

**ANALISIS KLASIFIKASI TAHU YANG MENGANDUNG
FORMALIN MENGGUNAKAN MODEL ARSITEKTUR
*GOOGLINET***

SKRIPSI

AINUN FADHILLAH

198160034



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MEDAN AREA
2024**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 13/12/24

Access From (repository.uma.ac.id)13/12/24

**ANALISIS KLASIFIKASI TAHU YANG MENGANDUNG FORMALIN
MENGUNAKAN MODEL ARSITEKTUR *GOOGLNET***

SKRIPSI

Diajukan Sebagai salah satu syarat untuk memperoleh

Gelar Sarjana di Fakultas Teknik

Universitas Medan Area



Oleh :

AINUN FADHILLAH

198160034

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS MEDAN AREA

2024

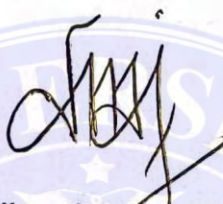
UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

LEMBAR PENGESAHAN

Judul Skripsi : Analisis Klasifikasi Tahu Yang Mengandung Formalin Menggunakan Model Arsitektur *GoogLeNet*
Nama : Ainun Fadhillah
NPM : 198160034
Fakultas : Teknik Informatika

Disetujui Oleh
Komisi Pembimbing



Susilawati, S.Kom, M.Kom
Pembimbing

Diketahui :



Tanggal Lulus : Agustus 2024


HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan,2024




Ainun Fadhillah
198160034


HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Ainun Fadhillah
NPM : 198160034
Program Studi : Teknik Informatika
Fakultas : Teknik
Jenis karya : Tugas Akhir/Skripsi/Tesis

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Non eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul: "**Analisis Klasifikasi Tahu Yang Mengandung Formalin Menggunakan Model Arsitektur *GoogLeNet***". Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non eksklusif ini Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir/skripsi/tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan
Pada tanggal : 2024
Yang menyatakan


Ainun Fadhillah
198160034

RIWAYAT HIDUP

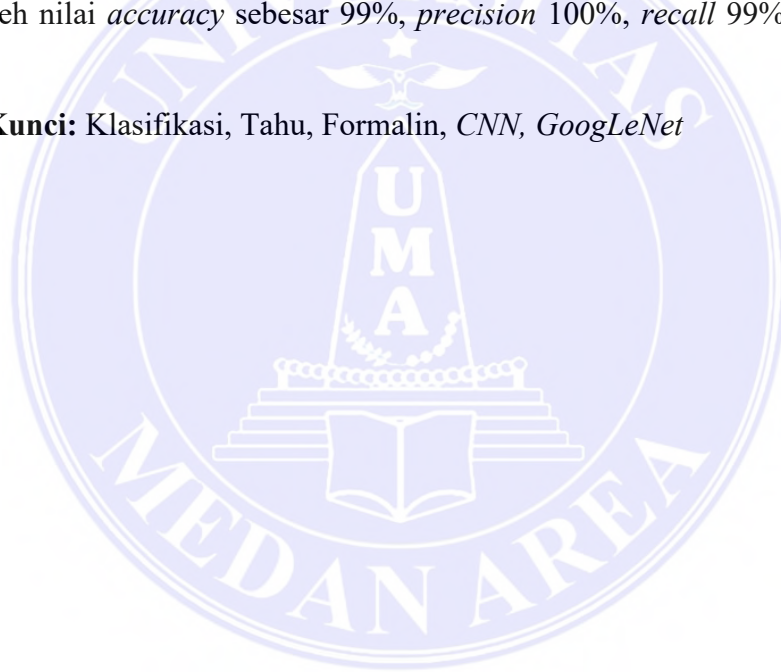
Penulis dilahirkan di Panyabungan Pada tanggal 05 Juni 2001 dari ayah Umri Nur dan ibu Sri Suryati. Penulis merupakan anak kelima dari lima bersaudara. Penulis pertama kali mengenyam pendidikan di bangku SD Negeri 3 Gunung Tua pada tahun 2007 dan lulus pada tahun 2013. Kemudian penulis melanjutkan pendidikan ke jenjang SMP pada tahun 2013 di MTs Negeri 2 Padang Lawas Utara dan lulus pada tahun 2016. Pada tahun yang sama, penulis melanjutkan ke jenjang selanjutnya yaitu di SMA Negeri 2 Sipirok dan pada bulan September tahun 2019, penulis melanjutkan pendidikan di bangku kuliah dan terdaftar sebagai mahasiswa Fakultas Teknik dengan Program Studi Teknik Informatika di Universitas Medan Area.



ABSTRAK

Tahu merupakan produk pangan yang menggunakan bahan dasar kacang kedelai yang diendapkan. Proses pembuatan tahu biasanya membutuhkan modal sedikit yang membuat para pelaku usaha melakukan kecurangan dengan mencampurkan zat berbahaya salah satunya adalah formalin ke dalam tahu. Hal ini berdampak pada Kesehatan. Apabila mengkonsumsi formalin pada dosis rendah dapat menyebabkan gangguan pencernaan disertai dengan muntah-muntah, timbulnya depresi, dan peredaran darah tidak lancar. Oleh sebab itu, diperlukan pendekatan digital agar dapat membantu untuk mengenali tahu tersebut mengandung formalin atau tidak. Analisis yang dilakukan dalam penelitian ini menggunakan model arsitektur *GoogLeNet* dari *CNN*. Terdapat 8 model skenario yang diuji (*training*) dan diperoleh performa model terbaik pada model menggunakan *hyperparameter* dengan jumlah *epoch* 30, *batch size* 32, *optimizer* *SGD*, *learning rate* 0.001 mendapat akurasi 99% pada proses *training*. Setelah diuji menggunakan data testing dan dievaluasi menggunakan *confusion matrix* dan *classification report* diperoleh nilai *accuracy* sebesar 99%, *precision* 100%, *recall* 99%, dan *f1-score* 99%.

