0	0	0	0,93	0
12	2	1	94	3	0,75	0,6	0,97	0,666667
13	1	1	94	4	0,8	0,8	0,98	0,8
14	5	0	95	0	0	0	0,95	0
15	2	2	93	3	0,6	0,6	0,96	0,6
16	5	1	94	0	0	0	0,94	0
17	2	4	91	3	0,428571	0,6	0,94	0,5
18	1	0	95	4	1	0,8	0,99	0,888889
19	2	2	93	3	0,6	0,6	0,96	0,6
	57	57	1843	43	0,405099	0,43	0,943	0,386123

Berdasarkan Tabel 6.10 pengujian menampilkan nilai rata-rata presisi sebesar 40%, rata-rata *recall* sebesar 43%, nilai rata-rata *accuracy* sebesar 94% dan rata-rata *F-score* sebesar 39%. Berdasarkan hasil pengujiannya nilai presisi dan *recall* relatif rendah hampir setengahnya. Maka bisa disimpulkan bahwa sistem terlalu detail dalam memahami ciri khas pada citra atau biasa disebut dengan *overfitting*.

Berdasarkan hasil pengujian Tabel 6.5, Tabel 6.7, Tabel 6.8 dan Tabel 6.9 dapat di sederhanakan dengan gambar 6.3 dan menampilkan rata-rata nilai dari presisi, recall, dan *F-score*. Penggunaan *Accuracy* sebagai acuan evaluasi tidak dapat digunakan karena nilai TN tidak simetris dengan keluaran yang lain.



Gambar 6.3 Grafik Hasil Rata-rata Pengujian

Gambar 6.3 menerangkan bahwa model 4 merupakan model yang terbaik dari model yang lainnya dengan nilai *F-score* rata-ratanya 68.8%. Model 6 memiliki nilai yang mendekati Model 4 dengan nilai *F-score* rata-ratanya 62.8%. Permasalahan yang terjadi disemua model adalah dalam pengambilan data latih hanya terdiri dari beberapa sudut pandang yang dilakukan berulang dan tidak sama setiap kelasnya. Pengambilan dengan sedikit sudut pandang akan membuat sistem kurang bisa dalam mempelajari ciri-ciri dari makanan, dan pengambilan secara berulang dalam satu sudut pandang akan membuat *overfitting* dini. Selain pengambilan data, proses pelatihan terlalu lama akan meningkatkan nilai *error* saat pengujian, yang dapat menurunkan nilai *recall* dan presisi. Terbukti dari Model 6 yang memiliki *F-score* 100% sebanyak 3, tepatnya pada data uji kelas makanan kode 6, 12, dan 18.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang sudah dilakukan, didapatkan kesimpulan sebagai berikut.

1. Dalam membangun asitektur model *Convolutional Neural Network* untuk pengklasifikasian citra makanan tradisional membutuhkan 4 *layer* konvolutional, 4 *layer maxpooling* dan 2 *layer Fully connected*. Arsitektur tersebut didapatkan karena mendapatkan nilai *loss value* terkecil dengan nilai 0.000044 pada *epoch* ke 15 saat proses pembelajaran dan mendapatkan nilai 73% presisi, 69% recall dan 69% *F-score*.

7.2 Saran

Penelitian ini masih jauh dari kata sempurna dan masih banyak yang perlu ditingkatkan. Diharapkan agar penelitian yang akan mendatang mampu melakukan perbaikan ataupun pengembangan lebih lanjut agar penelitian ini bisa menjadi lebih baik. Adapun beberapa saran yang dapat dilakukan untuk penelitian selanjutnya:

1. Dalam penelitian ini, citra makanan yang memiliki objek berbeda sudut pandang akan memiliki hasil yang berbeda pula. Diharapkan untuk penelitian selanjutnya data set diperbanyak dengan cara citra makanan di foto dari berbagai sudut. Banyaknya sudut pandang dimungkinkan untuk meningkatkan nilai presisi dan *recall*.

