MODIFIKASI ARSITEKTUR LENET-5 UNTUK MENDETEKSI TUMOR OTAK BERBASIS CITRA MRI

SKRIPSI

Oleh:
MUHAMMAD YUSRIL FAKKARUDDIN
NIM. 17650026

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

MODIFIKASI ARSITEKTUR LENET-5 UNTUK MENDETEKSI TUMOR OTAK BERBASIS CITRA MRI

SKRIPSI

Oleh:
MUHAMMAD YUSRIL FAKKARUDDIN
NIM. 17650026

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S. Kom)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

LEMBAR PERSETUJUAN

MODIFIKASI ARSITEKTUR LENET-5 UNTUK MENDETEKSI TUMOR OTAK BERBASIS CITRA MRI

SKRIPSI

Oleh:
MUHAMMAD YUSRIL FAKKARUDDIN
NIM. 17650026

Telah Diperiksa dan Disetujui Untuk Diuji
Tanggal: 12 Oktober 2022

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Irwan Budi Santoso, M.Kom	Okta Qomaruddin Aziz, M.Kom
NIP. 19770103 201101 1 004	NIP. 19911019 201903 1 013

Mengetahui,
Ketua Prodi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan ST., M.MT ., IPM
NIP. 19771020 200912 1 001

LEMBAR PENGESAHAN

MODIFIKASI ARSITEKTUR LENET-5 UNTUK MENDETEKSI TUMOR OTAK BERBASIS CITRA MRI

SKRIPSI

Oleh:
MUHAMMAD YUSRIL FAKKARUDDIN
NIM. 17650026

Telah Dipertahankan di Depan Dewan Penguji dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal: 12 Oktober 2022

Susunan Dewan Penguji:

Ketua Penguji	:	Dr. M. Amin Hariyadi, M.T NIP. 19670018 200501 1 001	( )
Anggota Penguji I	:	Dr. Cahyo Crysdian NIP. 19740424 200901 1 008	( )
Anggota Penguji II	:	Dr. Irwan Budi Santoso, M.Kom NIP. 19770103 201101 1 004	( )
Anggota Penguji III	:	Okta Qomaruddin Aziz, M.Kom NIP. 19911019 201903 1 013	( )

Mengetahui,
Ketua Prodi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Fachrul Kurniawan ST., M.MT ., IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Yusril Fakkaruddin
NIM : 17650026
Prodi : Teknik Informatika
Fakultas : Sains dan Teknologi
Judul Skripsi : Modifikasi Arsitektur LENET-5 Untuk Mendeteksi Tumor Otak Berbasis Citra MRI

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 12 Oktober 2022

Yang membuat pernyataan,

Muhammad Yusril Fakkaruddin
NIM. 17650026

HALAMAN MOTTO

"*Dan jangan kamu berputus asa dari rahmat Allah. Sesungguhnya tiada berputus asa dari rahmat Allah, melainkan kaum yang kafir."*
Q.S. Yusuf (87)

HALAMAN PERSEMBAHAN

Bismillahirrohmanirrohim, puji syukur kehadirat Allah SWT atas limpahan karunia-Nya yang diberikan sehingga penulis dapat menuntaskan skripsi ini untuk menyelesaikan program S1. Tak lupa pula shalawat serta salam dijunjungkan kepada baginda rasulullah SAW. Karya ini saya persembahkan kepada:

Kedua orang tua saya Bapak Muhammad Zainul Muhsinin, S.Pd. dan Ibu Didiet Nugrahayu yang saya sangat sayangi dan senantiasa selalu mendukung saya dengan do'a dan motivasi yang tak terhingga. Dan kepada adik saya Muhammad Faizul Fikri semoga apa yang dicita-citakan dapat terwujud dan menjadi kebanggaan keluarga, bangsa, dan negara.

Dosen pembimbing penulis, Bapak Dr. Irwan Budi Santoso, M.Kom. dan Bapak Okta Qomaruddin Aziz, M.Kom. yang telah dengan sabar, tulus ,dan ikhlas dalam membimbing penelitian skripsi ini dan selalu memberikan masukan-masukan untuk menjalani setiap tahapan skripsi.

Seluruh dosen dan staff Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang dan guru-guru penulis yang ikut andil dalam penyelesaian skripsi dan memberikan ilmu yang bermanfaat.

Tidak lupa seluruh Keluarga Teknik Informatika, terutama teman-teman Unocore (Teknik Informatika angkatan 2017) yang telah memberikan semangat dan motivasi.

Serta teman-teman, keluarga, dan orang-orang yang hadir dalam kehidupan penulis yang tidak bisa disebutkan satu persatu yang terus mendorong dalam menyelesaikan skripsi ini.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT tuhan yang maha esa, yang maha agung dari segala yang ada di dunia dan akhirat ini, yang mana telah tercurahkan rahmat serta hidayahnya. Tak lupa sertakan shalawat dan salam kepada nabi besar junjungan umat islam yaitu Muhammad SAW, nabi yang membawa lentera di dalam kegelapan dan kebodohan. Sehingga penulis dapat menyelesaikan skripsi yang mengangkat judul "Modifikasi Arsitektur Lenet-5 Untuk Mendeteksi Tumor Otak Berbasis Citra MRI" Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada :

1. Prof. Dr. H. M. Zainuddin, M.A, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim.
2. Dr. Sri Hariani, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim.
3. Dr. Fachrul Kurniawan ST., M.MT ., IPM selaku Ketua Prodi Teknik Informatika
4. Dr. Irwan Budi Santoso, M.Kom selaku Dosen Pembimbing I yang telah membimbing dengan sabar dan ikhlas dalam penyusunan skripsi ini hingga selesai.
5. Okta Qomaruddin Aziz, M.Kom selaku Dosen Pembimbing II yang juga telah membimbing dan memberi nasihat dalam penyusunan skripsi ini hingga selesai.

6. Dr. M. Amin Hariyadi, M.T dan Dr. Cahyo Crysdian selaku Dosen Penguji yang telah memberikan banyak saran untuk kebaikan penulis.
7. Syahiduz Zaman, M.Kom selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.
8. Seluruh Dosen dan Staff Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang yang telah memberikan ilmunya baik secara langsung maupun tidak langsung dalam menyelesaikan skripsi ini.
9. Ayahanda, ibunda, dan adik tercinta yang senantiasa selalu memberikan dukungan dan do'a sehingga dapat menyelesaikan skripsi ini.
10. Rekan-rekan Unocore (Teknik Informatika angkatan 2017) dan teman-teman yang tidak bisa saya sebutkan satu persatu yang telah memberikan dukungan kepada penulis.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Wassalamu'alaikum Wr. Wb.

DAFTAR ISI

HALAMAN JUDUL ..... **ii**
LEMBAR PERSETUJUAN ..... **iii**
LEMBAR PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN TULISAN ..... **v**
HALAMAN MOTTO ..... **vi**
HALAMAN PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **ix**
DAFTAR ISI ..... **xi**
DAFTAR GAMBAR ..... **xiii**
DAFTAR TABEL ..... **xiv**
ABSTRAK ..... **xv**
ABSTRACT ..... **xvi**
الملخص ..... **xvii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Pernyataan Masalah 5
1.3 Tujuan Penelitian 5
1.4 Batasan Penelitian 5
1.5 Manfaat Penelitian 6
BAB II STUDI LITERATUR **7**
2.1 Convolutional Neural Network 7
2.2 Identifikasi Tumor Otak 8
BAB III METODE PENELITIAN **12**
3.1 Pengumpulan Data 12
3.2 Desain Sistem 12
3.3 *Image Preprocessing* 13
3.4 Implementasi *Convolutional Neural Network* 17
3.4.1 Layer Konvolusi 18
3.4.2 *Pooling Layer* 19
3.4.3 *Fully-Connected Layer* 20
3.4.4 Fungsi Aktivasi 20
3.5 Modifikasi Model Arsitektur LeNet-5 Usulan 21
3.6 Rancangan Sistem Aplikasi 32
3.7 Proses Training 33
BAB IV UJI COBA DAN PEMBAHASAN **36**
4.1 Skenario Uji Coba 36
4.2 Hasil Uji Coba 39
4.2.1 Hasil Pengujian Dengan *K-Fold Cross Validation* 40
4.3 Pembahasan 47
BAB V KESIMPULAN DAN SARAN **52**
5.1 Kesimpulan 52
5.2 Saran 52
DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 3.1 Desain Sistem 13
Gambar 3.2 *Dataset* Augmentasi Citra Tumor Otak 14
Gambar 3 .3 Hasil Augmentasi *Shear* 14
Gambar 3.4 Hasil Augmentasi *Zoom* 15
Gambar 3.5 Hasil Augmentasi *Horizontal* 15
Gambar 3.6 Hasil Augmentasi Teknik *Rotation* 16
Gambar 3.7 Hasil Augmentasi *Width Shift* 16
Gambar 3.8 Hasil Augmentasi *Height Shift* 17
Gambar 3.9 Ilustrasi Layer Konvolusi (sumber:Kadam, 2021) 18
Gambar 3.10 Ilustrasi *Pooling layer* (sumber: Santoso & Aryanto, 2018). . 19
Gambar 3.11 Ilustrasi *Fully-Connected Layer* (sumber: Pelletier, 2019) 20
Gambar 3.12 Fungsi *ReLu* (sumber: S. Sharma et,al., 2020). 21
Gambar 3.13 Model Arsitektur LeNet-5 (sumber: LeCun et al., 1998) 22
Gambar 3.14 *Flowchart* Arsitektur Original 24
Gambar 3.15 Kode Program Arsitektur Original 24
Gambar 3.16 Model Arsitektur Modifikasi 25
Gambar 3.17 *Flowchart* Arsitektur Modifikasi 27
Gambar 3.18 Kode Program Arsitektur Modifikasi 28
Gambar 3.19 *Flowchart* Rancangan Sistem Aplikasi 33
Gambar 3.20 Program Training 34
Gambar 3.21 Grafik Hasil *Training* Arsitektur Modifikasi 35
Gambar 4.1 *K-Fold Cross Validation* 37
Gambar 4.2 Tampilan Aplikasi 38
Gambar 4.3 Grafik Hasil *Training Validation Accuracy* Fold-1 40
Gambar 4.4 Grafik Hasil *Training Validation Accuracy* Fold-2 41
Gambar 4.5 Grafik Hasil *Training Validation Accuracy* Fold-3 43
Gambar 4.6 Grafik Hasil *Training Validation Accuracy* Fold-4 44
Gambar 4.7 Grafik Hasil *Training Validation Accuracy* Fold-5 45

DAFTAR TABEL

Tabel 2.1 Perbedaan Dengan Penelitian Sebelumnya 10
Tabel 3.1 Arsitektur CNN Original (Yann LeCun) 22
Tabel 3.2 Arsitektur CNN Modifikasi 25
Tabel 3.3 Perbandingan Arsitektur LeNet-5 Original dan Modifikasi 28
Tabel 4.1 *K-Fold Cross Validation* Model Arsitektur Modifikasi 45
Tabel 4.2 Rata-Rata Performa Sistem *K-Fold Cross Validation* Model Modifikasi 46

ABSTRAK

Fakkaruddin, Muhammad Yusril. 2022. **Modifikasi Arsitektur Lenet-5 Untuk Mendeteksi Tumor Otak Berbasis Citra MRI.** Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom., (II) Okta Qomaruddin Aziz, M.Kom.

Kata Kunci: *Tumor otak, Convolutional Neural Network, Lenet-5, Modifikasi arsitektur*

Tumor otak merupakan salah satu penyakit mematikan yang menyerang otak manusia dimana penyakit ini terjadi akibat adanya pertumbuhan sel tidak normal pada otak. Gejala awal ini timbul ketika manusia mengalami sakit kepala, kejang-kejang, mudah lelah, dan mual. Dengan mengamati gejala tersebut, penyakit dapat segera diidentifikasi dan dilakukan pencegahan awal dengan membangun sebuah sistem yang dapat mengidentifikasi suatu objek dengan mengimplementasi algoritma *Convolutional Neural Network* dengan menggunakan arsitektur *Lenet-5* oleh Yann Lecun. Dalam penelitian ini bertujuan untuk mengukur hasil dari akurasi, sensitivitas, dan spesifisitas arsitektur sebelum dan sesudah dimodifikasi. Dataset yang digunakan merupakan data sekunder yang berjumlah 500 citra yang terbagi menjadi dua *class* yaitu otak normal dan otak dengan tumor. Prepocessing data dilakukan dengan teknik augmentasi untuk memperbanyak hasil dari citra asli. Pengujian arsitektur modifikasi dilakukan dengan hasil akurasi sebesar 88,00%, sensitivitas sebesar 90,00%, spesifitas sebesar 86,00%. Kemudian dilakukan pengujian menggunakan *K-Fold Cross Validation* diperoleh hasil untuk model arsitektur modifikasi diperoleh hasil akurasi rata-rata 94,40%, sensitivitas rata-rata 96,27%, dan spesifitas rata-rata 97,06%.

ABSTRACT

Fakkaruddin, Muhammad Yusril. 2022. **Modification of Lenet-5 Architecture for Detecting Brain Tumors Based on MRI Image**. Undergraduate Thesis. Informatics Engineering Department, Faculty of Science and Technology. Islamic State University Maulana Malik Ibrahim Malang. Supervisor: (I) Dr. Irwan Budi Santoso, M.Kom., (II) Okta Qomaruddin Aziz, M.Kom.

Keywords: *Brain tumor, Convolutional Neural Network, Lenet-5, Architectural modification*

Brain tumor is one of the deadly diseases that attack the human brain where this disease occurs due to abnormal cell growth in the brain. These early symptoms arise when humans experience headaches, convulsions, fatigue, and nausea. By observing these symptoms, diseases can be immediately identified and early prevention carried out by building a system that can identify an object by implementing the Convolutional Neural Network algorithm using the Lenet-5 architecture by Yann Lecun. This study aims to measure the results of the accuracy, sensitivity, and specificity of the architecture before and after modification. The dataset used is secondary data consisting of 500 images which are divided into two classes, namely normal brain and brain with tumors. Data preprocessing is done with augmentation techniques to reproduce the results of the original image. The modified architecture test was carried out with the results of an accuracy of 88.00%, sensitivity of 90.00%, specificity of 86.00%. Then testing using K-Fold Cross Validation obtained the results for the modified architectural model obtained an average accuracy of 94.40%, an average sensitivity of 92.18%, and an average specificity of 96,27%.

الملخص

فكر الدين ، محمد يسريل. 2022. تعديل هندسة لينت –٥ للكشف عن أورام الدماغ بناءً على صورة التصوير بالرنين المغناطيسي. قسم هندسة المعلومات لكلية العلوم والتكنولوجيا في جامعة مولانا مالك إبراهيم الإسلامية الحكومية بمالانق. المشرف: (١) اروان بودي سانتوسو ، املاجستر(٢) اوكتا قمر الدين عزيز، املاجستر

الكلمات الرئيسية: ورم الدماغ ، الشبكة العصبية التلافيفية ،لينت –٥، التعديل المعماري

يعتبر ورم الدماغ من الأمراض الفتاكة التي تصيب الدماغ البشري حيث يحدث هذا المرض نتيجة نمو الخلايا غير الطبيعية في الدماغ. تظهر هذه الأعراض المبكرة عندما يعاني الإنسان من الصداع والتشنجات والتعب والغثيان. من خلال مراقبة هذه الأعراض ، يمكن التعرف على الأمراض على الفور وتنفيذ الوقاية المبكرة من خلال بناء نظام يمكنه تحديد كائن من خلال تنفيذ خوارزمية الشبكة العصبية التلافيفية باستخدام هندسة Lenet - 5 بواسطة Yann Lecun. تهدف هذه الدراسة إلى قياس نتائج دقة وحساسية وخصوصية العمارة قبل التعديل وبعده. مجموعة البيانات المستخدمة عبارة عن بيانات ثانوية تتكون من 500 صورة مقسمة إلى فئتين ، وهما الدماغ الطبيعي والدماغ المصاب بالأورام. تتم المعالجة المسبقة للبيانات باستخدام تقنيات التعزيز لإعادة إنتاج نتائج الصورة الأصلية. تم إجراء اختبار العمارة المعدل بنتائج بدقة 88.00٪ وحساسية 90.00٪ ونوعية 86.00٪. ثم حصل الاختبار باستخدام K-Fold Cross Validation على نتائج النموذج المعماري المعدل التي حصلت على متوسط دقة 94.40٪ ، وحساسية متوسطة 92.18٪ ، ومتوسط خصوصية 96.27٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Otak manusia merupakan salah satu organ yang paling penting. Otak berfungsi sebagai pusat sistem saraf pada manusia. Otak terdiri lebih dari 100 miliar sel saraf dimana setiap neuron dapat berkomunikasi dengan ribuan neuron lainnya untuk membangun komunikasi dan kontrol jaringan yang kompleks. Ada sedikit kerusakan pada otak, maka tidak bisa ber-aktivitas dengan normal, Oleh karena itu, menjaga kesehatan otak sangatlah penting. Adapun salah satu penyakit otak yaitu tumor atau kanker. Tumor ataupun kanker otak merupakan perkembangan sel- sel abnormal yang terdapat pada jaringan otak, tumor otak ialah penyakit yang beresiko untuk manusia sebab bisa menimbulkan kematian. Tumor otak meliputi 85- 90% dari segala kanker lapisan saraf pusat (KPKN Indonesia, 2016). Menurut data Badan Kesehatan Dunia (WHO) pada tahun 2012, negara tersebut memiliki sekitar 14 juta kasus kanker baru dan sekitar 8,2 juta orang (58,57%) di antaranya meninggal akibat kanker. Tumor otak terbagi menjadi dua jenis ialah tumor jinak serta ganas. Tumor jinak hanya tumbuh di satu bagian tubuh dan tidak menyebar atau menyerang bagian lain. Sementara tumor ganas sering disebut sebagai kanker, itu adalah tumor yang dapat menyerang jaringan di sekitarnya dan menyebar ke bagian tubuh yang lain.

Gejala tumor otak dapat bervariasi dari setiap orang. Gejala pada penderita tumor otak tergantung pada lokasi dan kecepatan pertumbuhan tumor. Banyak hal

yang dapat mempengaruhi tumor, dari sakit kepala, mual, kehilangan nafsu makan, muntah, kejang, kelumpuhan anggota badan, perubahan kepribadian dan mental.

Untuk dapat mengidentifikasi jenis tumor otak dibutuhkan teknologi yang dapat memvisualisasikan citra otak, salah satunya yaitu *Brain Magnetic Resonance Imaging* (MRI otak). *Magnetic Resonance Imaging* (MRI) adalah suatu teknik pemeriksaan diagnostik imaging dengan menggunakan medan magnet yang kuat dan gelombang radio untuk menghasilkan pencitraan struktur internal tubuh secara detail. MRI secara khusus berguna untuk melihat jaringan lunak tubuh. Pemeriksaan MRI digunakan untuk menghasilkan citra organ, tulang, atau jaringan lunak tubuh, termasuk sistem saraf. Pemindaian MRI yang mendiagnosis tumor otak dapat memberikan informasi pencitraan yang lebih edukatif bila ditambahkan ke pemindaian MRI yang lebih canggih, seperti spektroskopi resonansi magnetik (MRS). (Hulmansyah, 2020).

Dalam mengidentifikasi hasil MRI tumor otak digunakan metode menghitung secara otomatis dan akurat data yang ada. Salah satu metode yang digunakan adalah *deep learning,* yang bagian dari kecerdasan buatan dimana pengembangan jaringan saraf berlapis-lapis yang melakukan tugas-tugas yang tepat seperti pengenalan objek, pengenalan ucapan, terjemahan bahasa, dan lainnya. D*eep learning* berbeda dari teknik *machine learning* tradisional karena secara otomatis merepresentasikan data seperti gambar, video, atau teks tanpa aturan pengkodean atau pengetahuan jaringan manusia. (Iqbal et al., 2018).

Dalam implementasi *deep learning*, salah satu model yang digunakan dalam menghitung akurasi yang baik yaitu *LeNet-5*, merupakan model arsitektur yang sederhana dan penggunaan *layer* yang tergolong tidak terlalu banyak. Model tersebut merupakan salah satu dari *Convolutional Neural Network* (CNN) (Alwanda *et al*., 2020).

Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang biasa digunakan dalam pengolahan data citra. CNN dapat digunakan untuk mendeteksi dan mengidentifikasi objek pada citra. CNN menggunakan proses konvolusi dengan menerapkan kernel konvolusi (filter) dengan ukuran tertentu pada citra, dan komputer memperoleh informasi representatif baru dengan mengalikan bagian citra dengan filter yang digunakan. *Convolutional Neural Network* (CNN) merupakan pengembangan lebih lanjut dari *Multilayer Perceptro*n (MLP) untuk mengolah data dua dimensi dalam bentuk citra. CNN tidak terlalu berbeda dari jaringan saraf normal. CNN terdiri dari neuron dengan fungsi bobot, bias, dan aktivasi. CNN memiliki berbagai model aristetur lainnya seperti VGG16, LeNet-5, dan Lu-Net (Nugroho *et al*., 2020).

Salah satu peneliti sebelumnya tentang tumor otak pada tahun 2020 yaitu Rai dan Chatterjee membuat penelitian tentang identifikasi dan klasifikasi tumor berdasarkan citra MR dimana metode yang digunakan adalah jaringan *Deep Neural* dengan jumlah lapisan yang lebih sedikit dan desain yang lebih kompleks bernama U-Net (LU-Net) untuk mendeteksi tumor. Pengerjaan ini terdiri dari mengklasifikasikan citra MR otak ke dalam kelas normal dan abnormal dari Kinerja model *Lu-Net* dievaluasi menggunakan lima jenis metrik penilaian

statistik *Precision, Recall, Specificity, F-score*, dan *Accuracy*, kemudian dibandingkan dengan dua jenis model *Le-Net* dan *VGG-16* lainnya. Keakuratan keseluruhan dari model yang digunakan yaitu *Le-Net*, *VGG-16*,dan *U-Net* adalah masing-masing 88%, 90%, dan 98% (Rai & Chatterjee, 2020).

Zhang (2020) pada penelitiannya yang berjudul "The Detection of Hyperthyroidism by the Modified LeNet-5 Network" melakukan sebuah modifikasi pada LeNet-5 dengan mengubah beberapa parameter seperti mengubah jumlah dan ukuran kernel konvolusi dengan tepat, mengganti fungsi aktifasi sigmoid dengan fungsi aktifasi *ReLu* dan *Fully Connected Layer* dengan *Pooling layer* bersifat global. Hasil deteksi menunjukkan bahwa tingkat keberhasilan jaringan *LeNet-5* yang ditingkatkan atau di modifikasi lebih tinggi daripada jaringan *LeNet-5* sebelum di modifikasi. Efisiensi deteksi *LeNet-5* algoritma jaringan sekitar 3 kali lebih tinggi dan sensitivitasnya terhadap hipertiroidisme juga secara signifikan lebih tinggi, yang menunjukkan bahwa algoritma jaringan *LeNet-5* dapat mendeteksi hipertiroid secara efisien dan cepat (Zhang et al, 2020).

Dalam penelitian ini akan mengusulkan tentang modifikasi model LeNet-5 untuk mengidentifikasi tumor otak pada MRI. Arsitektur yang akan dimodifikasi menggunakan arsitektur *LeNet-5* klasik berdasarkan dengan jurnal yang asli oleh Lecun (1998), modifikasi yang diusulkan yaitu penambahan jumlah konvolusi layer, perubahan ukuran input citra dan penambahan nilai filter agar didapatkan performa akurasi yang lebih tinggi. Modifikasi dilakukan karena jaringan klasik LeNet-5 sebelumnya memiliki sedikit jumlah kernel konvolusi per lapisan, Fungsi Tanh diadopsi oleh *LeNet-5* klasik tidak universal dan fungsi *ReLu* memiliki

karakteristik mencegah suatu nilai lewat dari sinyal negatif, yang lebih efisien dan dapat meningkatkan kecepatan konvergensi (Zhang et al, 2020).