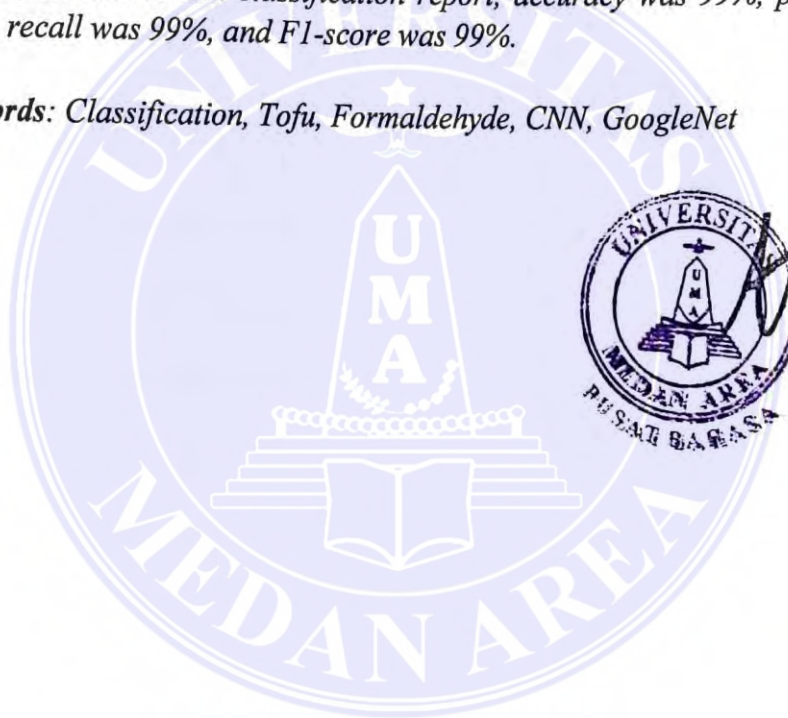
Kata Kunci: Klasifikasi, Tahu, Formalin, *CNN*, *GoogLeNet*



ABSTRACT

Tofu is a food product made from precipitated soybeans. The tofu-making process generally requires low capital, which leads some business practitioners to commit fraud by adding harmful substances, such as formaldehyde, to tofu. This has a detrimental effect on health. Consuming formaldehyde at low doses can cause digestive issues accompanied by vomiting, depression, and poor blood circulation. Therefore, a digital approach is needed to help identify whether tofu contains formaldehyde. The analysis conducted in this research used the GoogleNet architecture model from CNN. Eight model scenarios were tested (training), and the best model performance was achieved using hyperparameters with 30 epochs, a batch size of 32, the SGD optimizer, and a learning rate of 0.001, achieving 99% accuracy in the training process. After testing with test data and evaluating using a confusion matrix and classification report, accuracy was 99%, precision was 100%, recall was 99%, and F1-score was 99%.

Keywords: *Classification, Tofu, Formaldehyde, CNN, GoogleNet*



KATA PENGANTAR

Puji dan syukur penulis haturkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya-lah saya dapat menyelesaikan skripsi yang berjudul “Analisis Klasifikasi Tahu Yang Mengandung Formalin Menggunakan Model Arsitektur *GoogLeNet*”. Skripsi ini dibuat untuk memenuhi tugas akhir perkuliahan dan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Strata 1 di Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area. Selain itu, skripsi ini juga dibuat sebagai salah satu wujud implementasi dari ilmu yang didapatkan selama masa perkuliahan di Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.

Skripsi ini tentunya tidak lepas dari bimbingan, masukan, dan arahan dari berbagai pihak. Oleh karena itu, pada kesempatan ini saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Eng. Supriatno, ST., M.T. sebagai Dekan Fakultas Teknik Universitas Medan Area.
2. Bapak Rizki Muliono, S.Kom, M.Kom. sebagai Ketua Program Studi Teknik Informatika yang telah memberikan dukungan selama masa perkuliahan di Teknik Informatika Universitas Medan Area.
3. Ibu Susilawati, S.Kom., M.Kom. sebagai dosen pembimbing utama yang telah meluangkan waktu memberikan dukungan, bimbingan, dan motivasi dalam menyelesaikan skripsi ini.
4. Seluruh Dosen dan Staf Program Studi Teknik Informatika Universitas Medan Area.