DAFTAR PUSTAKA

- Abdul Kadir, A. S. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.
- Aggarwal, V., & Gagandeep. (2018). A review. *deep learning technique for image classification*, 21-25.
- Chamidah, N., Wiharto, & Salamah, U. (2012). Pengaruh Normalisasi Data pada Jaringan Syaraf Tiruan Backpropagasi Gradient Descent Adaptive Gain untuk Klasifikasi. *ITSMART*, 2301–7201.
- Feist, J., & Handriatno. (2010). *teori kepribadian*. Jakarta: Salemba Humanika.
- Ferguson, P. P. (2010). Culinary Nationalism. *Gastronomica the Journal of Food and Culture*, 102-109.
- Ghoneim, S. (2019, April 2). *Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?* Retrieved from Towards Data Science.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge London: The MIT Press.
- He, Y., Xu, C., Khanna, N., Boushey, C. J., & Delp, E. J. (2013). *FOOD IMAGE ANALYSIS. SEGMENTATION, IDENTIFICATION AND WEIGHT ESTIMATION*.
- Kingma, D. P., & Ba, J. I. (2015). *Adam. A METHOD FOR STOCHASTIC OPTIMIZATION*.
- Kumari, N., & Saxena, S. (2018). *Review of Brain Tumor Segmentation and Classification*.
- Madenda, S. (2015). *Pengolahan Citra & Video Digital*. Jakarta: Erlangga.
- Maslow, A. (1994). *Motivasi dan Keribadian 1*. Diterjemahkan dari Bahasa Inggris oleh Nurul Imam. Jakarta: LPPM dan PT. Pustaka Binaman Pressindo.
- Santosa, B., & Umam, A. (2018). *Data Mining dan Big Data Analytics*. Yogyakarta: Penebar Media Pustaka.
- Sokolova, M., & Lapalme, G. (2009). Information Processing and Management. *A systematic analysis of performance measures for classification tasks*, 427-437.
- Widodo Budiharto, D. S. (2014). *Artificial Intelligence Konsep dan Penerapannya*. Yogyakarta: Andi.

LAMPIRAN A HASIL KELUARAN PREDIKSI MODEL 4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	sum	
0	4																	1			5	
1		4						1													5	
2			4											1							5	
3				1										1	1	1	1				5	
4					5																5	
5						4				1											5	
6							5														5	
7								4						1							5	
8	1					1			2						1						5	
9				3						2											5	
10		1									2			1					1		5	
11							1					4									5	
12													4							1	5	
13														3					1		1	5
14				1											4						5	
15								1								4					5	

16								2									3				5
17	1																	3		1	5
18													1						4		5
19					1			1												3	5
sum	6	5	4	5	6	5	6	9	2	3	2	4	5	7	6	5	4	6	5	5	100



LAMPIRAN B HASIL KELUARAN PREDIKSI MODEL 5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	sum	
0	3				2																	5
1		4		1																		5
2			5																			5
3			1	3			1															5
4	1				4																	5
5						5																5
6							5															5
7			1					1				1		1							1	5
8						3			2													5
9						5				0												5
10						3					0						1	1				5
11		1				1						1	1							1		5
12							1						3		1							5
13			1											4								5
14							1		2				2		0							5
15				3		1			1							0						5

16				1				1									3				5
17						1				1								2		1	5
18																			5		5
19			1		1									2						1	5
sum	4	5	9	8	7	19	8	2	5	1	0	2	6	7	1	0	4	3	6	3	100



LAMPIRAN C HASIL KELUARAN PREDIKSI MODEL 6

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	sum	
0	3		1	1																		5
1		2						1										2				5
2	1		4																			5
3		1		1								1		1							1	5
4			3		2																	5
5						5																5
6							5															5
7								3						2								5
8						3			2													5
9						2				3												5
10						3					2											5
11												3			2							5
12													5									5
13														4							1	5
14						1				1					3							5
15															1	4						5

16				3										1			1				5
17				1						2							2				5
18																			5		5
19	1																			4	5
sum	5	3	8	6	2	14	5	4	2	4	4	4	5	8	6	4	1	4	5	6	100



LAMPIRAN D HASIL KELUARAN PREDIKSI MODEL 7

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	sum	
0	0	1			2															2	5	
1		1				2							1	1								5
2		1	1			2					1											5
3				1		2				1						1						5
4					2	1		1								1						5
5						5																5
6							4					1										5
7						1		3											1			5
8						1			0	2		1							1			5
9										5												5
10						3					1									1		5
11			1		1	1		1				0					1					5
12				1					1					3								5
13														4					1			5
14				1				4							0							5
15								1		1						3						5

16						3		2									0				5
17								1		1								3			5
18									1										4		5
19					1					1										3	5
sum	0	3	2	3	6	21	4	13	2	11	2	2	4	5	0	5	1	7	4	5	100