Peningkatan jumlah parameter ini disebabkan oleh meningkatnya kecepatan atau teknologi komputer yang memungkinkannya melakukan perhitungan matematis dengan lebih cepat. (Fitriati, 2016).

Penelitian ini diharapkan dapat bermanfaat bagi peneliti lain dalam pengembangan study lebih lanjut. Sesuai dengan firman Allah SWT dalam surat al-Zalzalah:7 berbunyi:

فَمَن يَعْمَلْ مِثْقَالَ ذَرَّةٍ خَيْرًا يَرَهُ

"Maka barang siapa yang mengerjakan kebaikan sebesar dzarrah-pun, ia akan mendapatkan balasannya " (QS al-Zalzalah: 7).

1.2 Pernyataan Masalah

Seberapa besar pengukuran tingkat performa (akurasi, sensitivitas, dan spesifisitas) sistem dalam mendeteksi tumor otak pada MRI menggunakan model modifikasi arsitektur LeNet-5?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengukur performa sistem (akurasi, sensitivitas dan spesifisitas) dalam mendeteksi tumor otak pada MRI menggunakan model arsitektur LeNet-5 yang dimodifikasi..

1.4 Batasan Penelitian

1. Citra yang digunakan dalam penelitian ini adalah citra MRI otak manusia yang terdiri dari otak normal dan otak tumor.

2. Data yang diambil dalam penelitian ini adalah data sekunder MRI scan otak yang dapat diakses secara publik melalui *website* Kaggle.com.
3. MRI scan otak yang diidentifikasi yaitu MRI otak normal dan MRI tumor otak.

1.5 Manfaat Penelitian

1. Dapat membantu dalam bidang kesehatan untuk mendeteksi penyakit tumor otak
2. Dapat menjadi referensi penelitian lebih lanjut mengenai modifikasi arsitektur convolutional neural network..

BAB II

STUDI LITERATUR

2.1 Convolutional Neural Network

Fitriati (2016) melakukan penelitian menggunakan perbandingan dua metode yang berbeda yaitu Metode *Convolutional Neural Nework* (CNN LeNet 5) dan Mesin Pembelajaran Ekstrim (ELM). Pada penelitian ini eksperimen dilakukan pada data sederhana, dengan gambar yang digunakan berupa angka tulisan tangan, dengan menggunakan dua jenis data primer sebanyak 700 citra angka dan data sekunder sebanyak 10.000 citra. Pada penelitian ini akurasi metode CNN adalah 98,04% untuk data sekunder dan 78,14% untuk data primer. Sedangkan metode ELM lebih baik dari segi waktu komputasi, namun mencapai 0,00078 milidetik.

Alwanda *et.al.* (2020) melakukan penelitian menggunakan arsitektur LeNet-5 untuk pengenalan jenis doodle dengan 5 objek gambar. *Dataset* yang digunakan sebanyak 150 citra. Penelitian ini berhasil memperoleh nilai akurasi sebesar 94%.

Rivan dan Riyadi (2021) melakukan penelitian menggunakan algoritma CNN dengan arsitektur yang digunakan yaitu LeNet dan AlexNet. *Dataset* yang digunakan sebanyak 100 data per huruf untuk skema pertama dan 1000 data per huruf skema kedua. Penelitian ini berhasil memperoleh nilai akurasi sebesar 48,332% untuk arsitektur LeNet dan 32,584% untuk arsitektur AlexNet untuk skema pertama. Untuk skema kedua memperoleh nilai akurasi sebesar 92,468% untuk arsitektur LeNet dan 91,618% untuk arsitektur AlexNet.

Alwanda *et.al.* (2021) melakukan penelitian menggunakan algoritma CNN arsitektur VGG16. *Dataset* yang digunakan yaitu 240 *sample* data video terdiri atas bahasa Inggris dan Kanada. Penelitian ini berhasil memperoleh nilai akurasi 90.10% untuk bahasa Inggris dan 91,90% untuk bahasa Kanada.

Santoso *et.al*. (2021) melakukan penelitian menggunakan algoritma CNN untuk mengklasifikasikan setiap segmen sinyal EEG pada setiap saluran uji. *Dataset* yang digunakan terdiri dari lima record EEG berlabel A, B, C, D, dan E yang masing-masing memiliki 100 data. Penelitian ini berhasil memperoleh nilai akurasi yaitu 99,33%, 100%, 100%, 99,5%, 99,8%, dan 99,4% untuk AB-C, AB-D, AB-E, AB-CD, AB- CDE, dan AB-CD-E.

2.2 Identifikasi Tumor Otak

Daimary et al. (2020) melakukan penelitian menggunakan metode tiga model CNN hybrid, yaitu U-SegNet, Res-SegNet, dan Seg-UNet yang dirancang untuk keandalan segmentasi otomatis tumor otak dari citra MRI dengan akurasi tinggi. Model yang diusulkan mewarisi properti SegNet, U-Net, dan ResNet, yang merupakan model CNN paling populer untuk segmentasi semantik. Data yang digunakan berasal dari *Dataset* BraTS yang diperoleh dari portal CBICA. Dari hasil percobaan didapatkan bahwa arsitektur hybrid yang diusulkan menghasilkan keluaran yang lebih akurat dibandingkan dengan model CNN eksisting lainnya. U-SegNet, Res-SegNet dan Seg-UNet masing-masing mencapai akurasi rata-rata 91.6%, 93.3% dan 93.1%.

Gu *et.al.* (2021) melakukan penelitian menggunakan convolutional dictionary learning with local constraint (CDLLC). Metode ini mengintegrasikan

pembelajaran kamus multi-layer ke dalam struktur jaringan saraf convolutional (CNN). Penelitian ini memperoleh hasil akurasi sekitar 96% .

Hao dong *et.al.*(2018) melakukan penelitian tentang deteksi tumor otak otomatis berdasarkan segmentasi MRI menggunakan jaringan konvolusi berbasis U-Net, kemudian di evaluasi dengan *Dataset Multimodal Brain Tumor Image Segmentation* (BRATS 2015), yang berisi 220 tumor otak tingkat tinggi dan 54 kasus tumor tingkat rendah. Penggunaan validasi silang telah menunjukkan bahwa metode tersebut dapat memperoleh segmentasi yang menjanjikan secara efisien.

Ming niwu *et.al.* (2018). melakukan penelitian tentang deteksi tumor berdasarkan segmentasi warna dimana metode yang digunakan menggunakan *K-Means*. Konsep utama dalam algoritma segmentasi berbasis warna ini dengan *K-Means* adalah untuk mengubah citra MR tingkat abu-abu yang diberikan menjadi citra ruang warna dan kemudian memisahkan posisi objek tumor dari *item* citra MR lainnya dengan menggunakan pengelompokan *K-Means* dan pengelompokan secara histogram. Metode yang diusulkan hanya menggabungkan kombinasi terjemahan warna, pengelompokan *K-Means* dan pengelompokan histogram, sehingga membuatnya efisien dan sangat mudah untuk diterapkan

Rai & Chatterjee (2020) melakukan penelitian tentang identifikasi dan klasifikasi tumor berdasarkan citra MR pada pikiran manusia dimana metode yang digunakan adalah jaringan *Deep Neural* dengan jumlah lapisan yang lebih sedikit dan desain yang lebih kompleks bernama U-Net (LU-Net) untuk mendeteksi tumor. Pengerjaan ini terdiri dari mengklasifikasikan citra MR otak ke dalam kelas normal dan abnormal dari kumpulan data 253 citra piksel tinggi. citra MR

pertama diubah ukurannya, kemudian ukurannya dipotong, diproses sebelumnya, dan ditambah untuk pelatihan model saraf dalam yang akurat dan cepat. Kinerja model Lu-Net dievaluasi menggunakan lima jenis metrik penilaian statistik *Precision, Recall, Specificity, F-score*, dan *Accuracy*, kemudian dibandingkan dengan dua jenis model Le-Net dan VGG-16 lainnya. Penjelasan masing-masing peneliti terdahulu ditabelkan pada Tabel 2.1.

Tabel 2.1 Perbedaan Dengan Para Peneliti Sebelumnya

No	**Peneliti**	**Metode dan Studi Kasus**
1.	Daimary *et al.* (2020)	- Menggunakan metode hybrid CNN - Citra input berdimensi 240x240x3 - Convulutin layer 1 berdimensi 240x240x64 - Convulutin layer 2 berdimensi 120x120x128 - Convulutin layer 3 berdimensi 60x60x256 - Convulutin layer 4 berdimensi 30x30x512 - Convulutin layer 5 berdimensi 15x15x512 - Final Convulutin layer berdimensi 240x240x64 - Akurasi metode hybrid CNNU-SegNet, Res-SegNet dan Seg-UNet masing-masing mencapai akurasi rata-rata 91.6%, 93.3% dan 93.1%.
2.	Gu *et.al.,* (2021)	- Menggunakan metode convolutional dictionary learning with local constraint (CDLLC) yang mengintegrasikan multi layer CNN - Citra input berdimensi 227x227 - Jumlah data sebanyak 1000 data citra - Semua convulution layer berdimensi 3x3 - Akurasi sekitar 96%

3.	Rai & Chatterjee, (2020)	- Metode yang digunakan adalah jaringan *Deep Neural* dengan jumlah lapisan yang lebih sedikit dan desain yang lebih kompleks bernama U-Net (LU-Net) untuk mendeteksi tumor. - Jumlah data sebanyak 253 data citra - Citra input berdimensi 224x224x3 - Keakuratan keseluruhan model Le-Net, VGG-16 dan Usulan yang diterima adalah 88%, 90%, dan 98% masing-masing.

BAB III

METODE PENELITIAN

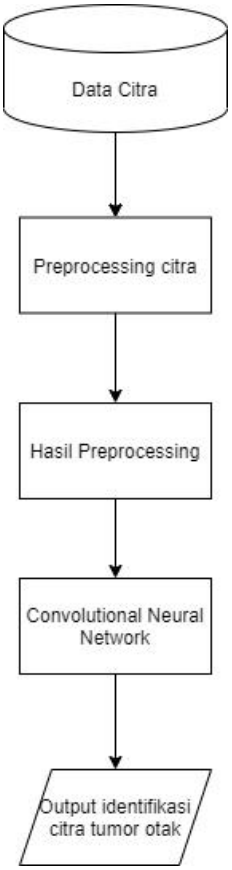
3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari Sartaj (2020) melalui website Kaggle.com. Data ini berupa sekumpulan citra yang kemudian disebut *Dataset* . *Dataset* ini terdiri dari 500 file dengan tipe citra format .jpg yang terbagi atas 2 kategori yaitu, citra otak normal dan citra tumor otak. Dalam hal ini adalah ukuran citra input yang digunakan dalam penelitian ini berukuran 256x256 piksel.

Kumpulan data ini dibagi menjadi data pelatihan dan data uji. Kemudian penelitian ini menggunakan metode *testing* yang disebut *K-Fold Cross Validation.* Kemudian dari hasil *K-Fold Cross Validation* akan dirata-rata seluruh *fold* sesuai dengan tingkat performa masing-masing yaitu akurasi, sensitifitas, dan spesifitas.

3.2 Desain Sistem

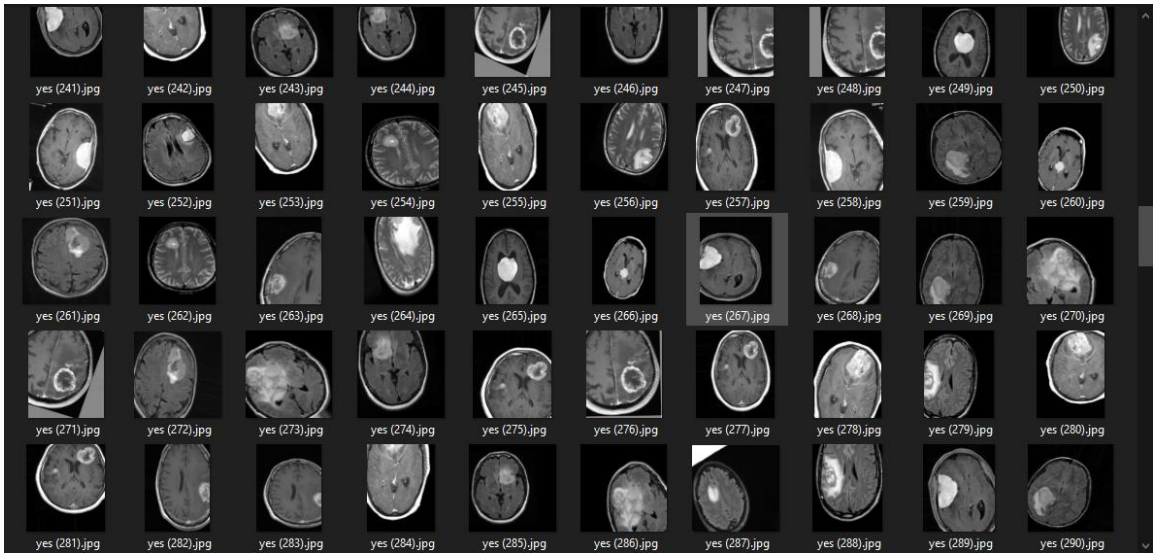
Pada penelitian ini dibuatkan desain sistem untuk mengidentifikasi tumor otak yang meliputi data citra, *image preprocessing*, kemudian hasil *image preprocessing,* penerapan model CNN, dan output hasil identifikasi tumor otak, yang mana pada gambar 3.1.



Gambar 3.1 Desain Sistem

3.3 *Image Preprocessing*

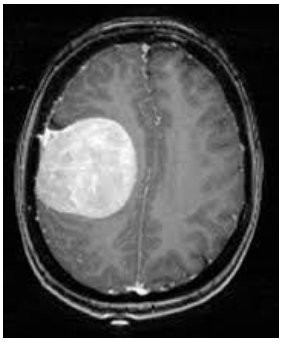
Pada tahap ini dilakukan penambahan data citra agar algoritma mendapatkan hasil optimal. Salah satu metode menambah jumlah data dengan cara augmentasi citra. Dengan augmentasi citra, data dapat diperbanyak tanpa merusak data aslinya. Adapun parameter dalam augmentasi citra seperti *shear_range* = 0,2, *zoom_range* = 0,2, *horizontal_flip* = true, *rotation* = 25, *width_shift* = 0,2, *height_shift* = 0,2 (Mubarok, 2019). Adapun hasil dari parameter augmentasi citra tersebut di gambar 3.2.



Gambar 3.2 *Dataset* Augmentasi Citra Tumor Otak

1. *Shear Range*

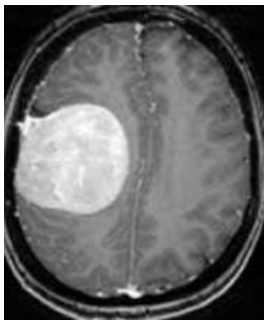
   Merupakan teknik merubah posisi sumbu citra atau memiringkan citra pada sudut tertentu, hal ini untuk memperbaiki satu sumbu dan meregangkan citra pada sudut tertentu yang dikenal sebagai transformasi geser. Adapun nilai s*hear_range* = 0.2. Berikut hasil *shear range* pada gambar 3.3.



Gambar 3.3 Teknik *Shear*

2. *Zoom Range*

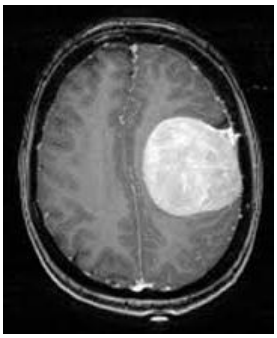
Merupakan teknik memperbesar citra sesuai dengan jarak tertentu. Jarak yang digunakan pada program ini yaitu zoom_range = 0.2. Berikut hasil untuk *zoom range* pada gambar 3.4.



Gambar 3.4 Teknik *Zoom*

3. *Horizontal Flip*

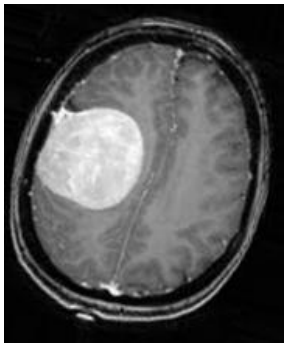
Merupakan teknik membalikkan suatu citra secara horizontal. Berikut hasil untuk *horizontal flip* pada gambar 3.5.



Gambar 3.5 Teknik *Horizontal*

4. *Rotation*

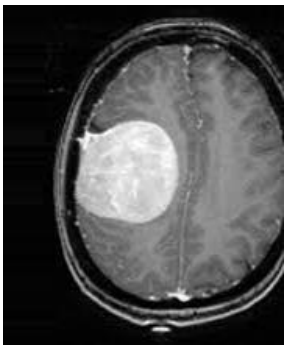
Teknik ini akan melakukan rotasi pada citra. Rotasi disesuaikan dengan rentang nilai (*range*) derajat yang ditentukan. Dalam penelitian digunakan rotation_range = 25. Berikut hasil *rotation* pada gambar 3.6.



Gambar 3.6 Teknik *Rotation*

5. *Width Shift*

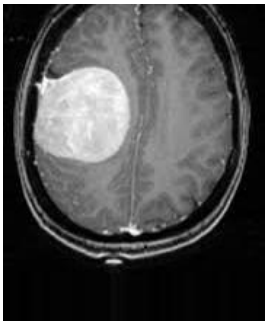
Teknik ini akan melakukan pergeseran pada citra secara horizontal. Rentang nilai (*range*) pergeseran ditentukan berdasarkan persentase dari total lebar citra. Dalam penelitian ini citra input berukuran 256 x 256 piksel dan *width_shift_range* = 0.2. Artinya pergeseran akan dilakukan -20% sampai +20% dari lebar gambar atau -25px sampai +25px. Citra akan digeser secara acak sesuai dengan rentang nilai tersebut. Apabila nilai yang terpilih bernilai positif maka citra akan bergeser ke kanan dan apabila nilai yang terpilih bernilai negatif maka citra akan bergeser ke kiri. Berikut hasil *width shift* pada gambar 3.7.



Gambar 3.7 Teknik *Width Shift*

6. *Height Shift*

Teknik ini akan melakukan pergeseran pada citra secara vertikal . Dalam penelitian ini citra input berukuran 256 x 256 piksel dan *height_shift_range* = 0.2. Artinya pergeseran akan dilakukan -20% sampai +20% dari lebar gambar atau -25px sampai +25px. Citra akan digeser secara acak sesuai dengan rentang nilai tersebut. Citra akan digeser secara acak sesuai dengan rentang nilai tersebut. Apabila nilai yang terpilih bernilai positif maka citra akan bergeser ke atas dan apabila nilai yang terpilih bernilai negatif maka citra akan bergeser ke bawah. Berikut hasil *height shift* pada gambar 3.8.



Gambar 3.8 Teknik *Height Shift*

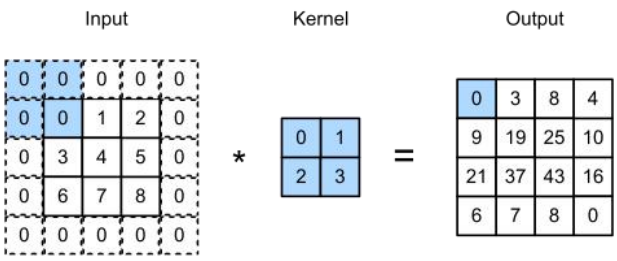
3.4 Implementasi *Convolutional Neural Network*

Tahapan implementasi CNN pada penelitian ini menggunakan alat *Google Collaboratory*. *Google Collaboratory* adalah alat yang dikembangkan oleh Google untuk tujuan penelitian ilmu data seperti kecerdasan buatan, pembelajaran mesin, dan analisis data.. Dalam implementasi model pada Google Colaboratory menggunakan bahasa python untuk proses t*raining* dan *testing* melalui browser. Pada Google Colaboratory, data tersebut dapat di proses secara online melalui

sistem *cloud* sehingga dalam proses implementasi tidak perlu menggunakan komputer lokal dengan proses yang berat, cukup dengan koneksi yang stabil untuk menggunakannya. Kemudian pembuatan desain modifikasi arsitektur LeNet-5, arsitektur dari LeNet-5 dibagi menjadi 2 bagian, yaitu *Feature Extraction Layer* dan *Fully-Connected Layer* (MLP). Lapisan ekstraksi fitur terdiri dari dua bagian yaitu lapisan konvolusi, lapisan pooling (*pooling layer*) dan juga fungsi aktivasi.

3.4.1 Layer Konvolusi

Lapisan konvolusi menghasilkan gambar baru dengan menampilkan fitur dari gambar masukan. Lapisan konvolusi menerapkan filter ke setiap gambar masukan. Lapisan konvolusi terdiri dari neuron tersusun membentuk filter. Filter pada layer ini berbentuk *array* dua dimensi. Setiap pergerseran layer melakukan suatu operasi antara input filter dan nilai untuk menghasilkan output yang dikenal sebagai *feature map*. Ilustrasinya pada gambar 3.9*:*



Gambar 3.9 Ilustrasi Layer Konvolusi (sumber:Kadam, 2021)

Pada Gambar 3.9 merupakan ilustrasi *Convolution Layer* untuk citra dua dimensi dengan kernel 2x2 maka setiap kali akan melakukan suatu konvolusi, maka hasil dari perkalian matriks dengan kernel 2x2 akan ditempatkan sesuai dengan indeks yang ditentukan dalam *feature map*.

Kemudian pergeseran kernel ditentukan dengan banyaknya jumlah *stride* yang ditentukan. Adapun rumus yang digunakan untuk melakukan perhitungan matriks konvolusi terhadap citra dua dimensi sebagai berikut:

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \quad (3.1)$$

Keterangan:

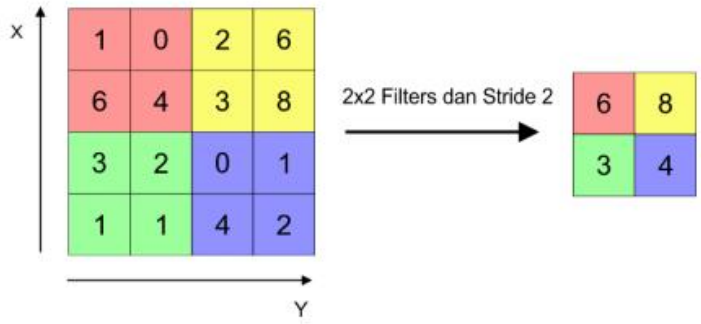
f = citra input

h = kernel

m, n = index matriks hasil konvolusi

3.4.2 *Pooling Layer*

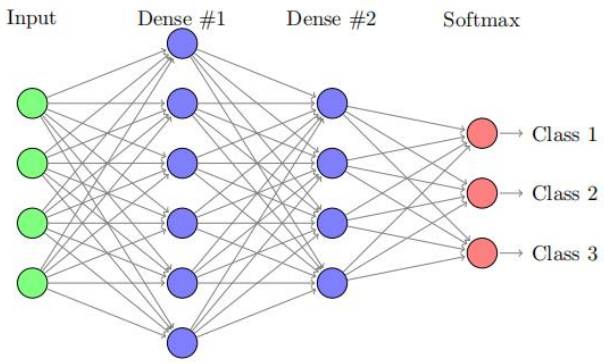
Terdiri dari filter dengan ukuran dan nilai tertentu. Setiap pergeseran ditentukan oleh jumlah *strides* yang harus dipindahkan melalui *feature map.* Lapisan pooling yang umum digunakan adalah m*ax Pooling* dan a*verage Pooling*. Saat menggunakan m*ax pooling* 2x2 dan *stride* 2, nilai yang diambil di setiap pergeseran filter adalah nilai terbesar dalam rentang 2x2, sedangkan a*verage pooling* mengambil rata-rata.(Santoso & Aryanto, 2018). Proses ini di ilustrasikan pada gambar 3.10.



Gambar 3.10 Ilustrasi *Pooling layer* (sumber: Santoso & Aryanto, 2018).