5. Kedua orang tua penulis, Umri Nur dan Sri Suryati, untuk beliau berdua lah skripsi ini saya persembahkan. Terima kasih atas segala kasih sayang yang diberikan dalam membesarkan dan membimbing penulis selama ini sehingga penulis dapat terus berjuang dalam meraih mimpi dan cita-cita. Kesuksesan dan segala hal baik yang kedepannya akan penulis dapatkan adalah karena dan untuk kalian berdua.
6. Serta saudara-saudara penulis, kak Rara Dwi Kartika, bang Rinaldi Syaputra, kak Ranggi Tria, kak Ringga Nabila, dan kak Yuliarni Zebua selaku kakak ipar penulis, terima kasih selalu percaya pada pilihan dan mimpi-mimpi penulis, terima kasih atas semangat yang tiada henti kalian berikan, kalian adalah yang terbaik.
7. Teman-teman seperjuangan penulis dari semester satu, Annisa Nst., Deanisa Luthfy Casenda, dan Nurul Almadinah Pane, terima kasih atas kenangan indah bersama kalian selama ini, terima kasih atas semangat dan motivasi, serta suka duka yang kita rasakan bersama selama menjalani perkuliahan, penelitian, hingga penyusunan skripsi ini. Semoga hubungan pertemanan yang kita jalin tetap terjaga.
8. Teman dan sahabat penulis sejak SMA hingga sekarang, para Idiots tercinta Annisa Amaliah Nasution, Fadina Garora Harahap, Novita Hutabalian, dan Sakinah, terima kasih kepada kalian yang telah memberikan dukungan dan mengajak penulis untuk healing melepaskan penat selama proses pengerjaan skripsi ini. Ada banyak mimpi yang ingin kita gapai bersama, semoga di masa depan mimpi itu bisa kita gapai, kalian yang terbaik.

9. Seluruh teman-teman yang sudah memberikan dukungannya selama penulisan proposal skripsi ini, khususnya teman-teman Teknik Informatika angkatan 2019.
10. Seluruh pihak yang tidak dapat disebutkan satu persatu yang membantu dalam penyelesaian skripsi ini.

Sebagai manusia biasa penulis menyadari penyusunan skripsi ini jauh dari kata sempurna karena keterbatasan kemampuan dan ilmu pengetahuan yang dimiliki oleh penulis. Oleh karena itu, atas kesalahan dan kekurangan dalam penulisan skripsi ini, penulis memohon maaf dan bersedia menerima kritikan yang membangun.

Terakhir harapan penulis, Semoga skripsi ini dapat memberikan manfaat bagi siapa saja yang membacanya, aamiin.

Medan, 2024

Ainun Fadhillah



Penulis,

DAFTAR ISI

LEMBAR PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS	v
RIWAYAT HIDUP	vi
ABSTRAK	vii
ABSTRACT	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	6
1.6 Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Tahu.....	8
2.1.1 Tahu Tanpa Formalin	8
2.1.2 Tahu Formalin	9
2.2 Formalin	10
2.3 Citra Digital.....	10
2.4 <i>Deep Learning</i>	12
2.5 <i>Convolutional Neural Network (CNN)</i>	12
2.5.1 <i>Convolution Layer</i>	13
2.5.2 <i>Rectification Linear Unit (ReLU)</i>	14
2.5.3 <i>Pooling Layer</i>	14
2.5.4 <i>Fully connected Layer</i>	14

2.5.5 <i>Softmax</i>	15
2.6 <i>GoogLeNet</i>	15
2.7 Penelitian Terkait	17
BAB III METODOLOGI PENELITIAN	20
3.1 Diagram Alur Penelitian.....	20
3.2 Identifikasi Masalah	21
3.3 Study Literatur.....	21
3.4 Pengumpulan Data	21
3.5 Analisis Data	22
3.5.1 <i>Cropping</i> Data	23
3.5.2 <i>Resizing</i> Data.....	23
3.5.3 <i>Labeling</i> Data	24
3.5.4 Augmentasi Data	24
3.5.5 <i>Modeling GoogLeNet</i>	26
3.6 Evaluasi	36
BAB IV HASIL DAN PEMBAHASAN	39
4.1 Hasil.....	39
4.2 Pembahasan	43
4.2.1 <i>Training</i> Model	43
4.2.2 Evaluasi Model.....	48
BAB V KESIMPULAN DAN SARAN	57
5.1 Kesimpulan.....	57
5.2 Saran	57
DAFTAR PUSTAKA	58
LAMPIRAN.....	62

DAFTAR TABEL

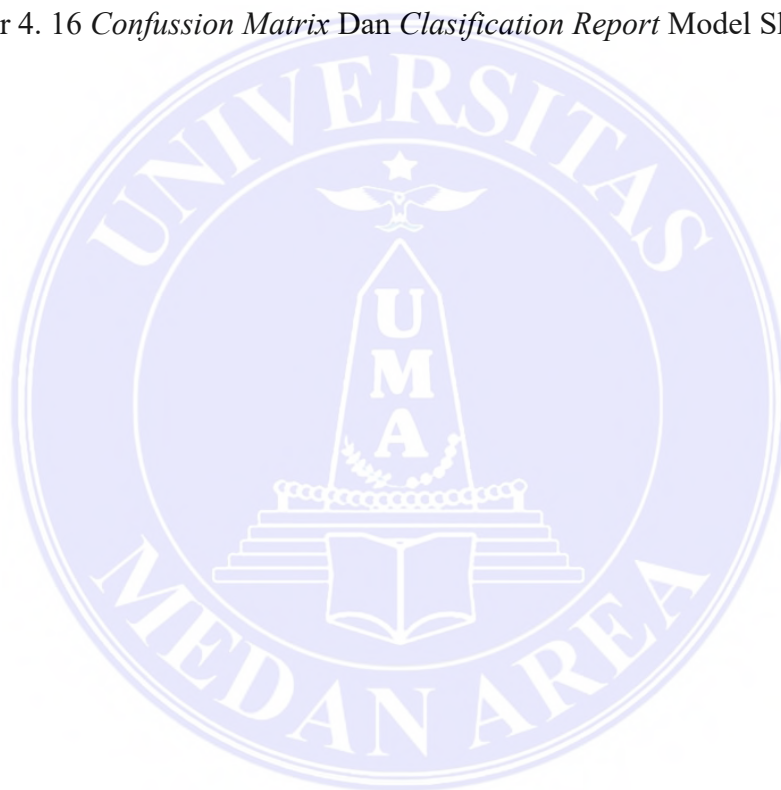
Tabel 2. 1 Ringkasan Penelitian Terkait	17
Tabel 3. 1 Deskripsi Citra dan Jumlah Data Citra	22
Tabel 4. 1 Perbandingan hasil inisialisasi <i>hyperparameter</i> dari setiap skenario ..	39



DAFTAR GAMBAR

Gambar 2. 1 Tahu Tanpa Formalin	9
Gambar 2. 2 Tahu Mengandung Formalin.....	9
Gambar 2. 3 Formalin	10
Gambar 2. 4 Representasi Citra Digital (Gifran et al., 2019)	11
Gambar 2. 5 Arsitektur <i>CNN</i> (Faizin et al., 2022)	13
Gambar 2. 6 Proses Konvolusi (Trisiawan et al., 2022)	13
Gambar 2. 7 Proses <i>Max Pooling</i> (Peryanto et al., 2020).....	14
Gambar 2. 8 Fully Connected Layer (Asrianda et al., 2021).....	15
Gambar 3. 1 Alur Penelitian Secara Umum.....	20
Gambar 3. 2 <i>Cropping</i> Data Gambar	23
Gambar 3. 3 <i>Risizing</i> Data Gambar.....	23
Gambar 3. 4 Proses <i>Labeling</i> Data.....	24
Gambar 3. 5 Hasil Augmentasi Tahu Formalin	25
Gambar 3. 6 Hasil Augmentasi Data Nonformalin.....	26
Gambar 3. 7 Arsitektur <i>Googlenet Inception V3</i>	26
Gambar 3. 8 <i>Stem Block</i>	27
Gambar 3. 9 <i>Inception A</i>	27
Gambar 3. 10 <i>Inception B</i>	27
Gambar 3. 11 <i>Inception C</i>	28
Gambar 3. 12 <i>Reduction A</i>	28
Gambar 3. 13 <i>Reduction B</i>	29
Gambar 3. 14 Nilai <i>Pixel RGB</i> Ukuran 7x7	29
Gambar 3. 15 Kernel 3x3	30
Gambar 3. 16 Ilustrasi <i>Fully Connected Layer</i> Dan <i>Softmax</i>	35
Gambar 3. 17 Model <i>Confusion Matrix</i>	37
Gambar 4. 1 Grafik Akurasi Dan <i>Loss</i> Model Skenario 1	43
Gambar 4. 2 Grafik Akurasi Dan <i>Loss</i> Model Skenario 2	44
Gambar 4. 3 Grafik Akurasi Dan <i>Loss</i> Model Skenario 3	45
Gambar 4. 4 Grafik Akurasi Dan <i>Loss</i> Model Skenario 4	45
Gambar 4. 5 Grafik Akurasi Dan <i>Loss</i> Model Skenario 5	46
Gambar 4. 6 Grafik Akurasi Dan <i>Loss</i> Model Skenario 6	47

Gambar 4. 7 Grafik Akurasi Dan <i>Loss</i> Model Skenario 7	47
Gambar 4. 8 Grafik Akurasi Dan <i>Loss</i> Model Skenario 8	48
Gambar 4. 9 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 1	49
Gambar 4. 10 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 2	50
Gambar 4. 11 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 3	51
Gambar 4. 12 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 4	52
Gambar 4. 13 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 5	53
Gambar 4. 14 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 6	54
Gambar 4. 15 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 7	55
Gambar 4. 16 <i>Confussion Matrix</i> Dan <i>Clasification Report</i> Model Skenario 8	56



BAB I

PENDAHULUAN

1.1 Latar Belakang

Makanan diperlukan untuk keberlangsungan hidup manusia, yang artinya tanpa makanan tubuh akan lemas dan mudah terserang penyakit (Dedy Kasingku & Lumoindong, 2023). Makanan merupakan salah satu kebutuhan primer yang sangat dibutuhkan untuk dikonsumsi karena memiliki peran penting dalam mencukupi nutrisi bagi tubuh manusia. Indonesia sendiri memiliki tanah yang subur, sehingga masyarakat memanfaatkan karunia tersebut untuk menanam berbagai tumbuhan baik tumbuhan yang tidak bisa dimakan atau tumbuhan yang bisa dimakan seperti sayuran, kacang-kacangan, buah-buahan, dan lain sebagainya. Dengan hasil panen ini mereka bisa jual atau mengolahnya langsung.

Makanan olahan atau produk pangan buatan yang sampai sekarang digemari berbagai kalangan masyarakat adalah tahu (Rottie, 2020), karena memiliki rasa yang enak, lembut, murah dan mudah didapat. Tahu merupakan produk pangan yang diolah menggunakan bahan dasar kacang kedelai yang diendapkan. Tahu mengandung asam amino yang lengkap serta terkandung protein nabati didalamnya, sehingga tahu dijadikan sebagai bahan pangan yang bagus untuk memperbaiki gizi (Admojo & Sulistya, 2022).