3.4.3 *Fully-Connected Layer*

Lapisan yang biasa digunakan untuk mengklasifikasikan gambar input ke dalam kelas yang berbeda berdasarkan banyaknya kelas data. Setiap neuron pada layer konvolusi terlebih dahulu harus dikonversi menjadi data satu dimensi sebelum dapat dimasukkan ke dalam *fully connected layer* (Ilahiyah & Nilogiri, 2018). Dapat diilustrasikan pada gambar 3.11.



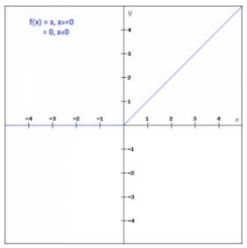
Gambar 3.11 Ilustrasi *Fully-Connected Layer* (sumber: Pelletier, 2019)

3.4.4 Fungsi Aktivasi

Fungsi Aktivasi secara khusus digunakan dalam saraf tiruan jaringan untuk mengubah sinyal input menjadi sinyal output. Keakuratan prediksi jaringan saraf ditentukan oleh jenis fungsi aktivasi yang digunakan dan jumlah lapisan yang digunakan. *ReLU* adalah singkatan dari *Rectified Liner Unit* dan merupakan fungsi aktivasi nonlinier yang banyak digunakan dalam jaringan saraf. Keuntungan menggunakan fungsi *ReLU* adalah tidak semua neuron diaktifkan secara bersamaan. Ini berarti bahwa neuron hanya dinonaktifkan ketika output

dari transformasi linier adalah nol (Sharma *et,al.,* 2020). Adapun rumus dan ilustrasi untuk fungsi aktifasi *ReLu* pada gambar 3.12 dan Persamaan 3.2.

$$R(z) = max\,(0, z)$$



Gambar 3.12 Fungsi *ReLu* (sumber: S. Sharma et,al., 2020).

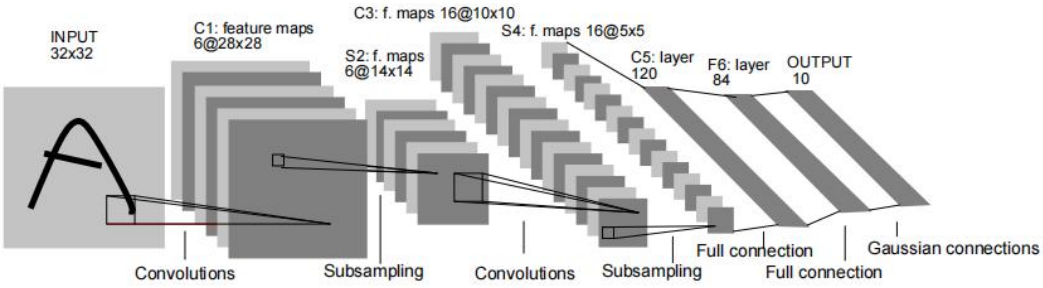
3.5 Modifikasi Model Arsitektur LeNet-5 Usulan

Pada penelitian ini arsitektur yang akan dimodifikasi diambil dari salah satu penelitian dari LeCun et al., (1998).

Adapun perubahan arsitektur original dan modifikasi terletak pada fungsi aktivasi pada layer C1, C2, dan C3 dari fungsi tanh ke fungsi *ReLU*. Karena fungsi *ReLU* memiliki perhitungan atau rumus yang sederhana.Fungsi aktivasi ini cenderung mengubah nilai negatif menjadi nol (menghilangkan nilai negatif dari matriks konvolusi) tidak ada operasi eksponensial, perkalian atau pembagian seperti *sigmoid* atau *tanh*, sehingga dapat mengurangi waktu komputasi (Sanjaya & Ayub, 2020).

Kemudian adanya penambahan layer setelah layer C3 yaitu *Pooling layer*, penambahan ini agar ada perbedaan dari sisi original dan modifikasi. Selain itu. *Pooling layer* untuk mengurangi pixel dari feature map dari layer C3, ini dapat mempercepat perhitungan karena ada lebih sedikit parameter untuk diperbarui dan *overfitting* untuk diatasi. *Pooling layer* yang digunakan adalah jenis *Max Pooling*

karena memiliki rumus yang sederhana karena *Max Pooling* hanya mencari nilai maksimum dari tiap matriks, hal ini akan mempercepat proses komputasi. Sedangkan untuk *Average Pooling* mencari nilai rata-rata dari tiap matriks yang telah dilewati oleh setiap kernel, sehingga proses komputasi akan sedikit lebih lama dibanding *Max Pooling* (Sanjaya & Ayub, 2020). Selanjutnya untuk output dari *fully connected* menggunakan fungsi aktivasi sigmoid dimana fungsi ini untuk klasifikasi dengan output 0 dan 1. Adapun model arsitektur dari LeNet-5 original pada gambar 3.13.



Gambar 3.13 Model Arsitektur LeNet-5 (sumber: LeCun et al., 1998)

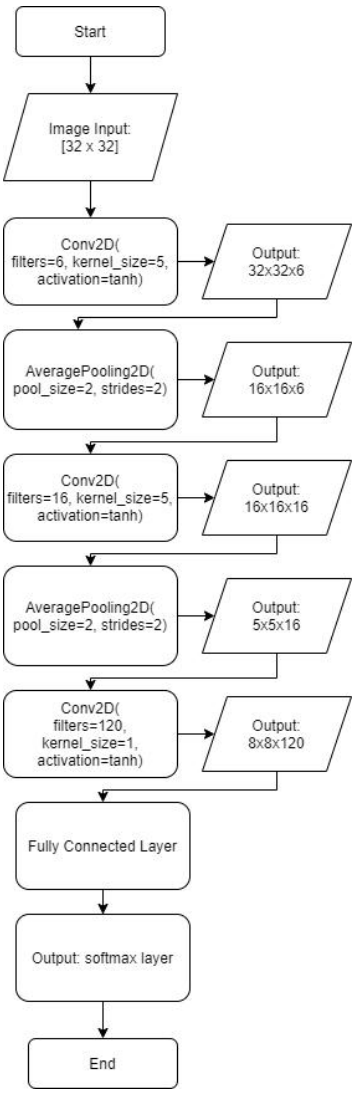
Pada Gambar 3.13 dapat dijelaskan bahwa struktur tiap layer dari *convolution layer, pooling layer*, dan fungsi aktivasi yang digunakan dibuatkan pada Tabel 3.1 sebagai berikut:

Tabel 3.1 Arsitektur CNN Original (Yann LeCun)

Layer	**Karakteristik**	**Filter**	**Feature Map**	**Kernel Size**	**Stride**
Input	Dimensi citra	1	32x32	-	-
1	-*Convolution layer* (C1)	6	32x32x6	5x5	1
2	Average Pooling	6	16x16x6	2x2	2

3	-Convolution layer (C2)	16	16x16x16	5x5	1
4	Average Pooling	16	5x5x16	2x2	2
5	-Convolution Layer (C3)	120	8x8x120	1x1	-
6	Fully Connected	-	-	84	-
7	-Fully Connected -Softmax (Output)	-	-	10	-

Input yang digunakan berukuran 32x32 dengan tiga layer konvolusi dan dua layer pooling dengan output menggunakan fungsi *Softmax,* dimana layer C1 menggunakan filter = 6 dan kernel = 5x5, kemudian layer kedua menggunakan *Average Pooling* dengan ukuran filter = 6, kernel = 5x5, dan *stride* = 2 menghasilkan *feature map* berukuran 16x16x6, pada layer ketiga yaitu C2 menggunakan filter = 6 dan kernel = 5x5 dan *stride* = 1 menghasilkan *feature map* berukuran 16x16x16, kemudian layer keempat menggunakan *Average Pooling* dengan ukuran filter = 16, kernel = 2x2, dan *stride* = 2 menghasilkan *feature map* berukuran 5x5x16, pada layer kelima yaitu C3 menggunakan filter = 120 dan kernel = 1x1 menghasilkan menghasilkan *feature map* berukuran 8x8x120, dan terakhir untuk output layer menggunakan fungsi aktivasi *softmax*. Alur proses algoritma yang diilustrasikan pada gambar 3.14.



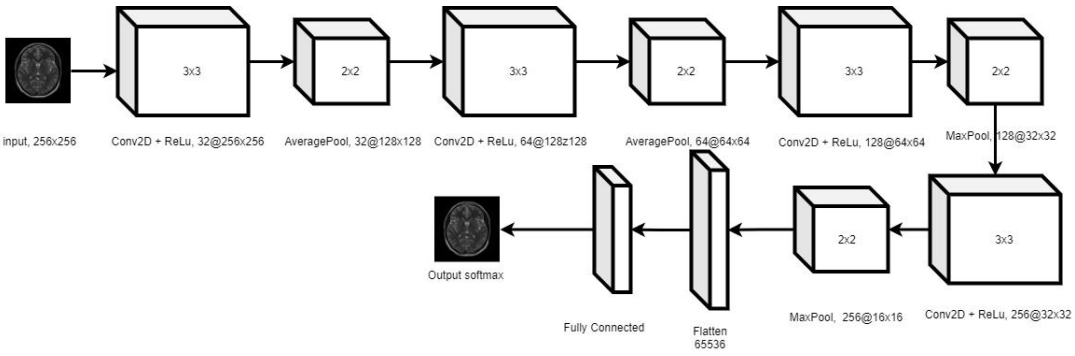
Gambar 3.14 *Flowchart* Arsitektur Original

Kemudian kode program dari model arsitektur original pada gambar 3.15.

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Conv2D(filters=6,kernel_size=5, activation = 'tanh', input_shape = [32,32,3]))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=16,kernel_size=5, activation = 'tanh'))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=120,kernel_size=1, activation = 'tanh'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(units=120, activation='relu'))
model.add(tf.keras.layers.Dense(units=1, activation='softmax'))
model.summary()
```

Gambar 3.15 Kode Program Arsitektur Original

Pada Gambar 3.15 adalah kode program yang digunakan dalam membangun arsitektur modifikasi. Fungsi *model.add()* digunakan untuk menginisialisasi layer konvolusi*, pooling layer,* fungsi aktifasi, dan *fully connected layer* yang telah dirancang sebelumnya. Selanjutnya untuk model aristektur LeNet-5 modifikasi LeNet-5 pada Gambar 3.16.



Gambar 3.16 Model Arsitektur Modifikasi

Pada Gambar 3.16 dapat dijelaskan bahwa struktur tiap layer dibuatkan dalam bentuk Tabel 3.2 sebagai berikut.

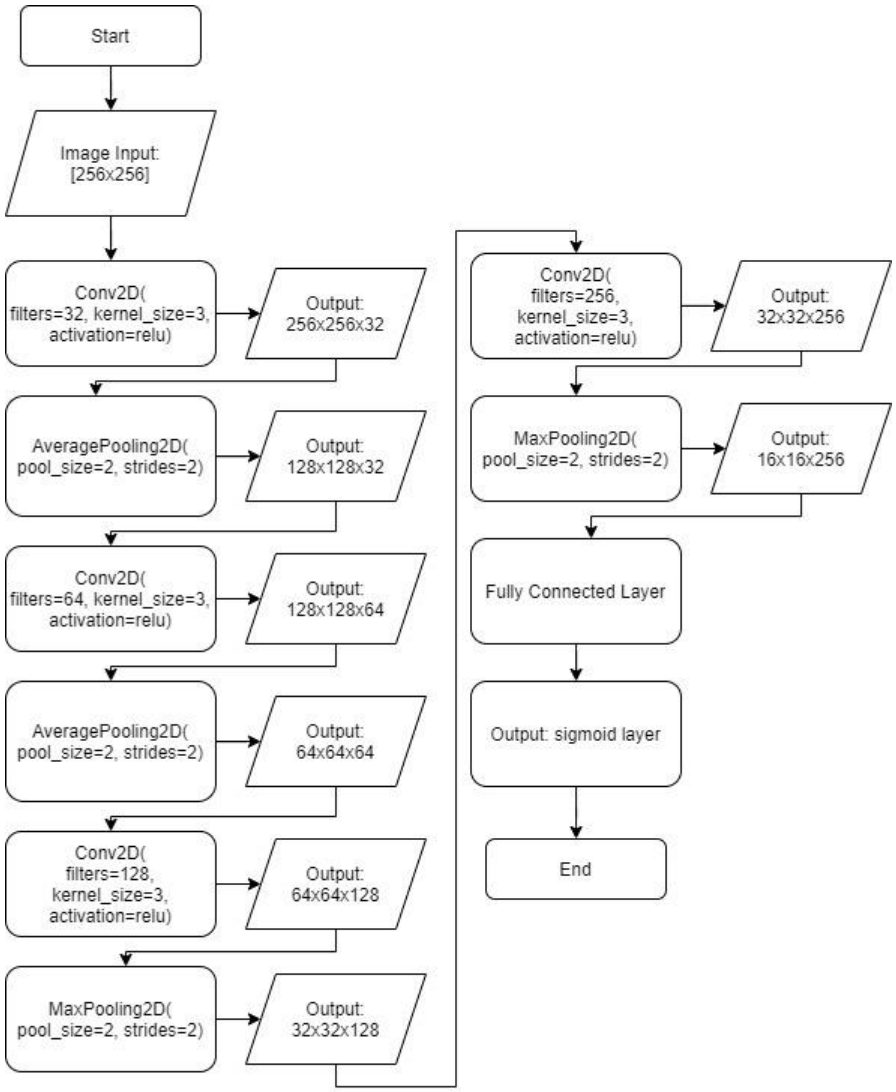
Tabel 3.2 Arsitektur CNN Modifikasi

Layer	**Karakteristik**	**Filter**	**Feature Map**	**Kernel Size**	**Stride**
Input	Dimensi citra	1	256x256	-	-
1	-*Convolution layer* (C1)	32	256x256x32	3x3	1
2	Average Pooling	32	128x128x32	2x2	2
3	-Convolution layer (C2)	64	128x128x64	3x3	1
4	Average Pooling	64	64x64x64	2x2	2
5	-Convolution Layer (C3)	128	64x64x128	3x3	1

6	Max Pooling	128	32x32x128	2x2	2
7	-Convolution Layer (C4)	256	32x32x256	3x3	1
8	Max Pooling	256	16x16x256	2x2	2
9	Fully Connected (Relu)	-	1x128	-	-
10	-Fully Connected	-	-	-	-

Input yang digunakan berukuran 256x256 dengan empat layer konvolusi dan empat layer pooling, dimana layer C1 menggunakan filter = 32, kernel = 2x2 dan *stride* = 1 menghasilkan *feature map* berukuran 256x256x32, kemudian layer kedua menggunakan a*verage pooling* dengan ukuran filter = 32, kernel = 2x2, dan *stride* = 2 menghasilkan *feature map* berukuran 128x128x32, pada layer ketiga yaitu C2 menggunakan filter = 64 dan kernel = 3x3 dan *stride* = 1 menghasilkan menghasilkan *feature map* berukuran 128x128x64, kemudian layer keempat menggunakan a*verage pooling* dengan ukuran filter = 64, kernel = 2x2, dan *stride* = 2 menghasilkan *feature map* berukuran 64x64x64, pada layer kelima yaitu C3 menggunakan filter = 128 , kernel = 1x1, dan *stride* = 1 menghasilkan *feature map* berukuran 64x64x128, kemudian layer keenam menggunakan m*ax pooling* dengan ukuran filter = 128, kernel = 2x2, dan *stride* = 2 menghasilkan *feature map* berukuran 32x32x128, pada layer ketujuh yaitu C4 menggunakan filter = 256, kernel = 1x1, dan *stride* = 1 menghasilkan *feature map* berukuran 32x32x256, kemudian layer kedelapan menggunakan m*ax pooling* dengan ukuran filter = 256, kernel = 2x2, dan *stride* = 2 menghasilkan *feature map* berukuran 16x16x256,

dan terakhir untuk output layer menggunakan fungsi aktivasi *softmax.* Alur proses arsitektur modifikasi dapat dilihat pada gambar 3.17.



Gambar 3.17 *Flowchart* Arsitektur Modifikasi

. Kemudian alur proses dari model arsitektur modifikasi dapat dilihat pada Gambar 3.18.

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3, activation = 'relu', input_shape = [256,256,3]))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3, activation = 'relu'))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=128,kernel_size=3, activation = 'relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=256,kernel_size=3, activation = 'relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(units=120, activation='relu'))
model.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
model.summary()
```

Gambar 3.18 Kode Program Arsitektur Modifikasi

Pada Gambar 3.18 merupakan algoritma yang digunakan dalam membangun arsitektur modifikasi, *function model.add()* digunakan untuk menginisialisasi layer konvolusi, *pooling layer,* fungsi aktivasi, dan *fully connected layer* .Sehingga dari algoritma dan alur proses dari arsiektur original dan modifikasi didapatkan rangkuman perbandingan antara kedua arsitektur pada Tabel 3.3 sebagai berikut.

Tabel 3.3 Perbandingan Arsitektur LeNet-5 Original dan Modifikasi

Layer	**Original (Yann LeCun)**					**Modifikasi**				
	Karakteristik	**Filter**	**Feature Map**	**Kernel Size**	**Stride**	**Karakteristik**	**Filter**	**Feature Map**	**Kernel Size**	**Stride**
Input	Dimensi citra	1	32x32	-	-	Dimensi citra	1	256x256	-	-
1	-*Convolution layer* (C1) -Tanh	6	32x32x6	5x5	1	-*Convolution layer* (C1) -ReLU	32	256x256x32	3x3	1
2	Average Pooling	6	16x16x6	2x2	2	Average Pooling	32	128x128x32	2x2	2
3	-Convolution layer (C2) -Tanh	16	16x16x16	5x5	1	-Convolution layer (C2) -ReLU	64	128x128x64	3x3	1
4	Average Pooling	16	5x5x16	2x2	2	Average Pooling	64	64x64x64	2x2	2
5	-Convolution Layer (C3) -Tanh	120	8x8x120	1x1	-	-Convolution Layer (C3) -ReLU	128	64x64x128	3x3	1
6	Fully Connected	-	-	84	-	Max Pooling	128	32x32x128	2x2	2

7	-Fully Connected -Softmax (Output)	-	-	10	-	-Convolution Layer (C4) -ReLU	256	32x32x256	3x3	1
8	-	-	-	-	-	Max Pooling	256	16x16x256	2x2	2
9					-	Fully Connected (Relu)	-	1x128	-	-
10	-	-	-	-	-	-Fully Connected -Sigmoid (Output)	-	-	-	-

a) Input citra yang digunakan berukuran 256x256, ukuran ini akan menghasilkan resolusi citra lebih besar sehingga semakin banyak pixel pada citra maka semakin banyak *feature map* yang dihasilkan (Herlambang, 2019).

b) Input citra yang digunakan berukuran 256x256, ukuran ini akan menghasilkan resolusi citra lebih besar sehingga semakin banyak pixel pada citra maka semakin banyak *feature map* yang dihasilkan (Herlambang, 2019).

c) Pada proses konvolusi pertama (C1) dimana lapisan ini memiliki kernel berukuran 3x3 dengan *stride* = 1 dan filter berjumlah 32. Adapun menggunakan kernel ganjil (3x3) yaitu karena merupakan matriks simetris jika menggunakan kernel berukuran genap akan tidak efektif, komputasi tinggi, dan ada bagian *feature map* yang tidak terhitung (Farokhah, 2022). Kemudian melalui fungsi aktivasi *ReLu* dimana fungsi ini untuk menghilangkan nilai negatif pada setiap pixelnya (semua nilai negatif akan dikonversi menjadi nol) (Sanjaya & Ayub, 2020). *Feature map* yang dihasilkan berukuran 256x256x32.

d) Selanjutnya proses *pooling layer* pertama menggunakan *average pooling* dimana pada proses ini mengurangi ukuran *feature map*. Pada penelitian ini menggunakan *pool_size* = 2x2 dan *stride* = 2. Sehingga menghasilkan *feature map* berukuran 128x128x32.

e) Proses konvolusi kedua (C2) dimana lapisan ini memiliki kernel berukuran 3x3 dengan *stride* = 1 dan filter bertambah menjadi 64 buah. Penambahan jumlah filter dilakukan agar variasi informasi dari hasil *feature map* sebelumnya yang diperoleh menjadi semakin banyak karena pada proses *pooling layer* terjadi penurunan *feature map* yang menyebabkan banyak informasi yang terbuang. Kemudian melalui fungsi aktivasi *ReLu* dimana menghasilkan f*eature map* yang berukuran 128x128x64.

f) Selanjutnya proses *pooling layer* kedua menggunakan *average pooling* dimana pada proses ini sama dengan *pooling layer* pertama menggunakan *pool_size* = 2x2 dan *stride* = 2. Sehingga menghasilkan *feature map* berukuran 128x128x32.

g) Proses konvolusi ketiga (C3) dimana lapisan ini memiliki kernel berukuran 3x3 dengan *stride* = 1 dan filter bertambah menjadi 128 buah. Kemudian melalui fungsi aktivasi *ReLu* dimana menghasilkan f*eature map* yang berukuran 64x64x128.

h) Selanjutnya menambahkan 1 layer konvolusi dan 2 *pooling layer* baru hal ini bertujuan untuk meningkatkan nilai akurasi dari model arsitektur yang akan dilatih (Herlambang, 2019).

i) Proses *pooling layer* ketiga menggunakan *max pooling* dimana pada proses ini menggunakan *pool_size* = 2x2 dan *stride* = 2. Adapun menggunakan *max pooling* untuk mencari nilai maksimum dari tiap matriks yang telah dilewati oleh setiap kernel, hal ini akan mempengaruhi efisiensi saat proses komputasi. Sehingga menghasilkan *feature map* berukuran 32x32x128

j) Proses konvolusi keempat (C4) dimana lapisan ini memiliki kernel berukuran 3x3 dengan stride = 1 dan filter bertambah menjadi 256 buah. Kemudian melalui fungsi aktivasi *ReLu* dimana menghasilkan f*eature map* yang berukuran 32x32x256.

k) Proses *pooling layer* keempat menggunakan *max pooling* dimana pada proses ini menggunakan *pool_size* = 2x2 dan *stride* = 2. Menghasilkan *feature map* berukuran 16x16x256.

l) Proses selanjutnya yaitu *fully connected layer* dimana *feature map* berukuran 16x16x256 dirubah kedalam bentuk matriks vektor tunggal yang berukuran 1x65536 yang akan digunakan sebagai input *fully connected layer* untuk *dense layer.* Adapun parameter untuk *dense layer* sebesar 128 units yang artinya sebanyak 128 neuron. Selanjutnya dimasukkan melaui fungsi aktivasi *ReLu*.

m) Proses terakhir hasil dari proses sebelumnya akan dimasukkan ke dalam fungsi aktivasi *sigmoid*. Fungsi aktivasi ini akan memberikan output 0 atau 1 dimana nilai 0 akan mewakili class otak normal dan nilai 1 akan mewakili class tumor otak berpenyakit, ini kemudian akan menjadi output dari

klasifikasi yang dilakukan. Berikut rumus yang digunakan pada persamaan 3.3.

$$o_j \quad = \sigma(\sum_{k=1}^{k} \quad x_k \; w_k \; + \beta_k) \tag{3.3}$$

Keterangan:

o_j = *Neuron* pada *hidden layer*

σ = Sigma atau melambangkan fungsi aktivasi ReLu

k = Banyaknya input unit

x = *Neuron* pada input layer

w = Bobot pada input *layer*

β_k = *Noise* atau bias pada input layer

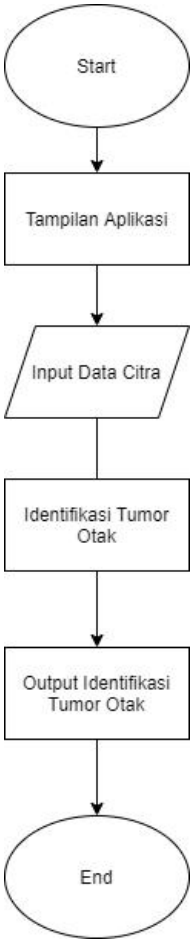
Kemudian rumus untuk fungsi aktifasi *sigmoid* pada Persamaan 3.4 sebagai berikut.

$$\sigma(y_i) = \frac{e^{zj}}{\sum_{k=1}^{k} e^{zk}} \tag{3.4}$$

Dimana j =1,2,3,….. K

3.6 Rancangan Sistem Aplikasi

Rancangan sistem aplikasi dilakukan untuk mempermudah implementasi dalam penerapan aplikasi untuk mengidentifikasi tumor otak menggunakan model arsitektur modifikasi. Rancangan desain sistem mencakup alur antarmuka sistem sampai hasil identifikasi citra tumor otak, adapun alur tersebut pada gambar 3.19.



Gambar 3.19 *Flowchart* Rancangan Sistem Aplikasi

Pada Gambar 3.19 menjelaskan dari tiap komponen-komponen alur sistem dimana pertama yaitu tampilan aplikasi, kemudian proses input gambar, setelah input input diproses untuk mengidentifikasi citra tumor otak sesuai dengan metode CNN model arsitektur modifikasi, dan terakhir output hasil identifikasi citra.

3.7 Proses Training

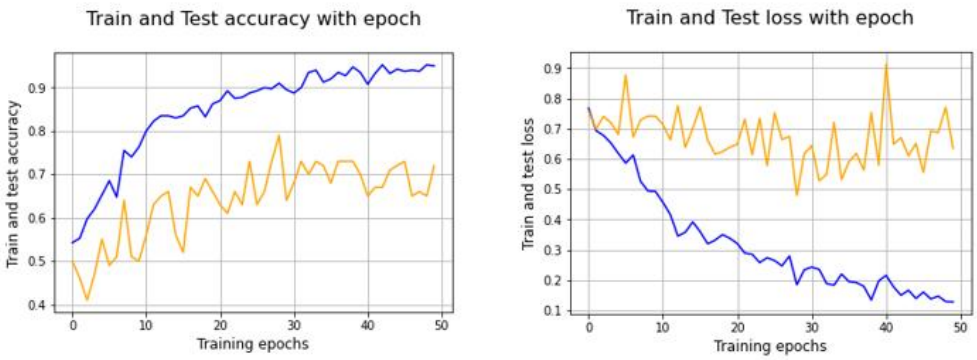
Tahapan proses training merupakan tahapan untuk melatih data pada algoritma LeNet-5. Untuk memperoleh tingkat akurasi yang tinggi dibutuhkan data latih sehingga dapat memperoleh bobot atau nilai yang sesuai selama proses

pelatihan. Jumlah data latih yang digunakan adalah 80% dari data keseluruhan. Dalam proses *training* menggunakan model arsitektur modifikasi yang telah dirancang sebelumnya dimana hasilnya akan dibandingkan apakah terdapat perubahan nilai akurasi dari model original dan setelah dimodifikasi. Adapun algoritma yang digunakan pada gambar 3.20.

```
model.compile( optimizer = 'Adam',loss = 'binary_crossentropy', metrics = ['accuracy'])
mod_chk = ModelCheckpoint(filepath='/content/gdrive/MyDrive/Atumor/Augmen/my_model.hdf5', monitor = 'val_accuracy', save_best_only=True)
hist=model.fit( x= training_set,callbacks=[mod_chk], validation_data = test_set, epochs=50)
```

Gambar 3.20 Kode Program *Training*

Parameter yang diberikan pada proses training meliputi iterasi dengan nilai 50 epoch sehingga proses t*raining* akan diulang sebanyak 50 kali, selanjutnya parameter *batch_size* diberikan nilai sebesar 32 yang berarti jumlah sampel data yang dikerjakan sebanyak kelipatan 32. Digunakan sebanyak 32 untuk menghemat waktu karena data yang digunakan cukup banyak. Sebagai contoh, jika data berjumlah 400 maka tiap iterasi atau *epoch* pertama diambil data mulai dari 0 sampai 31, epoch kedua mulai 32 sampai 63 seterusnya sampai *epoch* terakhir. Sehingga total batch keseluruhan dari banyaknya data tersebut dalam satu epoch yaitu 13 batch. Kemudian menggunakan algoritma *optimize* yang bernama Adam, karena algoritma ini sering dipakai dalam berbagai program karena menghasilkan nilai yang baik dan cepat dalam proses komputasi. Fungsi *ModelCheckpoint()* digunakan untuk menyimpan model hasil *training* berdasarkan dari nilai *val_accuracy* terbaik yang dimana digunakan untuk melakukan *single prediction* pada data citra. Hasil dari *training* tersebut pada gambar 3.21.



Gambar 3.21 Grafik Hasil *Training* Arsitektur Modifikasi

Pada proses data *training* menggunakan data asli arsitektur modifikasi diperoleh hasil *loss* sebesar 0,7669 dan akurasi sebesar 54,25% pada *epoch* ke-1, kemudian pada *epoch* ke-50 nilai *loss* menurun dengan nilai sebesar 0,1282 dan akurasi meningkat sebesar 95,00%. Kemudian untuk data *testing* pada *epoch* ke-1 *val_loss* mendapatkan nilai 0,7568 dan *val_accuracy* mendapatkan nilai 50,00%, pada *epoch* ke-50 *val_loss* mendapatkan nilai 0,6348 dan *val_accuracy* mendapatkan nilai 72,00%.

BAB IV

UJI COBA DAN PEMBAHASAN

Pada tahap ini akan menjelaskan tentang serangkaian pengujian dalam medeteksi tumor menggunakan citra MRI dengan algoritma CNN model arsitektur LeNet-5 original dan modifikasi.

4.1. Skenario Uji Coba

Pengujian dilakukan untuk mengetahui hasil akurasi dari arsitektur modifikasi. Langkah pertama dalam pengujian yaitu pengumpulan data. Data yang digunakan merupakan data citra MRI tumor otak dan otak normal sejumlah 500 citra dimana terbagi atas dua kelas yaitu otak normal dan otak tumor. Selanjutnya menentukan *ground truth* atau kebenaran mutlak yang akan digunakan dalam perbandingan hasil prediksi dari sistem.

Kemudian dilakukan pengukuran hasil akurasi model arsitektur agar menghasilkan tingkat akurasi yang maksimal. Dalam menguji suatu model, dilakukan sebuah metode dalam memprediksi keakuratan dan kinerja suatu model, salah satu Metode yang digunakan dalam penelitian ini adalah *K-Fold Cross Validation*. Metode ini digunakan digunakan dengan cara Membagi *dataset* menjadi data latih dan data uji. *Dataset* tersebut akan dibagi sebanyak *k*, dimana *k* merupakan banyaknya data uji yang dibagi dalam proses iterasi. Sebagai contoh jika *k*=5 maka *Dataset* akan terbagi menjadi 5 bagian atau *subset* kemudian sisanya akan menjadi data latih setiap *k*. Jika *Dataset* berjumlah 500 maka data uji berjumlah 100 dan 400 data latih yang kemudian dilakukan sebanyak 5 kali

sesuai dengan bagian data latih yang berbeda-beda. Adapun tabel pembagian data dari *K-Fold Cross Validation* ditunjukkan pada gambar 4.1 sebagai berikut:

Iterasi	Jumlah Data				
1	Uji	Latih	Latih	Latih	Latih
2	Latih	Uji	Latih	Latih	Latih
3	Latih	Latih	Uji	Latih	Latih
4	Latih	Latih	Latih	Uji	Latih
5	Latih	Latih	Latih	Latih	Uji

Gambar 4.1 *K-Fold Cross Validation*

Adapun proses skenario uji coba sebagai berikut:

a. Menyiapkan *dataset* citra dimana terbagi atas dua kelas yaitu 80% untuk data latih dan 20% untuk data uji.

b. Proses t*raining* merupakan proses untuk memberikan kecerdasan pada algoritma CNN. Data yang sudah disiapkan akan dilatih menggunakan model CNN LeNet-5 original dan modifikasi. Kemudian setelah dilatih data jaringan disimpan untuk digunakan dalam mengukur besaran akurasi model CNN LeNet-5 original dan modifikasi.

c. Proses *testing* digunakan dalam proses menguji data citra dalam mengidentifikasi tumor otak.

d. Membuka aplikasi melalui *website* yang telah dibuat. Proses pengujian menggunakan aplikasi google colab, yang merupakan *tools* yang terdapat *Jupyter Notebook* dengan bahasa python berbasis *cloud* dimana

e. Semua pekerjaan dilakukan secara *online*. Berikut untuk tampilan dari aplikasi yang dibuat pada gambar 4.2

Gambar 4.2 Tampilan Aplikasi

Pada Gambar 4.2 pengguna dapat melakukan unggah data citra untuk mengetahui hasil dari prediksi citra tumor yang di unggah.

f. Selanjutnya pengujian sistem pada tiap model menggunakan *confusion matrix.* Data yang digunakan yaitu data citra uji berjumlah sekitar 100 citra. Parameter perhitungannya yaitu akurasi, sensitivitas, dan spesifisitas. Untuk menghitung parameter tersebut confusion matrix memiliki 4 klasifikasi, yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP) dan *false negative* (FN). Adapun rumus yang digunakan sebagai berikut:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4.1)$$

$$\text{Sensitivitas} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.2)$$

$$\text{Spesifisitas} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (4.3)$$

Dimana:

1. True Positive (TP) : Citra dengan penyakit tumor, *ground truth* diprediksi oleh sistem masuk penyakit tumor
2. True Negative (TN) : Citra bukan penyakit tumor, *ground truth* diprediksi oleh sistem tidak masuk penyakit tumor atau normal
3. False Positive (FP) : Citra bukan penyakit tumor atau normal, *ground truth* diprediksi oleh sistem penyakit tumor
4. False Negative (FN) : Citra dengan penyakit tumor, *ground truth* diprediksi oleh sistem tidak masuk penyakit tumor atau normal

4.2. Hasil Uji Coba

Dari skenario pada sub bab 4.1, dengan menggunakan arsitektur modifikasi dengan label citra otak normal dan otak tumor. Pengujian data sesuai pada lampiran 2 menggunakan data uji sebanyak 100 citra, adapun hasil *confusion matrix* sebagai berikut:

1. TP = 45
2. TN = 43
3. FP = 7
4. FN = 5

Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

1. Akurasi $= \frac{45 + 43}{45+ 43+ 7+ 5}$ x 100% = 88,00%
2. Sensitivitas $= \frac{45}{45+ 5}$ x 100% = 90,00%
3. Spesifisitas $= \frac{43}{7+ 43}$ x 100% = 86,00%

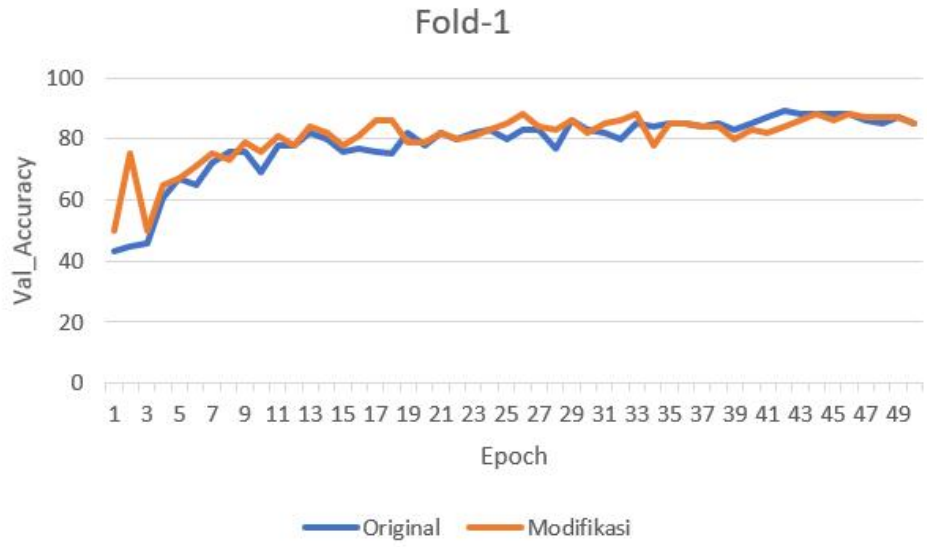
Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 88,00%, sensitivitas sebesar 90,00%, spesifitas sebesar 86,00%

4.2.1 Hasil Pengujian Dengan *K-Fold Cross Validation*

Pada tahap ini dilakukan pengujian pada kedua model arsitektur dengan *K-fold cross validation.* Sehingga didapatkan hasil *training* masing-masing *fold* dengan epoch 50.

1. Hasil Uji *Training* Fold-1

Pada Fold-1, hasil *training* kedua model ditunjukkan pada gambar 4.3.



Gambar 4.3 Grafik Hasil *Training Validation Accuracy* Fold-1

Pada proses *training* menunjukkan untuk model original diperoleh nilai *val_accuracy* sebesar 43,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 85,00%. Kemudian untuk model modifikasi diperoleh nilai *val_accuracy* sebesar 50,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar

85,00%. Berdasarkan hasil pengujian model modifikasi pada lampiran 3 yang dilakukan, didapat nilai *confussion matrix* sebagai berikut:

1. TP = 49
2. TN = 42
3. FP = 0
4. FN = 9

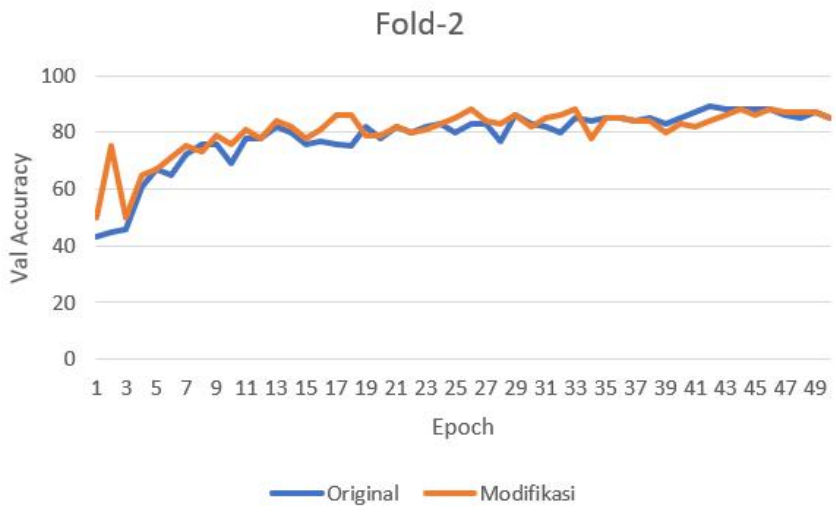
Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

1. Akurasi $= \frac{49+42}{49+42+0+9}$ x 100% = 91,00%
2. Sensitivitas $= \frac{49}{49+9}$ x 100% =84,48%
3. Spesifisitas $= \frac{42}{0+42}$ x 100% = 100%

Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 91,00%, sensitivitas sebesar 84,48%, spesifitas sebesar 100%.

2. Hasil Uji *Training* Fold-2

Pada Fold-2, hasil *training* kedua model ditunjukkan pada Gambar 4.4.



Gambar 4.4 Grafik Hasil *Training Validation Accuracy* Fold-2

Pada proses *training* menunjukkan untuk model original diperoleh nilai *val_accuracy* sebesar 50,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 82,00%. Kemudian untuk model modifikasi diperoleh nilai *val_accuracy* sebesar 54,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 87,00%. Berdasarkan hasil pengujian model modifikasi pada lampiran 4 yang dilakukan, didapat nilai *confussion matrix* sebagai berikut:

1. TP = 50
2. TN = 45
3. FP = 1
4. FN = 4

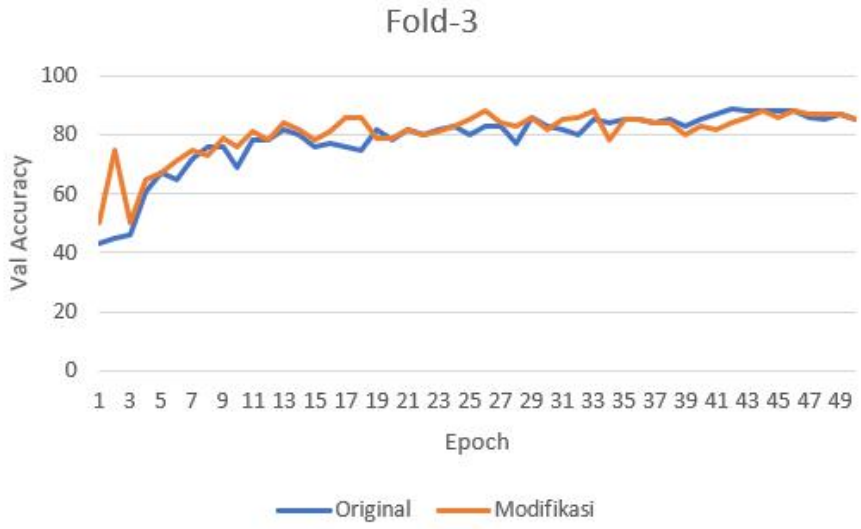
Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

1. Akurasi $= \frac{50+45}{50+45+1+4}$ x 100% = 95,00%
2. Sensitivitas $= \frac{50}{50+4}$ x 100% =92,59%
3. Spesifisitas $= \frac{45}{1+45}$ x 100% = 97,82%

Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 95,00%, sensitivitas sebesar 92,59%, spesifitas sebesar 97,82%.

3. Hasil Uji *Training* Fold-3

Pada Fold-3, hasil *training* kedua model ditunjukkan pada Gambar 4.5.



Gambar 4.5 Grafik Hasil *Training Validation Accuracy* Fold-3

Pada proses *training* menunjukkan untuk model original diperoleh nilai *val_accuracy* sebesar 50,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 82,00%. Kemudian untuk model modifikasi diperoleh nilai *val_accuracy* sebesar 54,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 87,00%. Berdasarkan hasil pengujian model modifikasi pada lampiran 5 yang dilakukan, didapat nilai *confussion matrix* sebagai berikut:

1. TP = 47
2. TN = 44
3. FP = 4
4. FN = 5

Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

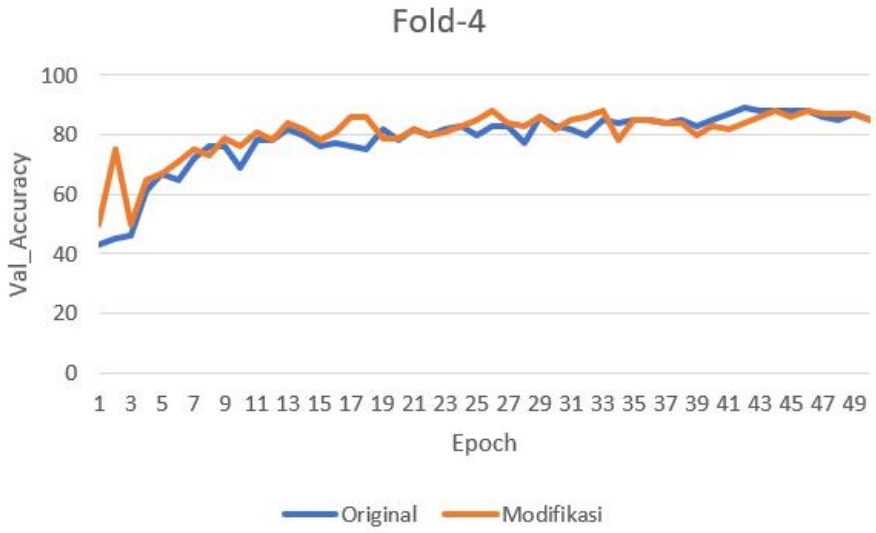
1. Akurasi $= \frac{47+44}{47+44+4+5}$ x 100% = 91,00%
2. Sensitivitas $= \frac{47}{47+5}$ x 100% =90,38%

3. Spesifisitas $= \frac{44}{4+44}$ x 100% = 91,66%

Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 91,00%, sensitivitas sebesar 90,38%, spesifitas sebesar 91,66%.

4. Hasil Uji *Training* Fold-4

Pada Fold-4, hasil *training* kedua model ditunjukkan pada Gambar 4.6.



Gambar 4.6 Grafik Hasil *Training Validation Accuracy* Fold-4

Pada proses *training* menunjukkan untuk model original diperoleh nilai *val_accuracy* sebesar 58,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 84,00%. Kemudian untuk model modifikasi diperoleh nilai *val_accuracy* sebesar 57,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 85,00%. Berdasarkan hasil pengujian model modifikasi pada lampiran 6 yang dilakukan, didapat nilai *confussion matrix* sebagai berikut:

1. TP = 51
2. TN = 43
3. FP = 2

4. FN = 4

Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

1. Akurasi $= \frac{51+45}{51+45+0+4}$ x 100% = 96,00%

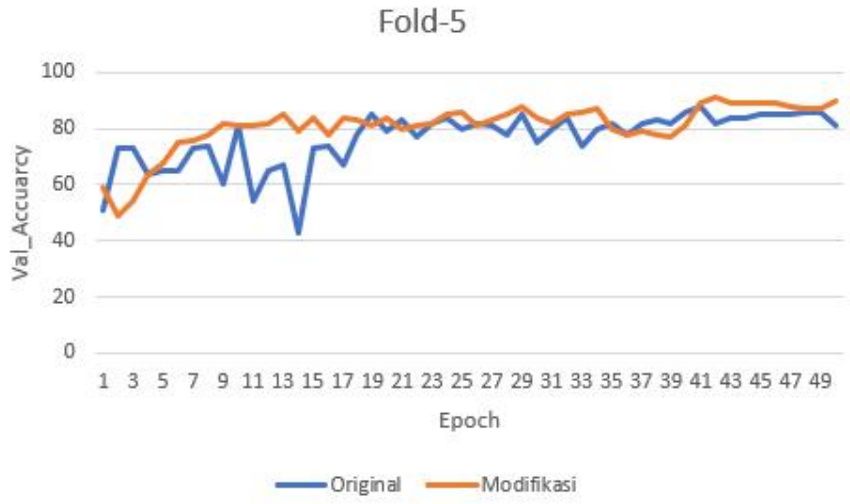
2. Sensitivitas $= \frac{51}{51+4}$ x 100% =92,72%

3. Spesifisitas $= \frac{43}{2+43}$ x 100% =95,55%

Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 96,00%, sensitivitas sebesar 92,72%, spesifitas sebesar 95,55%.

5. Hasil Uji *Training* Fold-5

Pada Fold-5, hasil *training* kedua model ditunjukkan pada Gambar 4.7.



Gambar 4.7 Grafik Hasil *Training Validation Accuracy* Fold-5

Pada proses *training* menunjukkan untuk model original diperoleh nilai *val_accuracy* sebesar 51,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar 84,00%. Kemudian untuk model modifikasi diperoleh nilai *val_accuracy* sebesar 59,00% pada *epoch* ke-1, pada *epoch* ke-50 nilai meningkat sebesar

90,00%. Berdasarkan hasil pengujian model modifikasi pada lampiran 7 yang dilakukan, didapat nilai *confussion matrix* sebagai berikut:

1. TP = 50
2. TN = 44
3. FP = 1
4. FN = 5

Kemudian perhitungan untuk menghitung performa sistem sebagai berikut:

1. Akurasi $= \frac{50+ 44}{50+ 44+ 1+ 5}$ x 100% = 94,00%
2. Sensitivitas $= \frac{50}{50+ 5}$ x 100% =90,90%
3. Spesifisitas $= \frac{44}{1+ 44}$ x 100% =97,77%

Berdasarkan hasil perhitungan untuk menghitung performa sistem, diperoleh hasil akurasi sebesar 94,00%, sensitivitas sebesar 90,90%, spesifitas sebesar 97,97%.

Setelah dilakukan pengujian menggunakan *K-Fold Cross validation,* selanjutnya dibuatkan tabel confusion matrix dari hasil uji tiap kategori dari data-data tersebut. Adapun nilai dari hasil tiap *K-Fold Cross Validation* model arsitektur modifikasi terdapat pada Tabel 4.1.