Menurut (Gradian Landang et al., 2023) mengatakan bahwa ada banyak sekali makanan pokok yang bisa dikonsumsi, tetapi ada sekitar 38% kedelai dikonsumsi dalam bentuk produk tahu di Indonesia. Dalam proses pembuatan tahu juga tidak memerlukan modal yang banyak, sehingga tidak sedikit pelaku usaha yang membangun usaha tahu baik dalam skala rumahan atau sudah sampai skala

yang besar (Pardede et al., 2022). Tidak sedikit juga pelaku usaha melakukan kecurangan dengan mencampurkan zat berbahaya dalam pembuatan tahu. hal ini dilakukan karena memproduksi tahu tidak memerlukan biaya yang besar, minimnya kerugian tanpa memikirkan orang yang mengkonsumsinya (Rosita et al., 2019). Zat berbahaya dicampurkan kedalam makanan agar dapat mencegah tumbuhnya mikroba perusak makanan, terlihat lebih menarik, dan menghemat biaya produksi (Setiawan, 2020). Umur simpan tahu juga sangat terbatas, dan cepat mengalami kerusakan yang digolongkan ke dalam *high perishable food*.

Pada temperatur normal atau suhu ruangan tahu dapat bertahan kurang lebih dua, apabila tahu disimpan kelamaan rasa tahu akan berubah menjadi asam dan akan mengeluarkan bau yang tidak sedap sehingga tidak layak untuk dikonsumsi (Waryat et al., 2020). Apabila tahu disimpan kelamaan rasa tahu akan berubah menjadi asam dan tidak layak untuk dikonsumsi. Hal ini membuat penjual mencoba beberapa trik agar tahu yang dijual tahan lama. Salah satu trik nya adalah dengan mencampurkan pengawet.

Ada banyak sekali jenis pengawet yang dicampurkan ke dalam bahan makanan, salah satunya adalah formalin. Pada dasarnya formalin tidak diperbolehkan digunakan sebagai pengawet makanan, tetapi sering dijadikan pengawet makanan karena harga nya yang murah, mudah didapat dan dapat meminimalkan biaya produksi (Asyfiradayati et al., 2019). Bahan kimia ini berfungsi sebagai desinfektan dan termasuk dalam golongan desinfektan kuat. Formalin yang ditambahkan pada makanan seperti tahu akan mengikat unsur protein, sehingga jika tahu ditekan akan terasa kenyal dan tahu tersebut tidak akan diserang oleh bakteri pembusuk yang artinya tahu akan lebih awet (Arziyah et al.,

2019). Dampak yang diterima oleh kesehatan apabila mengkonsumsi formalin tanpa sengaja adalah adanya gangguan pencernaan disertai dengan muntah-muntah, timbulnya depresi, dan peredaran darah tidak lancar (Restiani et al., 2020). Terdapat beberapa penelitian tentang kasus tahu formalin, dimana tahu yang dijual di pasar memiliki persentase tinggi akan formalin. Seperti pada penelitian (Hayat & Darusmin, 2022), disebutkan bahwa terdapat 29 (67,4%) dari 43 sampel tahu yang mengandung formalin.

Di era perkembangan teknologi pengolahan citra digital, solusi cerdas berbasis kecerdasan buatan (*AI*) dan *deep learning* telah menjadi terobosan penting dalam pengklasifikasian produk, termasuk deteksi tahu yang mengandung formalin. Penggunaan *image processing* dalam pengolahan citra digital memungkinkan dapat mengenali karakteristik atau ciri tertentu dari objek yang dianalisis, seperti tahu yang mengandung formalin atau tidak, secara akurat dan efisien.

Penelitian ini dilakukan untuk mencoba menerapkan metode *Convolutional Neural Network (CNN)* dalam pengolahan data citra. *Convolutional Neural Network (CNN)* merupakan salah satu metode yang dimiliki oleh *Deep Learning* yang dikembangkan berdasarkan *MLP (Multy Layer Perceptron)* (Asrianda et al., 2021). Metode *Convolutional Neural Network (CNN)* mempunyai beberapa jenis model arsitektur salah satunya adalah *GoogLeNet*. *GoogLeNet* adalah struktur dari *Deep Learning network* yang diusulkan oleh *Google*. Model ini dipilih karena memiliki akurasi yang tinggi dan telah memenangkan kompetisi klasifikasi citra (*ILSVRC*) pada tahun 2014 (Yu et al., 2020). *GoogLeNet* merupakan salah satu model *Deep Learning* yang memperkenalkan pendekatan yang disebut "*Inception module*" yang dapat membuat proses konvolusi dan *pooling* berjalan secara paralel,

dan memperpendek waktu komputasi tanpa mengurasi nilai akurasi sistem (Sa'idah et al., 2022). *GoogLeNet* memiliki lapisan konvolusi, lapisan *maxpooling*, dan lapisan *Inception*, lalu fitur dari *image* akan diekstrak dan akan diklasifikasi menggunakan *softmax* (Yuan & Kan, 2022).

Model arsitektur *GoogLeNet* ini memiliki 4 versi *Inception*, yaitu *Inception* penggunaan bahan tambahan makanan berbahaya boraks dan *Inception-V1*, *Inception-V2*, *Inception-V3*, dan *Inception-V4*. Penelitian ini menggunakan *Inception-V3* karena memiliki kinerja yang lebih unggul dalam pengenalan objek. Seperti penelitian yang dilakukan oleh (Pan et al., 2023) menghasilkan akurasi 93.81% dalam klasifikasi citra penyakit fundus pada mata dengan menggunakan 1.032 citra fundus. Klasifikasi kematangan buah kopi kuning berdasarkan warna yang dilakukan oleh (Ungkawa & Galih Al, 2023) mendapatkan akurasi sebesar 92%. Selanjutnya adanya penelitian yang dilakukan oleh (Mujahid et al., 2022) menghasilkan akurasi 99.29% melakukan klasifikasi terhadap citra *X-ray pneumonia*. Klasifikasi kanker paru-paru menggunakan model arsitektur *GoogLeNet* mendapatkan akurasi sebesar 99.05% oleh (Almas et al, 2019). Melakukan klasifikasi Diabetic Retinopathy pada citra retina fundus dengan menggunakan dataset sebanyak 2.500 data citra dan mendapatkan akurasi sebesar 84.16% oleh (Faizin et al., 2022)

Berdasarkan capaian dari penelitian terdahulu model arsitektur *GoogLeNet* ini dapat mengidentifikasi data citra dengan baik. Oleh karena itu, peneliti mencoba mengidentifikasikan tahu yang mengandung formalin menggunakan model arsitektur *GoogLeNet* berdasarkan tekstur dan warna. Diharapkan dapat menghasilkan akurasi terbaik dalam mengidentifikasi citra tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan sebelumnya, maka rumusan masalah pada penelitian ini adalah bagaimana mengidentifikasi tahu yang mengandung formalin berdasarkan warna dan tekstur menggunakan model arsitektur *GoogLeNet*.

1.3 Tujuan Penelitian

Adapun yang menjadi tujuan pada penelitian ini adalah untuk mengidentifikasi tahu yang mengandung formalin menggunakan model arsitektur *GoogLeNet* berdasarkan warna dan tekstur tahu dari data citra yang telah diperoleh.

1.4 Batasan Masalah

Batasan masalah yang diterapkan pada penelitian ini, yaitu sebagai berikut:

1. Data yang digunakan pada penelitian ini adalah data primer yang diambil secara langsung dengan jumlah data 2.400 *dataset* citra tahu dengan format *.jpg*.
2. Pada penelitian ini citra tahu yang digunakan adalah tahu formalin dan tahu tanpa formalin.
3. Citra yang diinput adalah tekstur dan warna tahu.
4. Pengujian model arsitektur *GoogLeNet* menggunakan 8 model skenario dengan *input shape* (299 x 299), jumlah *epoch* (20 dan 30), *batch size* (32 dan 64), *learning rate* (0.001), serta *batch size* (*Adam*, *RMSprop*, *SGD*, dan *Nadam*) dijadikan statis pada setiap skenario yang diuji.
5. Pengujian akurasi dilakukan menggunakan *Confusion Matrix* dan *classification report* untuk menghitung tingkat akurasi yang didapatkan dari

masing-masing skenario model, sehingga dapat ditentukan performa model terbaik.

1.5 Manfaat Penelitian

Manfaat penelitian ini dilakukan adalah untuk memberikan kontribusi pada bidang pengolahan citra digital dan *deep learning* dalam hal mengidentifikasi tahu yang mengandung formalin atau tahu yang tidak mengandung formalin.

1.6 Sistematika Penulisan

Sistematika penulisan pada skripsi memaparkan alur penulisan skripsi yang dapat mempermudah dalam memahami isi laporan dari awal sampai dengan akhir. Penulis membagi sistematika penulisan dalam lima bab. Berikut penjelasan dari masing-masing bab:

1. BAB I PENDAHULUAN

Bab satu berisi pendahuluan atau materi yang dapat menyempurnakan usulan dari suatu penelitian yang terdiri dari: latar belakang, rumusan masalah, Batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Pada bab dua ini membahas tentang study literatur atau teori yang ditemukan sebagai acuan dalam penelitian, dan deskripsi objek yang digunakan.

3. BAB III METODE PENELITIAN

Bab tiga memaparkan diagram penelitian mulai dari awan sampai akhir, menjelaskan bagaimana proses pengambilan data, tahap-tahap pembuatan perancangan model *GoogLeNet* serta *hyperparameter* apa yang digunakan.

4. BAB IV HASIL DAN PEMBAHASAN

Berisi hasil dan pembahasan dari keseluruhan penelitian yang dilakukan dalam waktu pengujian menggunakan model *GoogLeNet* pada tahu formalin

5. BAB V KESIMPULAN DAN SARAN

Bab ini menjelaskan isi kesimpulan dari keseluruhan penelitian dan terdapat beberapa saran untuk kekurangan pada penelitian ini.



BAB II

TINJAUAN PUSTAKA

2.1 Tahu

Tahu merupakan makanan tradisional asal Tiongkok yang sudah banyak dikonsumsi oleh masyarakat Indonesia karena harganya yang terjangkau oleh hampir semua lapisan masyarakat (Herdhiansyah et al., 2022). Tahu adalah suatu produk pangan padatan lunak yang diperoleh dengan mengolah kedelai dengan pengendapan protein, dengan atau tanpa penambahan bahan lain yang diperbolehkan. Pembuatan tahu dilakukan dengan cara merendam kacang kedelai selama 8-12 jam apabila menggunakan air dingin, dan 1-2 jam apabila menggunakan air panas. Lalu dilakukan proses pembersihan berkali-kali hingga bersih. Proses pembuatan tahu selanjutnya adalah kedelai yang sudah dicuci akan digiling sampai halus dan dimasukan larutan asam dan ditunggu sampai ada muncul kedelai halus yang mengendap. Terakhir dilakukan penyaringan untuk memisahkan air dan kedelai yang sudah diendapkan menggunakan batu tahu. Pada tahapan pembuatan tahu terdapat suatu cairan atau larutan asam yang dicampurkan, larutan asam yang digunakan biasanya campuran asam cuka dan perasan lemon.

Larutan asam ini digunakan untuk mengendapkan kedelai dan mengumpulkan protein tahu sehingga menghasilkan tahu yang lembut dan halus. Tahu yang baik dilihat dari teksturnya yaitu halus, terlihat tidak keras dan kenyal (Andarwulan et al., 2019). Tahu sering disebut dengan daging tanpa tulang karena memiliki kandungan gizi yang tinggi terutama mutu protein yang setara dengan daging hewan serta mengandung senyawa bioaktif yang baik untuk kesehatan (Barus et al., 2019).