Tabel 4.1 *K-Fold Cross Validation* Model Arsitektur Modifikasi

Iterasi	**Akurasi**	**Sensitivitas**	**Spesifitas**
Fold-1	96,00%	94,33%	97,87%
Fold-2	95,00%	92,59%	97,82%
Fold-3	91,00%	90,38%	91,66%
Fold-4	96,00%	92,72%	95,55%

Fold-5	94,00%	90,90%	97,97%

Dapat diketahui bahwa nilai akurasi terendah terletak pada *fold*-3 dengan nilai 91,00% dan tertinggi pada *fold*-1 dan *fold-4* dengan nilai 96,00%. Untuk nilai sensitivitas terendah terletak pada *fold*-3 dengan nilai 90,38% dan tertinggi terletak pada *fold*-1 dengan nilai 94,33%. Untuk nilai spesifitas terendah terletak pada *fold*-3 dengan nilai 91,66% dan tertinggi terletak pada *fold*-2 dengan nilai 97,82%.

4.3 Pembahasan

Dalam sub bab ini menjelaskan mengenai hasil identifikasi citra tumor otak menggunakan arsitektur original dan modifikasi. Setelah dilakukan uji coba hasil berdasarkan lampiran 2 didapatkan hasil performa sistem untuk akurasi sebesar 88,00%, sensitivitas sebesar 90,00%, spesifitas sebesar 86,00%. Kemudian dilakukan pengujian dengan *K-Fold Cross Validation*, dimana hasil dari tiap *fold* dihitung nilai rata-rata dari tiap akurasi, sensitivitas, dan spesivitas. Adapun tabel dari nilai rata-rata *K-Fold Cross Validation* pada Tabel 4.2.

Tabel 4.2 Rata-Rata Performa Sistem *K-Fold Cross Validation* Model Modifikasi

Performa Sistem	**Nilai**
Akurasi	94,40%
Sensitivitas	92,18%
Spesivitas	96,27%

Dari Tabel 4.2 diketahui bahwa nilai akurasi model aristektur modifikasi menghasilkan nilai yang tinggi. Hal ini dikarenakan bahwa *feature map* yang diterapkan berbeda dimana pada arsitektur modifikasi menerapkan input feature map berukuran 256x256 sedangkan untuk arsitektur original berukuran 32x32 sehingga pada proses pelatihan model feature map yang dihasilkan akan lebih banyak pada resolusi yang lebih besar dan pada saat layer *fully connected* data neuron lebih banyak dan menghasilkan nilai akurasi yang lebih besar.

Kemudian pada saat pelatihan model terjadi nilai error yang cukup tinggi pada pelatihan awal, hal ini dikarenakan model arsitektur yang dilatih menjadi *overfitting* dimana model yang dilakukan terlalu kompleks dimana pada arsitektur original menggunakan 6 layer sedangkan pada arsitektur modifikasi menggunakan 9 layer .

Convolutional Neural Network merupakan bentuk ilmu pengetahuan. Allah SWT mewajibkan umatnya dalam menuntut ilmu, Allah SWT meninggikan derajat orang-orang yang mencari ilmu karena ridha-Nya. Hal ini tertuang dalam firman Allah Surah Al- Mujadalah ayat 11 yang berbunyi,

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْٓا اِذَا قِيْلَ لَكُمْ تَفَسَّحُوْا فِى الْمَجٰلِسِ فَافْسَحُوْا يَفْسَحِ اللّٰهُ لَكُمْۚ وَاِذَا قِيْلَ انْشُزُوْا فَانْشُزُوْا يَرْفَعِ اللّٰهُ الَّذِيْنَ اٰمَنُوْا مِنْكُمْۙ وَالَّذِيْنَ اُوْتُوا الْعِلْمَ دَرَجٰتٍۗ وَاللّٰهُ بِمَا تَعْمَلُوْنَ خَبِيْرٌ

"Wahai orang-orang yang beriman! Apabila dikatakan kepadamu, "Berilah kelapangan di dalam majelis-majelis, maka lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan, "Berdirilah kamu," maka berdirilah, niscaya Allah akan mengangkat (derajat) orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Dan Allah Mahateliti apa yang kamu kerjakan" (Q.S. Al- Mujadalah :11).

Rasulullah SAW juga bersabda pada sebuah hadits oleh HR Tirmidzi yang berbunyi,

وَمَنْ سَلَكَ طَرِيقًا يَلْتَمِسُ فِيهِ عِلْمًا سَهَّلَ اللَّهُ لَهُ بِهِ طَرِيقًا إِلَى الْجَنَّةِ

"Siapa yang menempuh jalan untuk mencari ilmu, maka Allah akan mudahkan baginya jalan menuju surga" (HR Muslim, no. 2699).

Maka dari itu, menuntut ilmu bagi kaum muslimin harus disertai dengan keikhlasan tidak hanya semata-mata hanya untuk mencerdaskan diri sendiri namun juga mengharapkan ridha dari Allah SWT.

Di dalam penelitian ini juga membahas tumor otak, tumor otak merupakan penyakit yang ditandai dengan adanya pertumbuhan jaringan abnormal di otak. Penyakit ini bisa menyebabkan kematian jika tidak segera ditangani dengan baik. Maka dari itu manusia dianjurkan untuk menjaga kesehatan. Dalam Islam kita diwajibkan untuk menjaga kesehatan serta kebersihan setiap saat. Rasulullah SAW bersabda,

عمتان مغبون فيهما كثير من الناس الصحة والفراغ

“Dua kenikmatan yang sering dilupakan oleh kebanyakan manusia adalah kesehatan dan waktu luang” (HR. Al-Bukhari: 6412, at-Tirmidzi: 2304, Ibnu Majah: 4170).

Tafsir Ibnul Qayyim Al-Jauziyyah dalam kitabnya yang berjudul Zad al-Ma’ad, menjelaskan bahwa penyakit ada dua macam yaitu penyakit jasmani dan rohani. Alquran adalah penyembuh yang sempurna dari seluruh penyakit jasmani dan rohani, demikian pula penyakit dunia dan akhirat. Kaidah pengobatan jasmani ada tiga macam yaitu menjaga kesehatan, tindakan preventif, dan menghindari

dari hal-hal yang merusak dan berbahaya. Kemudian pengobatan hati atau rohani diperoleh dari menyebut asma-asma rasul dan pencipta-Nya, mengetahui sifat-sifat-Nya, hukum-hukum dan perbuatan-Nya, mementingka keridhaan-Nya dan menjauhi larangan-Nya.

Kemudian pada Surat Asy-Syu'ara Ayat 80 yang berbunyi:

وَإِذَا مَرِضْتُ فَهُوَ يَشْفِينِ

"Dan apabila aku sakit, Dialah Yang menyembuhkan aku" (Q.S. AsyAsy-Syu'ara:80).

Ayat ini menjelaskan bahwa ketika manusia sakit, Allah SWT yang menyembuhkannya. Allah Maha Kuasa untuk menyembuhkan segala penyakit yang diderita manusia.

Imam Jamaluddin al-Qasimi pada tafsirnya yang berjudul Mahasin al-Ta'wil dalam tafsirnya menjelaskan bahwa ayat ini menggambarkan akhlak seorang hamba Allah terhadap pencipta-Nya. Karena terkadang penyakit merupakan akibat dari perbuatan manusia, seperti pelanggaran standar kesehatan atau kebiasaan hidup sehari-hari, serangan penyakit pada tubuh tidak dapat dihindari. Di sisi lain, hanya Allah yang memiliki kekuatan untuk menyembuhkan. Jika orang sakit merasa seperti ini ketika sakit, maka dia akan benar-benar merasakan nikmat Allah setelah sembuh dari penyakitnya. Kenyataannya kebanyakan orang jatuh sakit karena tidak memperhatikan standar kesehatan saat ini.

Kemudian Ibnu Katsiralam pada tafsirnya berjudul Tafsir Ibnu Katsir jilid 6 halaman 158 menjelaskan ayat ini bahwa rasa sakit itu ditimpakan kepada Ibrahim, meskipun sebenarnya itu berasal dari takdir Allah dan perintah-Nya, juga

ciptaan-Nya, tetapi sengaja dikaitkan dengan Ibrahim sebagai cara ketakwaan terhadap Allah Swt. Nikmat dan hidayah diserahkan kepada Allah, sedangkan amarah dijauhkan dari penyebabnya demi kesopanan, dan kesalahan diserahkan kepada hamba-hamba-Nya,

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat disimpulkan sistem yang dibuat memperoleh hasil akurasi sebesar 88,00%, sensitivitas sebesar 90,00%, spesifitas sebesar 86,00%. Kemudian dilakukan metode uji coba menggunakan metode *K-Fold Cross Validation* dengan total rata-rata akurasi 94,40%, sensitivitas rata-rata 92,18%, dan spesifitas rata-rata 96,27%. Dari nilai hasil akurasi tersebut pengujian model arsitektur modifikasi termasuk dalam kategori yang baik.

5.2. Saran

Dalam penelitian ini, saran untuk penulis untuk penelitian selanjutnya adalah sebagai berikut:

1. Menambah data menggunakan data primer agar mendapatkan hasil yang lebih akurat
2. Memperbanyak jumlah data yang digunakan, hal ini akan mempengaruhi kinerja sistem agar lebih baik

DAFTAR PUSTAKA

Alwafi Ridho Subarkah (2018). "IMPLEMENTASI deep learning UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA citra WAYANG GOLEK" vol. 151, no. 2, pp. 10–17, 2018.

R. A. Aman et al. (2016), "Panduan Penatalaksanaan Tumor Otak," Kom. Penanggulangan Kanker Nas., pp. 1–79, 2016.

D. Hulmansyah,. (2020). "Prosedur Pemeriksaan Magnetic Resonance Spectroscopy (Mrs) Kepala Pada Kasus Tumor Otak Di Instalasi Radiologi Rs Awal Bros Pekanbaru," Ojs.Stikesawalbrospekanbaru.Ac.Id, pp. 41–47, 2020.

P. A. Nugroho, I. Fenriana, and R. Arijanto,.(2020). "Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia," Algor, vol. 2, pp. 12–21.

.

D. G. Brown, (2007). "Instrumentation for parallel magnetic resonance imaging," no. December, 2007.

B. B. Traore, B. Kamsu-Foguem, and F. Tangara, (2018). "Deep convolution neural network for image recognition," Ecol. Inform., vol. 48, pp. 257–268.

T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang. (2015). "Implementation of Training Convolutional Neural Networks," 2015, [Online]. Available: http://arxiv.org/abs/1506.01195.

A. Santoso and G. Ariyanto. (2018). "Implementasi deep learning Berbasis Keras Untuk Pengenalan Wajah," Emit. J. Tek. Elektro, vol. 18, no. 01, pp. 15–21.

H. Dong, G. Yang, F. Liu, Y. Mo, Y. Guo, and N. Heart. (2011). "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks," pp. 1–12.

M. N. Wu, C. C. Lin, and C. C. Chang. (2007). "Brain tumor detection using color-based K-means clustering segmentation," Proc. - 3rd Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IIHMSP 2007., vol. 2, pp. 245–248..

D. Daimary, M. B. Bora, K. Amitab, and D. Kandar. (2020). "Brain Tumor Segmentation from MRI Images using Hybrid Convolutional Neural Networks," Procedia Comput. Sci., vol. 167, no. 2019, pp. 2419–2428.

I. B. Santoso, Y. Adrianto, A. D. Sensusiati, D. P. Wulandari, and I. K. E. Purnama. (2021). “Epileptic EEG Signal Classification Using Convolutional Neural Network Based on Multi-Segment of EEG Signal,” Int. J. Intell. Eng. Syst., vol. 14, no. 3, pp. 160–176.

X. Gu, Z. Shen, J. Xue, Y. Fan, and T. Ni. (2021). “Brain Tumor MR Image Classification Using Convolutional Dictionary Learning With Local Constraint,” Front. Neurosci., vol. 15, no. May, pp. 1–12.

H. M. Rai and K. Chatterjee. (2020). “Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images,” Mach. Learn. with Appl., vol. 2, no. October, p. 100004.

W. S. Eka Putra. (2016). “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” J. Tek. ITS, vol. 5, no. 1.

S. Sharma, S. Sharma, and A. Anidhya, (2020). “Understanding Activation Functions in Neural Networks,” Int. J. Eng. Appl. Sci. Technol., vol. 4, no. 12.

S. Ilahiyah and A. Nilogiri, (2018) “Implementasi deep learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network,” JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones., vol. 3, no. 2.

M. E. Al Rivan and A. G. Riyadi, “Perbandingan Arsitektur LeNet dan lexNet Pada Metode Convolutional Neural Network Untuk Pengenalan American Sign Language,” J. Komput. Terap., vol. 7, no. 1, pp. 53–61, 2021.

L. Wan, Y. Chen, H. Li, and C. Li, “Rolling-element bearing fault diagnosis using improved lenet-5 network,” Sensors (Switzerland), vol. 20, no. 6, pp. 1–23, 2020, doi: 10.3390/s20061693.

B. Soundarya, R. Krishnaraj, and S. Mythili, “Visual Speech Recognition using Convolutional Neural Network,” IOP Conf. Ser. Mater. Sci. Eng., vol. 1084, no. 1, p. 012020, 2021, doi: 10.1088/1757-899x/1084/1/012020.

Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, “LeNet,” Proc. IEEE, no. November, pp. 1–46, 1998.

S. Iqbal, M. U. Ghani, T. Saba, and A. Rehman, “Brain tumor segmentation in multi-spectral MRI using convolutional neural networks (CNN),” Microsc. Res. Tech., vol. 81, no. 4, pp. 419–427, 2018, doi: 10.1002/jemt.22994.

M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, “Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk

Pengenalan Doodle," J. Algoritm., vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.

Q. Zhang, J. X. Hu, and S. Zhou, "The detection of hyperthyroidism by the modified LeNet-5 network," Indian J. Pharm. Sci., vol. 82, pp. 108–114, 2020, doi: 10.36468/pharmaceutical-sciences.spl.108.

D. Fitriati, "Perbandingan Kinerja CNN LeNet 5 dan Extreme Learning Machine pada Pengenalan Citra Tulisan Tangan Angka," J. Teknol. Terpadu, vol. 2, no. 1, pp. 10–16, 2016.

Kadam, R. (2021). Kernels (Filters) in convolutional neural network (CNN), Let's talk about them. Retrieved Maret 7, 2022, from https://medium.com/codex/kernels-filters-in-convolutional-neural-network-cnn-lets-talk-about-them-ee4e94f3319

Pelletier, Charlotte & Webb, Geoffrey & Petitjean, François. (2019). "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series". Remote Sensing. 11. 523. 10.3390/rs11050523.

I. W. G. Putra, (2020). "Bagian III Artificial Neural Network,"

N. Septiana, "Menggunakan Representasi Anti Textons Dan K-Nearest Neighbour," Informasi, Fak. Teknol., pp. 18–19, 2017.

S. Sarraf and G. Tofighi, "Classification of Alzheimer's Disease Structural MRI Data by deep learning Convolutional Neural Networks," no. March 2016, 2016, [Online]. Available: http://arxiv.org/abs/1607.06583.

Mubarok, Hamdani. (2019). "Identifikasi ekspresi wajah berbasis citra menggunakan algoritma Convolutional Neural Network (CNN)".vol. 3, no. 1, pp. 10–12, 2019.

J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," J. Tek. Inform. dan Sist. Inf., vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

Herlambang, MB. (2019). Deep Learning: Convolutional Neural Networks (aplikasi). Retreived Juni 25, 2022, from https://www.megabagus.id/deep-learning-convolutional-neural-networks-aplikasi/.

Farokhah, Lia. (2022). Kernels di Convolutional Neural Network/ CNN.Retreived Juni 25, 2022, from https://www.youtube.com/watch?v=5yi5MXEMjyY

Landis JR, Koch GG. (1977) The measurement of observer agreement for categorical data. Biometrics.(1):159–74

Jauzi, Ibnu Qayyim Al. (2006). “ Zaadul Ma'ad Bekal Menuju Ke Akherat”. Pustaka Azzam, Jakarta.

LAMPIRAN

Lampiran 1 Kode Program:

```
from google.colab import drive
drive.mount('/content/gdrive', force_remount=False)
cd /content/gdrive/MyDrive/Atumor
ls
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
tf.__version__
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   rotation_range = 25,
                                   width_shift_range = 0.2,
                                   height_shift_range = 0.2,
                                   horizontal_flip = True)
training_set = train_datagen.flow_from_directory('dataset/Train',
                                                 target_size = (256,256),
                                                 batch_size = 32,
                                                 class_mode = 'binary')
test_datagen = ImageDataGenerator(rescale = 1./255)
test_set = test_datagen.flow_from_directory('dataset/Test',
                                            target_size = (256,256),
                                            batch_size = 32,
                                            class_mode = 'binary')
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3, activation = 'relu', inp
ut_shape = [256,256,3]))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3, activation = 'relu'))
model.add(tf.keras.layers.AveragePooling2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3, activation = 'relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
model.add(tf.keras.layers.Flatten()) #merubah dataset menjadi 1 dimensi untuk ma
suk ke NN
model.add(tf.keras.layers.Dense(units=120, activation='relu')) #hidden layer
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(units=1, activation='sigmoid')) #semua output hi
dden layer akan terkoneksi menjadi 1 output
model.summary()
from keras.callbacks import ModelCheckpoint
import h5py
```

```
model.compile( optimizer = 'Adam',loss = 'binary_crossentropy', metrics = ['accur
acy']) #Mengkonfigurasi model untuk melatih data yang diberikan
mod_chk = ModelCheckpoint(filepath='/content/gdrive/MyDrive/Atumor/Augme
n/my_model.hdf5', monitor = 'val_accuracy', save_best_only=True)
hist=model.fit( x= training_set,callbacks=[mod_chk], validation_data = test_set, e
pochs=50)
#plotting result
import matplotlib.pyplot as plt
plt.plot(hist.history['loss'], color='blue', label='train')
plt.plot(hist.history['val_loss'], color='orange', label='train')
plt.grid(True)
plt.title("Train and Test loss with epoch\n", fontsize=16)
plt.xlabel("Training epochs", fontsize=12)
plt.ylabel("Train and test loss", fontsize=12)
plt.show()
#plotting result with accuracy
plt.plot(hist.history['accuracy'], color='blue', label='train')
plt.plot(hist.history['val_accuracy'], color='orange', label='train')
plt.grid(True)
plt.title("Train and Test accuracy with epoch\n", fontsize=16)
plt.xlabel("Training epochs", fontsize=12)
plt.ylabel("Train and test accuracy", fontsize=12)
plt.show()
```

Lampiran 2 Hasil Uji Citra Tumor Otak dan otak normal

Tabel Hasil Uji Citra Tumor Otak

No	Gambar	Input	Output	TP	TN	FP	FN
1	tumor (1).jpg	Tumor	Tumor	1	0	0	0
2	tumor (2).jpg	Tumor	Tumor	1	0	0	0
3	tumor (3).jpg	Tumor	Tumor	1	0	0	0
4	tumor (4).jpg	Tumor	Tumor	1	0	0	0
5	tumor (5).jpg	Tumor	Tumor	1	0	0	0
6	tumor (6).jpg	Tumor	Tumor	1	0	0	0
7	tumor (7).jpg	Tumor	Tumor	1	0	0	0
8	tumor (8).jpg	Tumor	Tumor	1	0	0	0
9	tumor (9).jpg	Tumor	Tumor	1	0	0	0
10	tumor (10).jpg	Tumor	Tumor	1	0	0	0
11	tumor (11).jpg	Tumor	Tumor	1	0	0	0
12	tumor (12).jpg	Tumor	Tumor	1	0	0	0
13	tumor (13).jpg	Tumor	Tumor	1	0	0	0
14	tumor (14).jpg	Tumor	Tumor	1	0	0	0
15	tumor (15).jpg	Tumor	Tumor	1	0	0	0
16	tumor (16).jpg	Tumor	Tumor	1	0	0	0
17	tumor (17).jpg	Tumor	Tumor	1	0	0	0
18	tumor (18).jpg	Tumor	Tumor	1	0	0	0
19	tumor (19).jpg	Tumor	Tumor	1	0	0	0
20	tumor (20).jpg	Tumor	Tumor	1	0	0	0
21	tumor (21).jpg	Tumor	Tumor	1	0	0	0
22	tumor (22).jpg	Tumor	Tumor	1	0	0	0
23	tumor (23).jpg	Tumor	Tumor	1	0	0	0
24	tumor (24).jpg	Tumor	Normal	0	0	0	1
25	tumor (25).jpg	Tumor	Tumor	1	0	0	0
26	tumor (26).jpg	Tumor	Tumor	1	0	0	0
27	tumor (27).jpg	Tumor	Tumor	1	0	0	0
28	tumor (28).jpg	Tumor	Tumor	1	0	0	0
29	tumor (29).jpg	Tumor	Tumor	1	0	0	0
30	tumor (30).jpg	Tumor	Tumor	1	0	0	0
31	tumor (31).jpg	Tumor	Tumor	1	0	0	0
32	tumor (32).jpg	Tumor	Normal	0	0	0	1
33	tumor (33).jpg	Tumor	Tumor	1	0	0	0
34	tumor (34).jpg	Tumor	Tumor	1	0	0	0
35	tumor (35).jpg	Tumor	Tumor	1	0	0	0
36	tumor (36).jpg	Tumor	Normal	0	0	0	1
37	tumor (37).jpg	Tumor	Tumor	1	0	0	0

38	tumor (38).jpg	Tumor	Tumor	1	0	0	0
39	tumor (39).jpg	Tumor	Tumor	1	0	0	0
40	tumor (40).jpg	Tumor	Tumor	1	0	0	0
41	tumor (41).jpg	Tumor	Normal	0	0	0	1
42	tumor (42).jpg	Tumor	Tumor	1	0	0	0
43	tumor (43).jpg	Tumor	Tumor	1	0	0	0
44	tumor (44).jpg	Tumor	Tumor	1	0	0	0
45	tumor (45).jpg	Tumor	Normal	0	0	0	1
46	tumor (46).jpg	Tumor	Tumor	1	0	0	0
47	tumor (47).jpg	Tumor	Tumor	1	0	0	0
48	tumor (48).jpg	Tumor	Tumor	1	0	0	0
49	tumor (49).jpg	Tumor	Tumor	1	0	0	0
50	tumor (50).jpg	Tumor	Tumor	1	0	0	0
51	normal (1).jpg	Normal	Normal	0	1	0	0
52	normal (2).jpg	Normal	Normal	0	1	0	0
53	normal (3).jpg	Normal	Normal	0	1	0	0
54	normal (4).jpg	Normal	Normal	0	1	0	0
55	normal (5).jpg	Normal	Normal	0	1	0	0
56	normal (6).jpg	Normal	Normal	0	1	0	0
57	normal (7).jpg	Normal	Normal	0	1	0	0
58	normal (8).jpg	Normal	Normal	0	1	0	0
59	normal (9).jpg	Normal	Normal	0	1	0	0
60	normal (10).jpg	Normal	Normal	0	1	0	0
61	normal (11).jpg	Normal	Normal	0	1	0	0
62	normal (12).jpg	Normal	Normal	0	1	0	0
63	normal (13).jpg	Normal	Normal	0	1	0	0
64	normal (14).jpg	Normal	Normal	0	1	0	0
65	normal (15).jpg	Normal	Normal	0	1	0	0
66	normal (16).jpg	Normal	Tumor	0	0	1	0
67	normal (17).jpg	Normal	Normal	0	1	0	0
68	normal (18).jpg	Normal	Normal	0	1	0	0
69	normal (19).jpg	Normal	Normal	0	1	0	0
70	normal (20).jpg	Normal	Normal	0	1	0	0
71	normal (21).jpg	Normal	Normal	0	1	0	0
72	normal (22).jpg	Normal	Tumor	0	0	1	0
73	normal (23).jpg	Normal	Normal	0	1	0	0
74	normal (24).jpg	Normal	Normal	0	1	0	0
75	normal (25).jpg	Normal	Normal	0	1	0	0
76	normal (26).jpg	Normal	Normal	0	1	0	0
77	normal (27).jpg	Normal	Normal	0	1	0	0
78	normal (28).jpg	Normal	Tumor	0	0	1	0
79	normal (29).jpg	Normal	Normal	0	1	0	0