2.1.1 Tahu Tanpa Formalin

Tahu yang baik atau tanpa campuran formalin memiliki tekstur yang halus dan lembut, terlihat tidak keras dan kenyal. Tahu yang baik memiliki kandungan protein yang

tinggi dan kadar air yang mencapai 85% (Saiman & Satra, 2021) , hal ini yang membuat tahu tidak dapat bertahan lama. Kerusakan pada produk pangan ini ditandai dengan adanya bau asam, rasa tahu juga menjadi asam dan berlendir. Umumnya umur simpan tahu ini kurang lebih hanya dua hari, lebih dari hari tersebut rasa tahu tidak layak lagi untuk dikonsumsi.



Gambar 2. 1 Tahu tanpa formalin

2.1.2 Tahu Formalin

Para produsen tahu menggunakan formalin pada tahu agar tahu lebih tahan lama. Praktek yang dilakukan adalah merendam tahu dengan formalin (Fitriani Sammulia et al., 2020). Penambahan zat berbahaya ini dapat merubah tekstur menjadi lebih kasar dan jika dilihat tidak gampang rusak atau pecah dan bahkan dapat merubah warna tahu menjadi lebih menarik yaitu lebih putih dibanding tahu tanpa formalin.

Umur simpan tahu yang mengandung formalin adalah kurang lebih satu minggu (tujuh hari). Beda dengan tahu tanpa formalin, tahu yang mengandung formalin dapat mempertahankan rasa, tidak berlendir, dan terlihat masih segar.



Gambar 2. 2 Tahu mengandung formalin

2.2 Formalin

Formalin atau yang dikenal dengan bahasa dagangnya *formaldehid* merupakan zat kimia berbahaya bagi manusia sehingga dilarang digunakan sebagai bahan baku pada pembuatan makanan seperti pembuatan mie basah, lontong, bakso, sosis, tahu, bahkan dalam pembuatan kecap pun masih menggunakan bahan formalin (Dewi, 2019). Penggunaan bahan ini dimaksudkan agar makanan yang dijual bisa disimpan dalam jangka waktu yang lama dan tidak mudah rusak. Formalin merupakan bahan tambahan pangan yang tidak diizinkan digunakan di Indonesia, hal ini dikarenakan formalin merupakan pengawet yang bersifat karsinogenik dan biasanya digunakan untuk mengawetkan mayat sehingga apabila mengkonsumsi formalin dengan terus-menerus dapat menimbulkan gangguan Kesehatan atau dapat memicu kanker (Berliana et al., 2021).

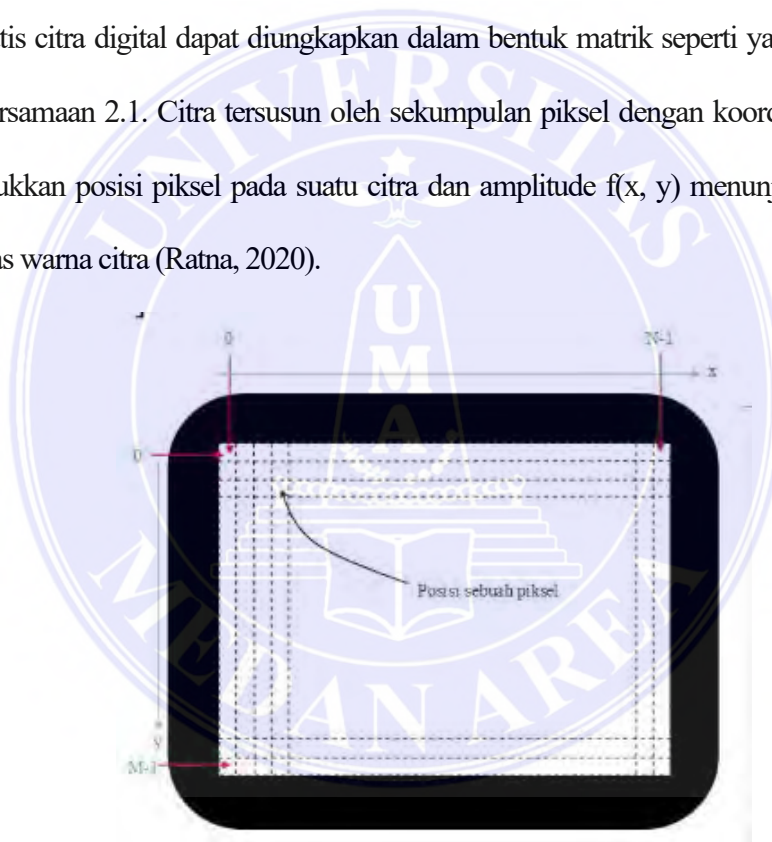


Gambar 2. 3 Formalin

2.3 Citra Digital

Citra Digital adalah merupakan suatu gambar dua dimensi yang dapat ditampilkan sebagai nilai digital. Citra digital dapat diperoleh dari proses *sampling* spasial dan kuantisasi intensitas. Sampling spasial menentukan resolusi dari citra sedangkan kuantisasi intensitas menentukan kedalaman piksel. Citra digital dibentuk dengan kumpulan titik yang dinamakan piksel. Pada setiap piksel dapat diimplementasikan sebagai satu kotak kecil dimana piksel memiliki titik koordinat yang posisi dan nilainya disebut dengan intensitas.