80	normal (30).jpg	Normal	Normal	0	1	0	0
81	normal (31).jpg	Normal	Normal	0	1	0	0
82	normal (32).jpg	Normal	Tumor	0	0	1	0
83	normal (33).jpg	Normal	Normal	0	1	0	0
84	normal (34).jpg	Normal	Normal	0	1	0	0
85	normal (35).jpg	Normal	Normal	0	1	0	0
86	normal (36).jpg	Normal	Normal	0	1	0	0
87	normal (37).jpg	Normal	Normal	0	1	0	0
88	normal (38).jpg	Normal	Normal	0	1	0	0
89	normal (39).jpg	Normal	Normal	0	1	0	0
90	normal (40).jpg	Normal	Tumor	0	0	1	0
91	normal (41).jpg	Normal	Normal	0	1	0	0
92	normal (42).jpg	Normal	Normal	0	1	0	0
93	normal (43).jpg	Normal	Normal	0	1	0	0
94	normal (44).jpg	Normal	Tumor	0	0	1	0
95	normal (45).jpg	Normal	Normal	0	1	0	0
96	normal (46).jpg	Normal	Normal	0	1	0	0
97	normal (47).jpg	Normal	Normal	0	1	0	0
98	normal (48).jpg	Normal	Tumor	0	0	1	0
99	normal (49).jpg	Normal	Normal	0	1	0	0
100	normal (50).jpg	Normal	Normal	0	1	0	0
Total				45	43	7	5

Lampiran 3

Tabel Hasil Uji Ctra Otak Fold-1

No	Gambar	Input	Output	TP	TN	FP	FN
1	tes(1).jpg	Tumor	Tumor	1	0	0	0
2	tes(2).jpg	Tumor	Tumor	1	0	0	0
3	tes(3).jpg	Tumor	Tumor	1	0	0	0
4	tes(4).jpg	Tumor	Tumor	1	0	0	0
5	tes(5).jpg	Tumor	Tumor	1	0	0	0
6	tes(6).jpg	Tumor	Tumor	1	0	0	0
7	tes(7).jpg	Tumor	Tumor	1	0	0	0
8	tes(8).jpg	Tumor	Tumor	1	0	0	0
9	tes(9).jpg	Tumor	Tumor	1	0	0	0
10	tes(10).jpg	Tumor	Normal	0	0	0	1
11	tes(11).jpg	Tumor	Tumor	1	0	0	0
12	tes(12).jpg	Tumor	Tumor	1	0	0	0
13	tes(13).jpg	Tumor	Tumor	1	0	0	0

14	tes(14).jpg	Tumor	Tumor	1	0	0	0
15	tes(15).jpg	Tumor	Normal	0	0	0	1
16	tes(16).jpg	Tumor	Tumor	1	0	0	0
17	tes(17).jpg	Tumor	Tumor	1	0	0	0
18	tes(18).jpg	Tumor	Tumor	1	0	0	0
19	tes(19).jpg	Tumor	Tumor	1	0	0	0
20	tes(20).jpg	Tumor	Tumor	1	0	0	0
21	tes(21).jpg	Tumor	Tumor	1	0	0	0
22	tes(22).jpg	Tumor	Tumor	1	0	0	0
23	tes(23).jpg	Tumor	Normal	0	0	0	1
24	tes(24).jpg	Tumor	Tumor	1	0	0	0
25	tes(25).jpg	Tumor	Tumor	1	0	0	0
26	tes(26).jpg	Tumor	Tumor	1	0	0	0
27	tes(27).jpg	Tumor	Normal	0	0	0	1
28	tes(28).jpg	Tumor	Tumor	1	0	0	0
29	tes(29).jpg	Tumor	Tumor	1	0	0	0
30	tes(30).jpg	Tumor	Tumor	1	0	0	0
31	tes(31).jpg	Tumor	Tumor	1	0	0	0
32	tes(32).jpg	Tumor	Normal	0	0	0	1
33	tes(33).jpg	Tumor	Tumor	1	0	0	0
34	tes(34).jpg	Tumor	Tumor	1	0	0	0
35	tes(35).jpg	Tumor	Tumor	1	0	0	0
36	tes(36).jpg	Tumor	Normal	0	0	0	1
37	tes(37).jpg	Tumor	Tumor	1	0	0	0
38	tes(38).jpg	Tumor	Tumor	1	0	0	0
39	tes(39).jpg	Tumor	Tumor	1	0	0	0
40	tes(40).jpg	Tumor	Tumor	1	0	0	0
41	tes(41).jpg	Tumor	Tumor	1	0	0	0
42	tes(42).jpg	Tumor	Normal	0	0	0	1
43	tes(43).jpg	Tumor	Tumor	1	0	0	0
44	tes(44).jpg	Tumor	Tumor	1	0	0	0
45	tes(45).jpg	Tumor	Tumor	1	0	0	0
46	tes(46).jpg	Tumor	Normal	0	0	0	1
47	tes(47).jpg	Tumor	Tumor	1	0	0	0
48	tes(48).jpg	Tumor	Tumor	1	0	0	0
49	tes(49).jpg	Tumor	Tumor	1	0	0	0
50	tes(50).jpg	Tumor	Tumor	1	0	0	0
51	tes(51).jpg	Tumor	Tumor	1	0	0	0
52	tes(52).jpg	Tumor	Tumor	1	0	0	0
53	tes(53).jpg	Tumor	Tumor	1	0	0	0
54	tes(54).jpg	Tumor	Normal	0	0	0	1
55	tes(55).jpg	Tumor	Tumor	1	0	0	0
56	tes(56).jpg	Tumor	Tumor	1	0	0	0
57	tes(57).jpg	Tumor	Tumor	1	0	0	0

58	tes(58).jpg	Tumor	Tumor	1	0	0	0
59	tes(59).jpg	Normal	Normal	0	1	0	0
60	tes(60).jpg	Normal	Normal	0	1	0	0
61	tes(61).jpg	Normal	Normal	0	1	0	0
62	tes(62).jpg	Normal	Normal	0	1	0	0
63	tes(63).jpg	Normal	Normal	0	1	0	0
64	tes(64).jpg	Normal	Normal	0	1	0	0
65	tes(65).jpg	Normal	Normal	0	1	0	0
66	tes(66).jpg	Normal	Normal	0	1	0	0
67	tes(67).jpg	Normal	Normal	0	1	0	0
68	tes(68).jpg	Normal	Normal	0	1	0	0
69	tes(69).jpg	Normal	Normal	0	1	0	0
70	tes(70).jpg	Normal	Normal	0	1	0	0
71	tes(71).jpg	Normal	Normal	0	1	0	0
72	tes(72).jpg	Normal	Normal	0	1	0	0
73	tes(73).jpg	Normal	Normal	0	1	0	0
74	tes(74).jpg	Normal	Normal	0	1	0	0
75	tes(75).jpg	Normal	Normal	0	1	0	0
76	tes(76).jpg	Normal	Normal	0	1	0	0
77	tes(77).jpg	Normal	Normal	0	1	0	0
78	tes(78).jpg	Normal	Normal	0	1	0	0
79	tes(79).jpg	Normal	Normal	0	1	0	0
80	tes(80).jpg	Normal	Normal	0	1	0	0
81	tes(81).jpg	Normal	Normal	0	1	0	0
82	tes(82).jpg	Normal	Normal	0	1	0	0
83	tes(83).jpg	Normal	Normal	0	1	0	0
84	tes(84).jpg	Normal	Normal	0	1	0	0
85	tes(85).jpg	Normal	Normal	0	1	0	0
86	tes(86).jpg	Normal	Normal	0	1	0	0
87	tes(87).jpg	Normal	Normal	0	1	0	0
88	tes(88).jpg	Normal	Normal	0	1	0	0
89	tes(89).jpg	Normal	Normal	0	1	0	0
90	tes(90).jpg	Normal	Normal	0	1	0	0
91	tes(91).jpg	Normal	Normal	0	1	0	0
92	tes(92).jpg	Normal	Normal	0	1	0	0
93	tes(93).jpg	Normal	Normal	0	1	0	0
94	tes(94).jpg	Normal	Normal	0	1	0	0
95	tes(95).jpg	Normal	Normal	0	1	0	0
96	tes(96).jpg	Normal	Normal	0	1	0	0
97	tes(97).jpg	Normal	Normal	0	1	0	0
98	tes(98).jpg	Normal	Normal	0	1	0	0
99	tes(99).jpg	Normal	Normal	0	1	0	0
100	tes(100).jpg	Normal	Normal	0	1	0	0
Total				49	42	0	9

Lampiran 4

Tabel Hasil Uji Ctra Otak Fold-2

No	Gambar	Input	Output	TP	TN	FP	FN
1	tes(101).jpg	Tumor	Tumor	1	0	0	0
2	tes(102).jpg	Tumor	Tumor	1	0	0	0
3	tes(103).jpg	Tumor	Tumor	1	0	0	0
4	tes(104).jpg	Tumor	Tumor	1	0	0	0
5	tes(105).jpg	Tumor	Tumor	1	0	0	0
6	tes(106).jpg	Tumor	Tumor	1	0	0	0
7	tes(107).jpg	Tumor	Tumor	1	0	0	0
8	tes(108).jpg	Tumor	Tumor	1	0	0	0
9	tes(109).jpg	Tumor	Tumor	1	0	0	0
10	tes(110).jpg	Tumor	Tumor	1	0	0	0
11	tes(111).jpg	Tumor	Tumor	1	0	0	0
12	tes(112).jpg	Tumor	Tumor	1	0	0	0
13	tes(113).jpg	Tumor	Tumor	1	0	0	0
14	tes(114).jpg	Tumor	Tumor	1	0	0	0
15	tes(115).jpg	Tumor	Tumor	1	0	0	0
16	tes(116).jpg	Tumor	Tumor	1	0	0	0
17	tes(117).jpg	Tumor	Tumor	1	0	0	0
18	tes(118).jpg	Tumor	Tumor	1	0	0	0
19	tes(119).jpg	Tumor	Tumor	1	0	0	0
20	tes(120).jpg	Tumor	Tumor	1	0	0	0
21	tes(121).jpg	Tumor	Tumor	1	0	0	0
22	tes(122).jpg	Tumor	Tumor	1	0	0	0
23	tes(123).jpg	Tumor	Tumor	1	0	0	0
24	tes(124).jpg	Tumor	Tumor	1	0	0	0
25	tes(125).jpg	Tumor	Tumor	1	0	0	0
26	tes(126).jpg	Tumor	Tumor	1	0	0	0
27	tes(127).jpg	Tumor	Tumor	1	0	0	0
28	tes(128).jpg	Tumor	Tumor	1	0	0	0
29	tes(129).jpg	Tumor	Normal	0	0	0	1
30	tes(130).jpg	Tumor	Tumor	1	0	0	0
31	tes(131).jpg	Tumor	Tumor	1	0	0	0
32	tes(132).jpg	Tumor	Tumor	1	0	0	0
33	tes(133).jpg	Tumor	Tumor	1	0	0	0
34	tes(134).jpg	Tumor	Tumor	1	0	0	0
35	tes(135).jpg	Tumor	Tumor	1	0	0	0
36	tes(136).jpg	Tumor	Tumor	1	0	0	0
37	tes(137).jpg	Tumor	Tumor	1	0	0	0

38	tes(138).jpg	Tumor	Tumor	1	0	0	0
39	tes(139).jpg	Tumor	Normal	0	0	0	1
40	tes(140).jpg	Tumor	Tumor	1	0	0	0
41	tes(141).jpg	Tumor	Tumor	1	0	0	0
42	tes(142).jpg	Tumor	Tumor	1	0	0	0
43	tes(143).jpg	Tumor	Tumor	1	0	0	0
44	tes(144).jpg	Tumor	Tumor	1	0	0	0
45	tes(145).jpg	Tumor	Normal	0	0	0	1
46	tes(146).jpg	Tumor	Tumor	1	0	0	0
47	tes(147).jpg	Tumor	Tumor	1	0	0	0
48	tes(148).jpg	Tumor	Tumor	1	0	0	0
49	tes(149).jpg	Tumor	Tumor	1	0	0	0
50	tes(150).jpg	Tumor	Normal	0	0	0	1
51	tes(151).jpg	Tumor	Tumor	1	0	0	0
52	tes(152).jpg	Tumor	Tumor	1	0	0	0
53	tes(153).jpg	Tumor	Tumor	1	0	0	0
54	tes(154).jpg	Tumor	Tumor	1	0	0	0
55	tes(155).jpg	Normal	Normal	0	1	0	0
56	tes(156).jpg	Normal	Normal	0	1	0	0
57	tes(157).jpg	Normal	Normal	0	1	0	0
58	tes(158).jpg	Normal	Normal	0	1	0	0
59	tes(159).jpg	Normal	Normal	0	1	0	0
60	tes(160).jpg	Normal	Normal	0	1	0	0
61	tes(161).jpg	Normal	Normal	0	1	0	0
62	tes(162).jpg	Normal	Normal	0	1	0	0
63	tes(163).jpg	Normal	Normal	0	1	0	0
64	tes(164).jpg	Normal	Normal	0	1	0	0
65	tes(165).jpg	Normal	Normal	0	1	0	0
66	tes(166).jpg	Normal	Normal	0	1	0	0
67	tes(167).jpg	Normal	Normal	0	1	0	0
68	tes(168).jpg	Normal	Normal	0	1	0	0
69	tes(169).jpg	Normal	Normal	0	1	0	0
70	tes(170).jpg	Normal	Normal	0	1	0	0
71	tes(171).jpg	Normal	Normal	0	1	0	0
72	tes(172).jpg	Normal	Normal	0	1	0	0
73	tes(173).jpg	Normal	Normal	0	1	0	0
74	tes(174).jpg	Normal	Normal	0	1	0	0
75	tes(175).jpg	Normal	Normal	0	1	0	0
76	tes(176).jpg	Normal	Normal	0	1	0	0
77	tes(177).jpg	Normal	Normal	0	1	0	0
78	tes(178).jpg	Normal	Normal	0	1	0	0
79	tes(179).jpg	Normal	Normal	0	1	0	0
80	tes(180).jpg	Normal	Normal	0	1	0	0
81	tes(181).jpg	Normal	Normal	0	1	0	0

82	tes(182).jpg	Normal	Normal	0	1	0	0
83	tes(183).jpg	Normal	Normal	0	1	0	0
84	tes(184).jpg	Normal	Normal	0	1	0	0
85	tes(185).jpg	Normal	Normal	0	1	0	0
86	tes(186).jpg	Normal	Tumor	0	0	1	0
87	tes(187).jpg	Normal	Normal	0	1	0	0
88	tes(188).jpg	Normal	Normal	0	1	0	0
89	tes(189).jpg	Normal	Normal	0	1	0	0
90	tes(190).jpg	Normal	Normal	0	1	0	0
91	tes(191).jpg	Normal	Normal	0	1	0	0
92	tes(192).jpg	Normal	Normal	0	1	0	0
93	tes(193).jpg	Normal	Normal	0	1	0	0
94	tes(194).jpg	Normal	Normal	0	1	0	0
95	tes(195).jpg	Normal	Normal	0	1	0	0
96	tes(196).jpg	Normal	Normal	0	1	0	0
97	tes(197).jpg	Normal	Normal	0	1	0	0
98	tes(198).jpg	Normal	Normal	0	1	0	0
99	tes(199).jpg	Normal	Normal	0	1	0	0
100	tes(200).jpg	Normal	Normal	0	1	0	0
Total				50	45	1	4

Lampiran 5

Tabel Hasil Uji Ctra Otak Fold-3

No	Gambar	Input	Output	TP	TN	FP	FN
1	tes(201).jpg	Tumor	Tumor	1	0	0	0
2	tes(202).jpg	Tumor	Tumor	1	0	0	0
3	tes(203).jpg	Tumor	Tumor	1	0	0	0
4	tes(204).jpg	Tumor	Tumor	1	0	0	0
5	tes(205).jpg	Tumor	Tumor	1	0	0	0
6	tes(206).jpg	Tumor	Tumor	1	0	0	0
7	tes(207).jpg	Tumor	Tumor	1	0	0	0
8	tes(208).jpg	Tumor	Normal	0	0	0	1
9	tes(209).jpg	Tumor	Tumor	1	0	0	0
10	tes(210).jpg	Tumor	Tumor	1	0	0	0
11	tes(211).jpg	Tumor	Tumor	1	0	0	0
12	tes(212).jpg	Tumor	Tumor	1	0	0	0
13	tes(213).jpg	Tumor	Tumor	1	0	0	0
14	tes(214).jpg	Tumor	Tumor	1	0	0	0
15	tes(215).jpg	Tumor	Tumor	1	0	0	0
16	tes(216).jpg	Tumor	Normal	0	0	0	1
17	tes(217).jpg	Tumor	Tumor	1	0	0	0

18	tes(218).jpg	Tumor	Tumor	1	0	0	0
19	tes(219).jpg	Tumor	Tumor	1	0	0	0
20	tes(220).jpg	Tumor	Tumor	1	0	0	0
21	tes(221).jpg	Tumor	Tumor	1	0	0	0
22	tes(222).jpg	Tumor	Tumor	1	0	0	0
23	tes(223).jpg	Tumor	Tumor	1	0	0	0
24	tes(224).jpg	Tumor	Normal	0	0	0	1
25	tes(225).jpg	Tumor	Tumor	1	0	0	0
26	tes(226).jpg	Tumor	Tumor	1	0	0	0
27	tes(227).jpg	Tumor	Tumor	1	0	0	0
28	tes(228).jpg	Tumor	Tumor	1	0	0	0
29	tes(229).jpg	Tumor	Tumor	1	0	0	0
30	tes(230).jpg	Tumor	Tumor	1	0	0	0
31	tes(231).jpg	Tumor	Tumor	1	0	0	0
32	tes(232).jpg	Tumor	Tumor	1	0	0	0
33	tes(233).jpg	Tumor	Tumor	1	0	0	0
34	tes(234).jpg	Tumor	Tumor	1	0	0	0
35	tes(235).jpg	Tumor	Tumor	1	0	0	0
36	tes(236).jpg	Tumor	Tumor	1	0	0	0
37	tes(237).jpg	Tumor	Normal	0	0	0	1
38	tes(238).jpg	Tumor	Tumor	1	0	0	0
39	tes(239).jpg	Tumor	Tumor	1	0	0	0
40	tes(240).jpg	Tumor	Tumor	1	0	0	0
41	tes(241).jpg	Tumor	Tumor	1	0	0	0
42	tes(242).jpg	Tumor	Normal	0	0	0	1
43	tes(243).jpg	Tumor	Tumor	1	0	0	0
44	tes(244).jpg	Tumor	Tumor	1	0	0	0
45	tes(245).jpg	Tumor	Tumor	1	0	0	0
46	tes(246).jpg	Tumor	Tumor	1	0	0	0
47	tes(247).jpg	Tumor	Tumor	1	0	0	0
48	tes(248).jpg	Tumor	Tumor	1	0	0	0
49	tes(249).jpg	Tumor	Tumor	1	0	0	0
50	tes(250).jpg	Tumor	Tumor	1	0	0	0
51	tes(251).jpg	Tumor	Tumor	1	0	0	0
52	tes(252).jpg	Tumor	Tumor	1	0	0	0
53	tes(253).jpg	Normal	Normal	0	1	0	0
54	tes(254).jpg	Normal	Normal	0	1	0	0
55	tes(255).jpg	Normal	Normal	0	1	0	0
56	tes(256).jpg	Normal	Normal	0	1	0	0
57	tes(257).jpg	Normal	Normal	0	1	0	0
58	tes(258).jpg	Normal	Normal	0	1	0	0
59	tes(259).jpg	Normal	Normal	0	1	0	0
60	tes(260).jpg	Normal	Tumor	0	0	1	0
61	tes(261).jpg	Normal	Normal	0	1	0	0

62	tes(262).jpg	Normal	Normal	0	1	0	0
63	tes(263).jpg	Normal	Normal	0	1	0	0
64	tes(264).jpg	Normal	Normal	0	1	0	0
65	tes(265).jpg	Normal	Normal	0	1	0	0
66	tes(266).jpg	Normal	Normal	0	1	0	0
67	tes(267).jpg	Normal	Tumor	0	0	1	0
68	tes(268).jpg	Normal	Normal	0	1	0	0
69	tes(269).jpg	Normal	Normal	0	1	0	0
70	tes(270).jpg	Normal	Normal	0	1	0	0
71	tes(271).jpg	Normal	Normal	0	1	0	0
72	tes(272).jpg	Normal	Tumor	0	0	1	0
73	tes(273).jpg	Normal	Normal	0	1	0	0
74	tes(274).jpg	Normal	Normal	0	1	0	0
75	tes(275).jpg	Normal	Normal	0	1	0	0
76	tes(276).jpg	Normal	Normal	0	1	0	0
77	tes(277).jpg	Normal	Normal	0	1	0	0
78	tes(278).jpg	Normal	Normal	0	1	0	0
79	tes(279).jpg	Normal	Normal	0	1	0	0
80	tes(280).jpg	Normal	Normal	0	1	0	0
81	tes(281).jpg	Normal	Normal	0	1	0	0
82	tes(282).jpg	Normal	Normal	0	1	0	0
83	tes(283).jpg	Normal	Normal	0	1	0	0
84	tes(284).jpg	Normal	Normal	0	1	0	0
85	tes(285).jpg	Normal	Normal	0	1	0	0
86	tes(286).jpg	Normal	Normal	0	1	0	0
87	tes(287).jpg	Normal	Tumor	0	0	1	0
88	tes(288).jpg	Normal	Normal	0	1	0	0
89	tes(289).jpg	Normal	Normal	0	1	0	0
90	tes(290).jpg	Normal	Normal	0	1	0	0
91	tes(291).jpg	Normal	Normal	0	1	0	0
92	tes(292).jpg	Normal	Normal	0	1	0	0
93	tes(293).jpg	Normal	Normal	0	1	0	0
94	tes(294).jpg	Normal	Normal	0	1	0	0
95	tes(295).jpg	Normal	Normal	0	1	0	0
96	tes(296).jpg	Normal	Normal	0	1	0	0
97	tes(297).jpg	Normal	Normal	0	1	0	0
98	tes(298).jpg	Normal	Normal	0	1	0	0
99	tes(299).jpg	Normal	Normal	0	1	0	0
100	tes(300).jpg	Normal	Normal	0	1	0	0
Total				47	44	4	5

Lampiran 6

Tabel Hasil Uji Ctra Otak Fold-4

No	Gambar	Input	Output	TP	TN	FP	FN
1	tes(301).jpg	Tumor	Tumor	1	0	0	0
2	tes(302).jpg	Tumor	Tumor	1	0	0	0
3	tes(303).jpg	Tumor	Tumor	1	0	0	0
4	tes(304).jpg	Tumor	Tumor	1	0	0	0
5	tes(305).jpg	Tumor	Tumor	1	0	0	0
6	tes(306).jpg	Tumor	Tumor	1	0	0	0
7	tes(307).jpg	Tumor	Tumor	1	0	0	0
8	tes(308).jpg	Tumor	Normal	0	0	0	1
9	tes(309).jpg	Tumor	Tumor	1	0	0	0
10	tes(310).jpg	Tumor	Tumor	1	0	0	0
11	tes(311).jpg	Tumor	Tumor	1	0	0	0
12	tes(312).jpg	Tumor	Tumor	1	0	0	0
13	tes(313).jpg	Tumor	Normal	0	0	0	1
14	tes(314).jpg	Tumor	Tumor	1	0	0	0
15	tes(315).jpg	Tumor	Tumor	1	0	0	0
16	tes(316).jpg	Tumor	Tumor	1	0	0	0
17	tes(317).jpg	Tumor	Tumor	1	0	0	0
18	tes(318).jpg	Tumor	Tumor	1	0	0	0
19	tes(319).jpg	Tumor	Tumor	1	0	0	0
20	tes(320).jpg	Tumor	Tumor	1	0	0	0
21	tes(321).jpg	Tumor	Tumor	1	0	0	0
22	tes(322).jpg	Tumor	Tumor	1	0	0	0
23	tes(323).jpg	Tumor	Tumor	1	0	0	0
24	tes(324).jpg	Tumor	Normal	0	0	0	1
25	tes(325).jpg	Tumor	Tumor	1	0	0	0
26	tes(326).jpg	Tumor	Tumor	1	0	0	0
27	tes(327).jpg	Tumor	Tumor	1	0	0	0
28	tes(328).jpg	Tumor	Tumor	1	0	0	0
29	tes(329).jpg	Tumor	Tumor	1	0	0	0
30	tes(330).jpg	Tumor	Tumor	1	0	0	0
31	tes(331).jpg	Tumor	Tumor	1	0	0	0
32	tes(332).jpg	Tumor	Tumor	1	0	0	0
33	tes(333).jpg	Tumor	Tumor	1	0	0	0
34	tes(334).jpg	Tumor	Tumor	1	0	0	0
35	tes(335).jpg	Tumor	Tumor	1	0	0	0
36	tes(336).jpg	Tumor	Tumor	1	0	0	0
37	tes(337).jpg	Tumor	Tumor	1	0	0	0
38	tes(338).jpg	Tumor	Tumor	1	0	0	0

39	tes(339).jpg	Tumor	Tumor	1	0	0	0
40	tes(340).jpg	Tumor	Tumor	1	0	0	0
41	tes(341).jpg	Tumor	Tumor	1	0	0	0
42	tes(342).jpg	Tumor	Tumor	1	0	0	0
43	tes(343).jpg	Tumor	Tumor	1	0	0	0
44	tes(344).jpg	Tumor	Tumor	1	0	0	0
45	tes(345).jpg	Tumor	Normal	0	0	0	1
46	tes(346).jpg	Tumor	Tumor	1	0	0	0
47	tes(347).jpg	Tumor	Tumor	1	0	0	0
48	tes(348).jpg	Tumor	Tumor	1	0	0	0
49	tes(349).jpg	Tumor	Tumor	1	0	0	0
50	tes(350).jpg	Tumor	Tumor	1	0	0	0
51	tes(351).jpg	Tumor	Tumor	1	0	0	0
52	tes(352).jpg	Tumor	Tumor	1	0	0	0
53	tes(353).jpg	Tumor	Tumor	1	0	0	0
54	tes(354).jpg	Tumor	Tumor	1	0	0	0
55	tes(355).jpg	Tumor	Tumor	1	0	0	0
56	tes(356).jpg	Normal	Normal	0	1	0	0
57	tes(357).jpg	Normal	Normal	0	1	0	0
58	tes(358).jpg	Normal	Normal	0	1	0	0
59	tes(359).jpg	Normal	Normal	0	1	0	0
60	tes(360).jpg	Normal	Normal	0	1	0	0
61	tes(361).jpg	Normal	Normal	0	1	0	0
62	tes(362).jpg	Normal	Normal	0	1	0	0
63	tes(363).jpg	Normal	Normal	0	1	0	0
64	tes(364).jpg	Normal	Normal	0	1	0	0
65	tes(365).jpg	Normal	Normal	0	1	0	0
66	tes(366).jpg	Normal	Normal	0	1	0	0
67	tes(367).jpg	Normal	Normal	0	1	0	0
68	tes(368).jpg	Normal	Normal	0	1	0	0
69	tes(369).jpg	Normal	Normal	0	1	0	0
70	tes(370).jpg	Normal	Normal	0	1	0	0
71	tes(371).jpg	Normal	Normal	0	1	0	0
72	tes(372).jpg	Normal	Normal	0	1	0	0
73	tes(373).jpg	Normal	Tumor	0	0	1	0
74	tes(374).jpg	Normal	Normal	0	1	0	0
75	tes(375).jpg	Normal	Normal	0	1	0	0
76	tes(376).jpg	Normal	Normal	0	1	0	0
77	tes(377).jpg	Normal	Normal	0	1	0	0
78	tes(378).jpg	Normal	Normal	0	1	0	0
79	tes(379).jpg	Normal	Normal	0	1	0	0
80	tes(380).jpg	Normal	Normal	0	1	0	0
81	tes(381).jpg	Normal	Normal	0	1	0	0
82	tes(382).jpg	Normal	Normal	0	1	0	0

83	tes(383).jpg	Normal	Normal	0	1	0	0
84	tes(384).jpg	Normal	Normal	0	1	0	0
85	tes(385).jpg	Normal	Normal	0	1	0	0
86	tes(386).jpg	Normal	Tumor	0	0	1	0
87	tes(387).jpg	Normal	Normal	0	1	0	0
88	tes(388).jpg	Normal	Normal	0	1	0	0
89	tes(389).jpg	Normal	Normal	0	1	0	0
90	tes(390).jpg	Normal	Normal	0	1	0	0
91	tes(391).jpg	Normal	Normal	0	1	0	0
92	tes(392).jpg	Normal	Normal	0	1	0	0
93	tes(393).jpg	Normal	Normal	0	1	0	0
94	tes(394).jpg	Normal	Normal	0	1	0	0
95	tes(395).jpg	Normal	Normal	0	1	0	0
96	tes(396).jpg	Normal	Normal	0	1	0	0
97	tes(397).jpg	Normal	Normal	0	1	0	0
98	tes(398).jpg	Normal	Normal	0	1	0	0
99	tes(399).jpg	Normal	Normal	0	1	0	0
100	tes(400).jpg	Normal	Normal	0	1	0	0
Total				51	43	2	4

Lampiran 7

Tabel Hasil Uji Ctra Otak Fold-5

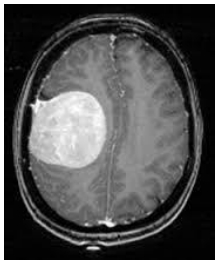
No	Gambar	Input	Output	TP	TN	FP	FN
1	tes(401).jpg	Tumor	Tumor	1	0	0	0
2	tes(402).jpg	Tumor	Tumor	1	0	0	0
3	tes(403).jpg	Tumor	Tumor	1	0	0	0
4	tes(404).jpg	Tumor	Tumor	1	0	0	0
5	tes(405).jpg	Tumor	Tumor	1	0	0	0
6	tes(406).jpg	Tumor	Tumor	1	0	0	0
7	tes(407).jpg	Tumor	Tumor	1	0	0	0
8	tes(408).jpg	Tumor	Tumor	1	0	0	0
9	tes(409).jpg	Tumor	Tumor	1	0	0	0
10	tes(410).jpg	Tumor	Tumor	1	0	0	0
11	tes(411).jpg	Tumor	Tumor	1	0	0	0
12	tes(412).jpg	Tumor	Tumor	1	0	0	0
13	tes(413).jpg	Tumor	Normal	0	0	0	1
14	tes(414).jpg	Tumor	Tumor	1	0	0	0
15	tes(415).jpg	Tumor	Tumor	1	0	0	0
16	tes(416).jpg	Tumor	Tumor	1	0	0	0
17	tes(417).jpg	Tumor	Tumor	1	0	0	0
18	tes(418).jpg	Tumor	Tumor	1	0	0	0
19	tes(419).jpg	Tumor	Tumor	1	0	0	0

20	tes(420).jpg	Tumor	Normal	0	0	0	1
21	tes(421).jpg	Tumor	Tumor	1	0	0	0
22	tes(422).jpg	Tumor	Tumor	1	0	0	0
23	tes(423).jpg	Tumor	Tumor	1	0	0	0
24	tes(424).jpg	Tumor	Tumor	1	0	0	0
25	tes(425).jpg	Tumor	Tumor	1	0	0	0
26	tes(426).jpg	Tumor	Tumor	1	0	0	0
27	tes(427).jpg	Tumor	Tumor	1	0	0	0
28	tes(428).jpg	Tumor	Tumor	1	0	0	0
29	tes(429).jpg	Tumor	Tumor	1	0	0	0
30	tes(430).jpg	Tumor	Normal	0	0	0	1
31	tes(431).jpg	Tumor	Tumor	1	0	0	0
32	tes(432).jpg	Tumor	Tumor	1	0	0	0
33	tes(433).jpg	Tumor	Tumor	1	0	0	0
34	tes(434).jpg	Tumor	Tumor	1	0	0	0
35	tes(435).jpg	Tumor	Tumor	1	0	0	0
36	tes(436).jpg	Tumor	Tumor	1	0	0	0
37	tes(437).jpg	Tumor	Tumor	1	0	0	0
38	tes(438).jpg	Tumor	Tumor	1	0	0	0
39	tes(439).jpg	Tumor	Normal	0	0	0	1
40	tes(440).jpg	Tumor	Tumor	1	0	0	0
41	tes(441).jpg	Tumor	Tumor	1	0	0	0
42	tes(442).jpg	Tumor	Tumor	1	0	0	0
43	tes(443).jpg	Tumor	Tumor	1	0	0	0
44	tes(444).jpg	Tumor	Normal	0	0	0	1
45	tes(445).jpg	Tumor	Tumor	1	0	0	0
46	tes(446).jpg	Tumor	Tumor	1	0	0	0
47	tes(447).jpg	Tumor	Tumor	1	0	0	0
48	tes(448).jpg	Tumor	Tumor	1	0	0	0
49	tes(449).jpg	Tumor	Tumor	1	0	0	0
50	tes(450).jpg	Tumor	Tumor	1	0	0	0
51	tes(451).jpg	Tumor	Tumor	1	0	0	0
52	tes(452).jpg	Tumor	Tumor	1	0	0	0
53	tes(453).jpg	Tumor	Tumor	1	0	0	0
54	tes(454).jpg	Tumor	Tumor	1	0	0	0
55	tes(455).jpg	Tumor	Tumor	1	0	0	0
56	tes(456).jpg	Normal	Normal	0	1	0	0
57	tes(457).jpg	Normal	Normal	0	1	0	0
58	tes(458).jpg	Normal	Normal	0	1	0	0
59	tes(459).jpg	Normal	Normal	0	1	0	0
60	tes(460).jpg	Normal	Normal	0	1	0	0
61	tes(461).jpg	Normal	Normal	0	1	0	0
62	tes(462).jpg	Normal	Normal	0	1	0	0
63	tes(463).jpg	Normal	Normal	0	1	0	0

64	tes(464).jpg	Normal	Normal	0	1	0	0
65	tes(465).jpg	Normal	Normal	0	1	0	0
66	tes(466).jpg	Normal	Normal	0	1	0	0
67	tes(467).jpg	Normal	Normal	0	1	0	0
68	tes(468).jpg	Normal	Normal	0	1	0	0
69	tes(469).jpg	Normal	Normal	0	1	0	0
70	tes(470).jpg	Normal	Normal	0	1	0	0
71	tes(471).jpg	Normal	Normal	0	1	0	0
72	tes(472).jpg	Normal	Normal	0	1	0	0
73	tes(473).jpg	Normal	Normal	0	1	0	0
74	tes(474).jpg	Normal	Normal	0	1	0	0
75	tes(475).jpg	Normal	Normal	0	1	0	0
76	tes(476).jpg	Normal	Normal	0	1	0	0
77	tes(477).jpg	Normal	Normal	0	1	0	0
78	tes(478).jpg	Normal	Normal	0	1	0	0
79	tes(479).jpg	Normal	Normal	0	1	0	0
80	tes(480).jpg	Normal	Normal	0	1	0	0
81	tes(481).jpg	Normal	Normal	0	1	0	0
82	tes(482).jpg	Normal	Normal	0	1	0	0
83	tes(483).jpg	Normal	Normal	0	1	0	0
84	tes(484).jpg	Normal	Normal	0	1	0	0
85	tes(485).jpg	Normal	Normal	0	1	0	0
86	tes(486).jpg	Normal	Normal	0	1	0	0
87	tes(487).jpg	Normal	Normal	0	1	0	0
88	tes(488).jpg	Normal	Normal	0	1	0	0
89	tes(489).jpg	Normal	Normal	0	1	0	0
90	tes(490).jpg	Normal	Normal	0	1	0	0
91	tes(491).jpg	Normal	Normal	0	1	0	0
92	tes(492).jpg	Normal	Normal	0	1	0	0
93	tes(493).jpg	Normal	Tumor	0	0	1	0
94	tes(494).jpg	Normal	Normal	0	1	0	0
95	tes(495).jpg	Normal	Normal	0	1	0	0
96	tes(496).jpg	Normal	Normal	0	1	0	0
97	tes(497).jpg	Normal	Normal	0	1	0	0
98	tes(498).jpg	Normal	Normal	0	1	0	0
99	tes(499).jpg	Normal	Normal	0	1	0	0
100	tes(500).jpg	Normal	Normal	0	1	0	0
Total				50	44	1	5

Lampiran 8 Perhitungan Manual

Adapun salah satu contoh input citra yang digunakan pada penelitian ini untuk menampilkan proses ekstrasi CNN kedalam bentuk matriks.



Gambar Contoh citra sebagai input

Gambar tersebut merupakan salah satu input dengan label "1_Tumor.jpg" dengan ukuran 256x256, adapun representasi matriks sebagai berikut:

Tabel 1 Representasi Citra Input Ke Dalam Bentuk Matriks

Matriks Input														
128	113	127	138	137	120	107	103	96	83	85	80	77	74	66
137	118	123	129	117	98	88	73	54	46	42	50	55	52	47
136	120	101	104	89	72	68	63	56	57	69	76	82	81	78
128	113	84	86	73	63	70	78	85	95	109	119	128	133	137
105	85	72	78	76	81	100	115	127	141	155	164	173	179	186
80	59	69	83	96	114	140	154	161	174	188	194	197	196	198
82	72	100	117	135	159	184	190	191	200	196	200	200	196	195
109	115	145	156	169	187	205	205	202	211	196	198	198	194	194
132	146	167	173	176	187	199	195	192	202	202	203	200	194	192
156	171	168	161	154	151	154	161	172	180	193	202	209	194	210
155	160	143	143	140	134	127	129	141	153	176	199	213	183	174
146	140	132	133	131	126	123	125	130	135	152	188	204	159	134
132	123	133	130	127	129	138	144	140	133	145	177	182	126	104
123	120	129	133	135	135	139	144	143	137	154	173	163	104	91
122	130	138	152	156	143	130	132	141	147	157	166	155	105	98

Tabel 2 Representasi Citra Input Ke Dalam Bentuk Matriks Setelah Proses Padding

Matriks Input															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	128	113	127	138	137	120	107	103	96	83	85	80	77	74	66
0	137	118	123	129	117	98	88	73	54	46	42	50	55	52	47
0	136	120	101	104	89	72	68	63	56	57	69	76	82	81	78
0	128	113	84	86	73	63	70	78	85	95	109	119	128	133	137
0	105	85	72	78	76	81	100	115	127	141	155	164	173	179	186
0	80	59	69	83	96	114	140	154	161	174	188	194	197	196	198
0	82	72	100	117	135	159	184	190	191	200	196	200	200	196	195
0	109	115	145	156	169	187	205	205	202	211	196	198	198	194	194
0	132	146	167	173	176	187	199	195	192	202	202	203	200	194	192
0	156	171	168	161	154	151	154	161	172	180	193	202	209	194	210
0	155	160	143	143	140	134	127	129	141	153	176	199	213	183	174
0	146	140	132	133	131	126	123	125	130	135	152	188	204	159	134
0	132	123	133	130	127	129	138	144	140	133	145	177	182	126	104
0	123	120	129	133	135	135	139	144	143	137	154	173	163	104	91
0	122	130	138	152	156	143	130	132	141	147	157	166	155	105	98

Kemudian diberikan contoh dari representasi matriks untuk kernel yang digunakan pada *layer* ini

Tabel 3 Representasi Filter Kernel Ke Dalam Bentuk Matriks Untuk *Convolution Layer* 1

Filter Input		
-0,0515038200	0.0077721600	0.0512389700
0,0365313000	-0,0629468600	0,0986095700
-0,1305493600	0,0329678400	0,0965579800

Dengan menggunakan persamaan 3.1, adapun salah satu contoh perhitungan untuk satu elemen matriks sebagai berikut:

G[1,1] = (0 * -0,0515038200) + (0 * 0.0077721600) + (0 * 0.0512389700) + (0 * 0,0365313000) + (128 * -0,0629468600) + (113 * 0,0986095700) + (0 * -0,1305493600) + (137 * 0,0329678400) + (118 * 0,0965579800) = 26,52

Sehingga didapatkan hasil keseluruhan matriks sebagai berikut

Tabel 4 Matriks Hasil Perhitungan Pada Proses Konvolusi 1

Matriks Input											
26,52	6,48	9,31	6,29	1,48	0,23	0,47	-0,02	0,03	1,26	4,79	4,28
26,34	1,73	5,44	2,94	-0,73	0,93	3,44	3,89	5,95	10,17	11,96	14,00
22,66	-0,63	3,17	3,89	2,61	6,61	10,39	11,72	13,83	17,09	18,71	18,76
17,54	-0,19	3,74	7,67	8,95	14,11	17,59	17,87	19,20	21,26	21,64	21,00
16,48	5,03	10,08	13,69	16,35	21,20	22,20	20,35	21,06	21,00	19,93	19,41
21,19	13,62	18,27	19,58	22,02	24,72	23,37	20,17	20,61	18,55	17,83	16,77
27,69	18,94	21,78	20,23	21,33	22,89	20,05	16,42	17,91	16,54	16,24	16,04
34,22	18,71	17,79	14,47	14,79	15,31	14,06	13,22	14,94	15,89	15,31	17,32
35,50	14,15	12,67	10,49	9,15	8,93	8,74	9,28	11,53	15,62	17,04	18,59
33,67	9,24	8,46	7,78	7,31	7,32	8,69	10,24	11,00	33,67	9,24	8,46
30,03	8,71	8,07	8,81	8,53	9,60	11,01	11,73	11,31	30,03	8,71	8,07
28,03	9,41	9,38	10,58	10,44	11,36	12,58	12,54	11,48	28,03	9,41	9,38
28,38	11,09	13,68	12,90	11,33	10,48	11,40	11,84	11,22	28,38	11,09	13,68
29,95	14,52	16,98	13,92	9,72	7,57	9,66	11,75	11,77	29,95	14,52	16,98

Kemudian hasil dari Tabel tersebut di hitung dengan rumus fungsi aktivasi ReLu pada persamaan 3.2, sebagai berikut:

R(z) = max(0,26.52)

R(z) = 26.52

Sehingga didapatkan hasil keseluruhan matriks dari sebagai berikut

Tabel 5 Matriks Hasil Perhitungan Setelah Melewati Fungsi Aktifasi ReLu

Matriks Input										
26,52	6,48	9,31	6,29	1,48	0,23	0,47	0,00	0,03	1,26	4,79
26,34	1,73	5,44	2,94	0,00	0,93	3,44	3,89	5,95	10,17	11,96
22,66	0,00	3,17	3,89	2,61	6,61	10,39	11,72	13,83	17,09	18,71
17,54	0,00	3,74	7,67	8,95	14,11	17,59	17,87	19,20	21,26	21,64
16,48	5,03	10,08	13,69	16,35	21,20	22,20	20,35	21,06	21,00	19,93
21,19	13,62	18,27	19,58	22,02	24,72	23,37	20,17	20,61	18,55	17,83
27,69	18,94	21,78	20,23	21,33	22,89	20,05	16,42	17,91	16,54	16,24
34,22	18,71	17,79	14,47	14,79	15,31	14,06	13,22	14,94	15,89	15,31

35,50	14,15	12,67	10,49	9,15	8,93	8,74	9,28	11,53	15,62	17,04
33,67	9,24	8,46	7,78	7,31	7,32	8,69	10,24	11,00	13,58	18,31
30,03	8,71	8,07	8,81	8,53	9,60	11,01	11,73	11,31	13,21	19,51
28,03	9,41	9,38	10,58	10,44	11,36	12,58	12,54	11,48	14,19	19,54
28,38	11,09	13,68	12,90	11,33	10,48	11,40	11,84	11,22	14,19	17,45
29,95	14,52	16,98	13,92	9,72	7,57	9,66	11,75	11,77	13,01	16,11
31,03	17,93	15,85	10,29	4,48	3,46	7,85	11,23	10,41	12,62	15,22

Berikut perhitungan untuk *Average Pooling layer 1* sebagai berikut:

Tabel 6 Matriks Perhitungan Untuk Proses Average Pooling 1

Fiter Input			
26,52	6,48	9,31	6,29
26,34	1,73	5,44	2,94

G[1,1] = avg (26.52, 6.48, 9.31, 26.34, 1.73, 5.44, 22.66, 0.00, 3.17) = 11,29

G[1,2] = avg (6.29, 1.48, 0.23, 2.94, 0.00, 0.93, 3.89, 2.61, 6.61) = 2,78

Sehingga didapatkan hasil keseluruhan matriks sebagai berikut

Tabel 7 Matriks Hasil Perhitungan Setelah Proses *Average Pooling* 1

Matriks Input														
11,29	2,78	13,18	14,15	12,86	10,50	8,89	8,98	9,49	9,09	7,64	6,21	5,55	5,23	10,11
10,97	10,22	10,65	10,11	23,11	24,48	23,79	19,76	19,09	16,94	26,78	27,47	21,04	18,92	10,97
8,46	6,53	5,97	4,96	12,55	12,27	17,33	18,45	17,94	12,99	19,04	12,81	10,74	11,54	8,46
3,99	2,33	2,46	3,02	12,59	8,25	10,38	8,05	4,94	3,26	18,94	23,07	19,93	16,51	3,99
2,12	2,51	3,89	6,07	20,96	16,50	20,06	6,62	20,21	33,05	30,89	28,68	20,88	18,65	2,12
5,97	7,38	8,77	10,89	17,01	15,47	16,81	21,49	36,57	33,42	20,45	17,53	16,53	7,44	5,97
11,68	12,88	12,82	13,00	8,85	9,26	20,12	24,78	29,91	20,04	15,12	18,02	12,34	6,84	11,68
14,39	13,97	11,83	10,00	5,96	9,86	21,02	28,07	11,28	7,25	9,01	15,47	12,61	5,32	14,39
12,44	10,95	7,83	5,36	7,08	11,02	20,39	25,09	18,16	8,11	5,75	8,86	10,38	5,01	12,44
7,86	6,38	3,65	2,10	8,65	11,41	22,24	21,00	24,23	12,54	12,39	7,03	6,15	5,88	7,86
4,36	3,90	2,78	2,91	9,56	12,23	20,71	22,95	15,98	9,38	10,45	6,67	4,41	6,60	4,36
3,17	4,27	5,30	7,18	7,87	9,28	10,27	24,80	9,13	7,11	9,18	6,28	5,52	6,42	3,17
5,25	7,79	10,57	13,34	9,19	10,09	10,98	25,12	10,45	7,79	9,73	9,90	7,27	5,85	5,25
10,19	13,37	16,48	18,95	14,29	14,49	17,36	23,46	7,48	9,36	10,65	10,89	6,86	5,33	10,19

15,69	18,15	20,22	21,57	13,88	13,61	21,61	17,17	5,29	7,54	7,34	7,04	5,43	6,63	15,69

Tabel 8 Matriks Input Setelah Proses Padding

Matriks Input															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	26,52	6,61	9,45	13,10	6,54	8,65	11,41	22,24	21,00	24,23	12,54	12,39	7,03	6,15	27,90
0	32,75	6,51	6,48	8,40	5,63	9,56	12,23	20,71	22,95	15,98	9,38	10,45	6,67	4,41	32,75
0	31,57	5,23	6,43	6,67	6,45	7,87	9,28	10,27	24,80	9,13	7,11	9,18	6,28	5,52	31,57
0	26,85	5,32	6,09	5,99	8,36	9,19	10,09	10,98	25,12	10,45	7,79	9,73	9,90	7,27	26,85
0	19,08	6,61	7,63	9,49	12,89	14,29	14,49	17,36	23,46	7,48	9,36	10,65	10,89	6,86	19,08
0	15,50	6,58	6,40	10,15	14,32	13,88	13,61	21,61	17,17	5,29	7,54	7,34	7,04	5,43	15,50
0	16,53	6,92	6,31	11,04	15,21	17,02	16,33	22,85	15,73	15,52	10,35	10,07	8,89	7,03	16,53
0	18,24	6,89	6,76	15,80	18,56	14,76	6,78	19,92	22,39	24,67	10,67	12,89	12,61	8,65	18,24
0	19,11	6,88	8,91	16,64	12,49	7,35	9,58	10,30	27,20	18,47	7,24	8,38	8,04	6,00	19,11
0	18,02	6,84	8,40	7,39	5,60	10,15	12,03	13,80	29,95	11,44	7,22	8,52	8,81	5,42	18,02
0	18,27	6,46	8,26	5,90	9,08	11,77	9,90	21,20	27,68	4,11	7,55	7,40	7,11	6,18	18,27
0	17,81	7,98	7,90	6,71	10,73	8,07	10,14	18,49	20,60	15,15	8,91	6,34	5,97	7,17	17,81
0	17,05	8,06	6,34	6,43	6,23	8,68	8,39	12,20	22,51	4,37	7,12	7,36	6,94	7,39	17,05
0	17,77	7,12	6,46	5,72	5,76	9,42	9,05	10,68	16,93	7,33	8,02	5,34	7,09	7,31	17,77
0	18,16	6,63	6,67	6,97	5,43	9,70	9,33	12,30	18,77	9,20	10,29	5,87	6,38	9,15	18,16

Tabel 9 Representasi Filter Ke Dalam Bentuk Matriks Untuk *Convolution Layer* 2

Filter Input		
0.0077721600	0,0588773800	0,0421962400
-0,1181599100	-0,1125107000	-0,0538383200
-0,0515038200	0,0767164400	0,1069133400

Berikut salah satu contoh perhitungan untuk *Convolution Layer* 2:

G[1,1] = (0 * 0.0077721600) + (0 * 0.0077721600) + (0 * 0.0512389700) + (0 * 0,0365313000) + (128 * -0,1125107000) + (113 * -0,0538383200) + (0 * --0,0515038200) + (137 * 0,0767164400) + (118 * 0,1069133400) = 0,84

Berikut hasil dari perhitungan matriks keseluruhan:

Tabel 10 Matriks Hasil Perhitungan Setelah Proses Convolution 2

Matriks Input														
0,84	-6,61	-0,37	-0,91	-1,62	-0,62	-1,27	-0,77	-3,41	-4,24	-3,60	-1,75	-1,99	-0,79	-0,53
1,00	-6,96	-0,65	-0,57	-0,60	-0,01	-0,17	0,63	-1,91	-4,24	-1,68	-0,66	-1,26	-0,84	-0,59
0,94	-6,01	-0,04	0,12	0,23	0,08	0,10	0,51	-1,58	-5,22	-1,09	-0,66	-1,63	-1,43	-0,81
1,20	-4,88	-0,63	-0,41	-0,66	-1,51	-1,02	-1,01	-2,74	-5,01	-1,53	-1,55	-2,03	-1,75	-0,83
1,31	-3,35	-0,35	-0,03	-0,40	-1,10	-0,85	-1,33	-2,41	-2,74	0,00	-0,46	-0,90	-1,31	-0,88
1,09	-3,11	0,07	0,79	-0,81	-3,16	-2,45	-1,06	-2,38	-2,68	-1,41	-0,73	-0,99	-1,24	-0,81
1,04	-3,21	0,02	-0,48	-2,81	-2,80	-1,85	-0,51	-3,29	-4,50	-4,28	-2,18	-2,57	-2,19	-1,20
0,96	-3,66	-1,07	-1,57	-2,43	-1,79	-0,66	1,74	-1,81	-4,97	-3,76	-0,64	-1,42	-1,64	-0,90
1,11	-3,56	-0,51	-0,30	-1,02	-1,17	-0,22	0,56	-2,04	-6,08	-2,64	-1,03	-1,39	-1,26	-0,70
1,17	-3,35	-0,74	-0,83	-0,82	-0,86	-0,84	-0,83	-2,15	-4,68	-1,04	-1,02	-1,07	-1,07	-0,60
1,08	-3,66	-1,10	-1,39	-0,73	-1,15	-1,10	0,16	-3,88	-6,35	-1,56	-0,98	-0,72	-0,77	-0,86
1,16	-3,43	-1,14	-0,87	-0,33	-0,86	-0,43	-0,24	-2,50	-3,37	-1,55	-1,35	-0,85	-0,73	-1,02
1,06	-3,41	-0,96	-0,75	-0,45	-0,39	-0,75	0,39	-1,14	-3,65	-0,49	-0,98	-0,53	-0,42	-0,74
1,13	-3,45	-0,79	-0,97	-0,56	-0,23	-0,99	-0,55	-1,40	-2,28	-1,03	-1,49	0,01	-0,33	-1,03
1,12	-3,53	-0,79	-1,08	-0,74	-0,12	-0,92	-0,77	-1,72	-3,13	-2,57	-2,38	0,28	-0,20	-1,26

Kemudian hasil dari Tabel tersebut di hitung dengan rumus fungsi aktivasi ReLu, berikut hasil dari matriks tersebut:

$R(z) = \max(0, 0.84)$

$R(z) = 0.84$

Tabel 11 Matriks Hasil Perhitungan Setelah Melewati Fungsi Aktifasi ReLu

Matriks Input														
0,84	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,63	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,94	0,00	0,00	0,12	0,23	0,08	0,10	0,51	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,31	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,09	0,00	0,07	0,79	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,04	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,96	0,00	0,00	0,00	0,00	0,00	0,00	1,74	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,11	0,00	0,00	0,00	0,00	0,00	0,00	0,56	0,00	0,00	0,00	0,00	0,00	0,00	0,00

1,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,08	0,00	0,00	0,00	0,00	0,00	0,00	0,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,06	0,00	0,00	0,00	0,00	0,00	0,00	0,39	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00
1,12	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,28	0,00	0,00

Berikut perhitungan untuk *Average Pooling layer 2* sebagai berikut:

Tabel 12 Matriks Perhitungan Untuk Proses Average Pooling 2

Filter Input			
0,94	0,00	0,00	0,12
1,20	0,00	0,00	0,00

G[1,1] = avg (0.94, 0.00, 0.00, 1.20, 0.00, 0.00, 1.31, 0.00, 0.00) = 0,38

G[1,2] = avg (0.12, 0.23, 0.02, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00) = 0.48

Sehingga didapatkan hasil keseluruhan matriks sebagai berikut

Tabel 13 Matriks Hasil Perhitungan Setelah Proses Average Pooling 2

Matriks Input														
0,38	0,48	0,51	0,00	0,00	0,00	0,00	0,00	0,53	0,12	0,23	0,51	0,00	0,00	1,31
1,09	0,79	1,74	0,00	0,00	0,00	0,00	0,00	0,00	0,79	0,00	1,74	0,00	0,00	1,09
1,17	0,00	0,56	0,00	0,00	0,00	0,00	0,35	0,76	0,00	0,00	0,56	0,00	0,00	1,17
1,16	0,00	0,39	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,39	0,01	0,00	1,16
1,40	0,00	0,00	0,00	1,07	0,00	0,00	0,00	0,89	0,00	0,00	0,00	1,07	0,00	1,40
1,83	0,00	0,00	0,20	0,68	0,00	0,00	0,00	0,05	0,39	0,00	0,00	0,68	0,00	1,83
2,02	0,00	0,00	0,70	0,00	0,00	0,00	0,00	0,00	0,66	0,00	0,00	0,70	0,00	2,02
2,18	0,00	0,01	0,51	0,00	0,00	0,00	0,00	0,24	0,99	0,01	0,00	0,51	0,00	2,18
1,96	0,00	0,00	0,00	0,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,09	0,00	1,96
1,44	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	1,44
1,95	0,04	0,00	0,00	0,00	0,00	0,00	0,00	1,09	0,04	0,00	0,00	0,00	0,00	1,95
2,22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2,46	0,00	0,00	0,00	0,00	0,00	2,22
2,65	0,01	0,71	0,00	0,00	0,00	0,00	0,21	2,33	0,00	0,01	0,71	0,00	0,00	2,65
2,61	0,01	0,00	0,00	0,16	0,83	0,00	0,00	1,07	0,00	0,06	0,00	1,05	0,82	2,61
2,56	0,00	0,25	0,00	0,02	0,00	0,00	0,56	0,13	0,67	0,00	0,00	0,42	0,00	2,56

Tabel 14 Matriks Input Setelah Proses Padding

Matriks Input

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1,31	0,23	0,51	0,00	0,00	0,00	0,00	0,00	0,53	0,12	0,23	0,51	0,00	0,00	1,31
0	1,09	0,79	1,74	0,00	0,00	0,00	0,00	0,00	0,00	0,79	0,00	1,74	0,00	0,00	1,09
0	1,17	0,00	0,56	0,00	0,00	0,00	0,00	0,35	0,76	0,00	0,00	0,56	0,00	0,00	1,17
0	1,16	0,00	0,39	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,39	0,01	0,00	1,16
0	1,40	0,00	0,00	0,00	1,07	0,00	0,00	0,00	0,89	0,00	0,00	0,00	1,07	0,00	1,40
0	1,83	0,00	0,00	0,20	0,68	0,00	0,00	0,00	0,05	0,39	0,00	0,00	0,68	0,00	1,83
0	2,02	0,00	0,00	0,70	0,00	0,00	0,00	0,00	0,00	0,66	0,00	0,00	0,70	0,00	2,02
0	2,18	0,00	0,01	0,51	0,00	0,00	0,00	0,00	0,24	0,99	0,01	0,00	0,51	0,00	2,18
0	1,96	0,00	0,00	0,00	0,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,09	0,00	1,96
0	1,44	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	1,44
0	1,95	0,04	0,00	0,00	0,00	0,00	0,00	0,00	1,09	0,04	0,00	0,00	0,00	0,00	1,95
0	2,22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2,46	0,00	0,00	0,00	0,00	0,00	2,22
0	2,65	0,01	0,71	0,00	0,00	0,00	0,00	0,21	2,33	0,00	0,01	0,71	0,00	0,00	2,65
0	2,61	0,01	0,00	0,00	0,16	0,83	0,00	0,00	1,07	0,00	0,06	0,00	1,05	0,82	2,61
0	2,56	0,00	0,25	0,00	0,02	0,00	0,00	0,56	0,13	0,67	0,00	0,00	0,42	0,00	2,56

Tabel 15 Representasi Filter Ke Dalam Bentuk Matriks Untuk Convolution Layer 3

Input Filter		
-0,0515038200	-0,07426903	0,0421962400
-0,00986491	-0,0377998900	0,1052214500
-0,1181599100	-0,1055901800	0,1092180500

Berikut salah satu contoh perhitungan untuk *Convolution Layer* 3:

G[1,1] = (0 * -0,0515038200) + (0 * -0,07426903) + (0 * 0,0421962400) + (0 * -0,00986491) + (128 * -0,0377998900) + (113 * 0,1052214500) + (0 * -0,1181599100) + (137 *-0,1055901800) + (118 * 0,1092180500) = 0,17

Berikut hasil dari perhitungan matriks keseluruhan:

Tabel 16 Matriks Hasil Perhitungan Setelah Proses Convolution 3

Matriks Input

0,17	-0,15	-0,08	0,00	0,00	0,04	0,02	-0,03	-0,05	0,19	-0,11	-0,08	0,00	0,02	0,17
-0,03	-0,01	0,10	0,00	0,00	0,04	0,07	-0,08	-0,04	0,09	-0,08	0,09	0,00	0,02	-0,03
-0,04	-0,02	0,15	-0,11	-0,08	-0,02	0,06	-0,07	0,00	0,02	0,10	-0,08	-0,08	-0,01	-0,04
-0,12	0,02	0,20	-0,13	-0,10	0,00	0,10	0,00	-0,09	-0,05	0,18	-0,08	-0,10	0,00	-0,12
-0,11	0,10	-0,06	-0,10	0,06	0,00	-0,03	0,10	-0,01	-0,07	0,10	-0,11	0,01	0,00	-0,11
-0,10	0,12	-0,11	-0,06	0,06	0,00	0,02	0,13	-0,15	-0,07	0,10	-0,09	-0,01	0,00	-0,10
-0,07	0,02	-0,02	0,03	-0,01	0,00	0,03	0,07	-0,05	0,01	0,03	-0,04	0,03	0,00	-0,07
-0,01	-0,02	0,01	0,04	0,00	0,00	-0,01	-0,05	0,01	0,09	-0,01	-0,01	0,04	0,00	-0,01
-0,05	0,00	0,00	0,00	0,01	0,00	0,12	-0,11	-0,09	0,00	0,00	0,00	0,01	0,00	-0,05
-0,13	0,00	0,00	0,00	0,00	0,00	0,38	-0,30	-0,23	0,00	0,00	0,00	0,00	0,00	-0,13
-0,05	-0,07	-0,05	0,00	0,00	0,02	0,44	-0,37	-0,19	0,08	-0,08	-0,05	0,00	0,00	-0,05
-0,05	-0,03	-0,01	0,07	-0,10	-0,04	0,25	-0,23	0,03	0,07	0,08	-0,05	-0,16	-0,06	-0,05
-0,08	-0,03	0,06	0,08	-0,04	0,01	-0,04	-0,03	0,08	-0,08	0,15	0,06	-0,11	-0,04	-0,08
-0,02	0,08	-0,03	-0,01	-0,19	0,02	-0,04	0,05	0,01	-0,06	0,00	-0,06	0,11	0,03	-0,02
0,13	-0,29	-0,13	0,27	-0,25	0,04	-0,05	-0,31	-0,14	0,04	0,00	-0,02	0,07	-0,02	0,13

Kemudian hasil dari Tabel tersebut di hitung dengan rumus fungsi aktivasi ReLu, berikut hasil dari matriks tersebut:

$R(z) = \max(0,0.17)$

$R(z) = 0.17$

Tabel 17 Matriks Hasil Perhitungan Setelah Melewati Fungsi Aktifasi ReLu

Matriks Input														
0,17	0,00	0,00	0,00	0,00	0,04	0,02	0,00	0,00	0,19	0,00	0,00	0,00	0,02	0,17
0,00	0,00	0,10	0,00	0,00	0,04	0,07	0,00	0,00	0,09	0,00	0,09	0,00	0,02	0,00
0,00	0,00	0,15	0,00	0,00	0,00	0,06	0,00	0,00	0,02	0,10	0,00	0,00	0,00	0,00
0,00	0,02	0,20	0,00	0,00	0,00	0,10	0,00	0,00	0,00	0,18	0,00	0,00	0,00	0,00
0,00	0,10	0,00	0,00	0,06	0,00	0,00	0,10	0,00	0,00	0,10	0,00	0,01	0,00	0,00
0,00	0,12	0,00	0,00	0,06	0,00	0,02	0,13	0,00	0,00	0,10	0,00	0,00	0,00	0,00
0,00	0,02	0,00	0,03	0,00	0,00	0,03	0,07	0,00	0,01	0,03	0,00	0,03	0,00	0,00
0,00	0,00	0,01	0,04	0,00	0,00	0,00	0,00	0,01	0,09	0,00	0,00	0,04	0,00	0,00
0,00	0,00	0,00	0,00	0,01	0,00	0,12	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,38	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,02	0,44	0,00	0,00	0,08	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,07	0,00	0,00	0,25	0,00	0,03	0,07	0,08	0,00	0,00	0,00	0,00
0,00	0,00	0,06	0,08	0,00	0,01	0,00	0,00	0,08	0,00	0,15	0,06	0,00	0,00	0,00
0,00	0,08	0,00	0,00	0,00	0,02	0,00	0,05	0,01	0,00	0,00	0,00	0,11	0,03	0,00
0,13	0,00	0,00	0,27	0,00	0,04	0,00	0,00	0,00	0,04	0,00	0,00	0,07	0,00	0,13

Berikut perhitungan untuk *Max Pooling layer 3* sebagai berikut:

Tabel 18 Matriks Perhitungan Untuk Proses Max Pooling 3

Filter Input			
0,17	0,00	0,00	0,00
0,00	0,00	0,10	0,00

G[1,1] = max (0.17, 0.00, 0.00, 0.00) = 0.17

G[1,2] = max (0.00, 0.00, 0.10, 0.00,) = 0.10

Sehingga didapatkan hasil keseluruhan matriks sebagai berikut

Tabel 19 Matriks Hasil Perhitungan Setelah Proses Max Pooling 3

Matriks Input											
0,17	0,00	0,00	0,04	0,07	0,19	0,00	0,04	0,15	0,02	0,06	0,09
0,00	0,15	0,00	0,07	0,00	0,10	0,09	0,09	0,09	0,07	0,00	0,00
0,00	0,20	0,06	0,10	0,10	0,18	0,01	0,14	0,11	0,00	0,01	0,00
0,00	0,12	0,06	0,03	0,13	0,10	0,03	0,03	0,19	0,03	0,05	0,00
0,00	0,01	0,04	0,12	0,01	0,09	0,04	0,02	0,13	0,19	0,00	0,13
0,00	0,00	0,00	0,44	0,00	0,08	0,00	0,07	0,16	0,21	0,11	0,00
0,00	0,06	0,08	0,25	0,08	0,15	0,06	0,09	0,03	0,07	0,06	0,14
0,13	0,08	0,27	0,04	0,05	0,04	0,11	0,14	0,14	0,05	0,08	0,00
0,14	0,27	0,12	0,30	0,00	0,03	0,01	0,41	0,21	0,01	0,00	0,00
0,05	0,36	0,05	0,70	0,28	0,05	0,19	0,12	0,00	0,10	0,00	0,00
0,18	0,00	0,27	0,07	0,03	0,05	0,05	0,05	0,21	0,09	0,00	0,00
0,19	0,00	0,00	0,00	0,23	0,00	0,02	0,00	0,00	0,07	0,00	0,00

Tabel 20 Matriks Input Setelah Proses Padding

Matriks Input													
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0,17	0,00	0,00	0,04	0,07	0,19	0,00	0,04	0,15	0,02	0,06	0,09	0,00
0	0,00	0,15	0,00	0,07	0,00	0,10	0,09	0,09	0,09	0,07	0,00	0,00	0,00
0	0,00	0,20	0,06	0,10	0,10	0,18	0,01	0,14	0,11	0,00	0,01	0,00	0,00
0	0,00	0,12	0,06	0,03	0,13	0,10	0,03	0,03	0,19	0,03	0,05	0,00	0,01
0	0,00	0,01	0,04	0,12	0,01	0,09	0,04	0,02	0,13	0,19	0,00	0,13	1,07
0	0,00	0,00	0,00	0,44	0,00	0,08	0,00	0,07	0,16	0,21	0,11	0,00	0,68
0	0,00	0,06	0,08	0,25	0,08	0,15	0,06	0,09	0,03	0,07	0,06	0,14	0,70
0	0,13	0,08	0,27	0,04	0,05	0,04	0,11	0,14	0,14	0,05	0,08	0,00	0,51
0	0,14	0,27	0,12	0,30	0,00	0,03	0,01	0,41	0,21	0,01	0,00	0,00	0,09
0	0,05	0,36	0,05	0,70	0,28	0,05	0,19	0,12	0,00	0,10	0,00	0,00	0,00

0	0,18	0,00	0,27	0,07	0,03	0,05	0,05	0,05	0,21	0,09	0,00	0,00	0,00
0	0,19	0,00	0,00	0,00	0,23	0,00	0,02	0,00	0,00	0,07	0,00	0,00	0,00
0	2,65	0,01	0,71	0,00	0,00	0,00	0,00	0,21	2,33	0,00	0,01	0,71	0,00

Tabel 213 Representasi Filter Ke Dalam Bentuk Matriks Untuk Convolution Layer 4

Filter Input		
-0,1155515800	-0,1155515800	-0,1155515800
-0,0098649100	-0,1271653200	0,1052214500
-0,0377998900	-0,1055901800	-0,0678783000

Berikut salah satu contoh perhitungan untuk *Convolution Layer* 4:

G[1,1] = (0 * -0,1155515800) + (0 *-0,1155515800) + (0 * -0,1155515800) + (0 * -0,0098649100) + (0,17 * -0,1271653200) + (0* 0,1052214500) + (0 * --0,0377998900) + (0*-0,1055901800) + (0,15 * -0,0678783000) = -0,4

Berikut hasil dari perhitungan matriks keseluruhan:

Tabel 22 Matriks Hasil Perhitungan Setelah Proses Convolution 4

Matriks Input													
0,00	-0,06	0,00	-0,06	-0,01	-0,02	0,02	-0,04	-0,02	-0,02	0,01	0,01	0,00	0,00
-0,03	-0,05	-0,08	-0,11	-0,07	-0,02	-0,04	-0,07	-0,06	-0,02	-0,01	0,01	0,01	0,00
-0,05	0,00	-0,04	-0,03	-0,05	-0,05	-0,03	-0,01	-0,04	0,00	-0,02	-0,03	0,02	0,02
-0,04	0,00	0,01	-0,01	-0,02	-0,01	-0,01	-0,02	-0,01	-0,01	-0,01	-0,03	-0,03	0,01
-0,02	-0,04	0,00	-0,03	-0,02	-0,02	-0,01	-0,05	-0,01	0,01	0,01	-0,01	-0,05	0,00
-0,02	-0,05	-0,03	-0,01	-0,04	-0,03	-0,01	-0,04	-0,04	0,02	0,00	0,01	-0,01	-0,01
-0,01	-0,02	-0,05	0,02	0,00	-0,02	-0,01	-0,03	-0,04	-0,01	0,02	-0,01	0,00	0,00
0,00	0,00	-0,03	-0,05	0,01	-0,01	0,00	-0,01	-0,05	-0,05	-0,02	0,00	0,00	0,00
0,01	-0,01	-0,01	-0,07	-0,02	-0,03	-0,01	-0,02	-0,02	-0,03	-0,01	-0,03	0,01	0,01
-0,03	0,00	-0,05	0,00	0,00	-0,03	-0,06	-0,01	0,02	0,00	-0,02	-0,04	-0,02	0,03
-0,07	-0,03	-0,07	0,01	0,07	0,00	-0,02	-0,06	-0,05	0,00	0,00	-0,02	-0,04	-0,01
-0,03	-0,09	-0,04	-0,11	-0,04	-0,01	-0,05	-0,08	-0,01	0,01	0,01	-0,02	-0,03	-0,01
-0,03	-0,04	-0,08	-0,13	-0,02	-0,01	-0,04	-0,03	-0,04	-0,02	0,01	-0,01	-0,01	0,01
-0,05	-0,02	-0,04	-0,02	-0,04	-0,01	-0,01	-0,02	-0,03	-0,02	0,00	-0,01	-0,02	-0,01

Kemudian hasil dari Tabel tersebut di hitung dengan rumus fungsi aktivasi ReLu, berikut hasil dari matriks tersebut:

R(z) = max(-0,4)

R(z) = -0,4

Tabel 23 Matriks Hasil Perhitungan Setelah Melewati Fungsi Aktifasi ReLu

Matriks Input													
0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,01	0,01	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,02
0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,01	0,00	0,00
0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,03
0,00	0,00	0,00	0,01	0,07	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Berikut perhitungan untuk *Max Pooling layer* 4 sebagai berikut:

Tabel 24 Matriks Perhitungan Untuk Proses Max Pooling 4

Filter Input			
0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00

G[1,1] = max (0.00, 0.00, 0.00, 0.00) = 0.00

G[1,2] = max (0.00, 0.00, 0.00, 0.00,) = 0.00

Sehingga didapatkan hasil keseluruhan matriks sebagai berikut

Tabel 25 Matriks Hasil Perhitungan Setelah Proses Max Pooling 4

Matriks Input													
0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,01	0,01	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,02
0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00

0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,01	0,00	0,00
0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,03
0,00	0,00	0,00	0,01	0,07	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00	0,00	0,01
0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00