Masing-masing piksel memiliki nilai derajat keabuannya sendiri. Setiap kumpulan piksel tersebut disimpan dalam komputer dalam bentuk array dua dimensi(matriks) dengan ukuran $M \times N$ piksel, dimana nilai variabel M mempresentasikan jumlah piksel baris dan untuk nilai variabel N adalah jumlah piksel untuk kolom pada citra digital, yang secara umum $m = 0, 1, \dots, M-1$ dan $n = 0, 1, \dots, N-1$. Di setiap posisi (m, n) terletak detail selanjutnya yang disebut dengan unsur gambar atau picture element yang disingkat menjadi pel, pixel atau piksel (Gifran et al., 2019). Representasi citra digital diperlihatkan pada gambar 2.4. Secara matematis citra digital dapat diungkapkan dalam bentuk matrik seperti yang diperlihatkan pada persamaan 2.1. Citra tersusun oleh sekumpulan piksel dengan koordinat (x, y) yang menunjukkan posisi piksel pada suatu citra dan amplitude $f(x, y)$ menunjukkan nilai dari intensitas warna citra (Ratna, 2020).



Gambar 2. 4 Representasi Citra Digital (Gifran et al., 2019)

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & f(0,0) & \dots & f(1, M - 1) \\ \vdots & \vdots & \dots & \vdots \\ f(N - 1,0) & f(N - 1,1) & \dots & f(N - 1, M - 1) \end{bmatrix} \dots\dots\dots (2.1)$$

Resolusi citra menunjukkan jumlah baris atau kolom elemen citra (piksel) per satuan pengukuran, misalnya *dot per inch (dpi)* yang menyatakan banyaknya piksel setiap jarak per

inci. Resolusi citra juga menyatakan tingkat detail suatu citra, semakin tinggi resolusi citra maka semakin tinggi juga detail suatu citra. Informasi yang terdapat pada citra tergantung pada tipe data yang digunakan untuk menyatakannya. Nilai piksel pada dasarnya selalu bilangan biner dengan panjang k , nilai k disebut juga dengan kedalaman bit (*bit depth*). Citra yang memiliki kedalaman 1 bit disebut dengan citra biner. Setiap pikselnya dikodekan dalam 1 bit sehingga hanya terdiri dari dua kemungkinan, yaitu 0 untuk hitam dan 1 untuk putih. Citra biner biasanya didapati pada citra hasil pendeteksian tepi, sedangkan pada citra warna pembentukan warna didasarkan pada perpaduan warna primer yang memiliki kedalaman 8 bit per kanalnya. Warna primer yang dimaksud, yaitu merah (*red*), hijau (*green*), dan biru (*blue*) atau sering disingkat dengan *RGB*.

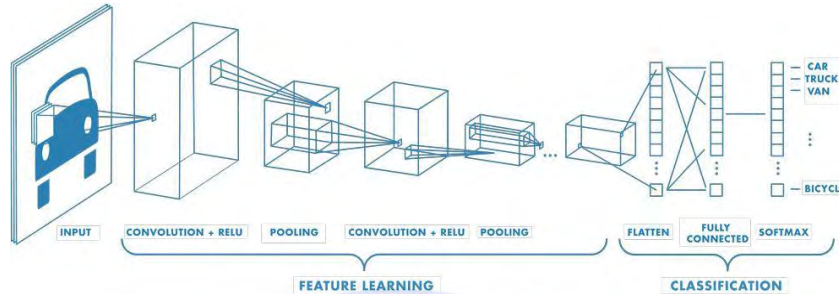
2.4 Deep Learning

Deep Learning merupakan sebuah teknik berbasis syaraf tiruan yang banyak digunakan sebagai salah satu metode implementasi *Machine Learning* (Haris et al., 2021). Tidak hanya spesifik untuk bidang tertentu, *Deep Learning* juga dapat menyelesaikan berbagai masalah di banyak bidang, seperti pengenalan suara, citra, bahasa, dll (Anshori, Fauzi, & Siadari, 2022), dan dapat memahami berbagai macam data dengan jumlah *layer* yang banyak. Perancangan *Deep Learning* dilakukan agar dapat menganalisa data contohnya otak manusia dalam mengambil keputusan (Yanto et al., 2021).

2.5 Convolutional Neural Network (CNN)

CNN digunakan pada bidang visi komputer dalam mengklasifikasikan gambar dan pengenalan objek, dan berbagai bidang visi lainnya (Mulyanto, Susanti, Rosi, Wajiran, & Borman, 2021). *CNN* menggambarkan varian dari *multilayer perceptron* yang mampu beroperasi pada data dua dimensi, seperti gambar atau suara dengan cara kerja yang mirip dengan jaringan syaraf manusia. Secara teknis, *CNN* adalah sebuah arsitektur yang

dipergunakan untuk melakukan klasifikasi data berlabel dengan menerapkan metode *supervised learning*, dimana data berlabel tersebut akan dilatih dengan beberapa tahapan *ted Layer*, dan *Dropout* (Peryanto et al., 2019).



Gambar 2. 5 Arsitektur *CNN* (Faizin et al., 2022)

2.5.1 Convolution Layer

Convolution Layer merupakan lapisan inti dan yang paling penting dari sistem untuk komputasi atau perhitungan perkalian pada lebar, tinggi, kedalaman suatu gambar (pixel) dan kernel, fungsi dari *Convolution Layer* adalah melakukan *filter* dan *feature map* pada masukan gambar (Ristiawanto et al., 2021).

Persamaan dan contoh operasi konvolusi :

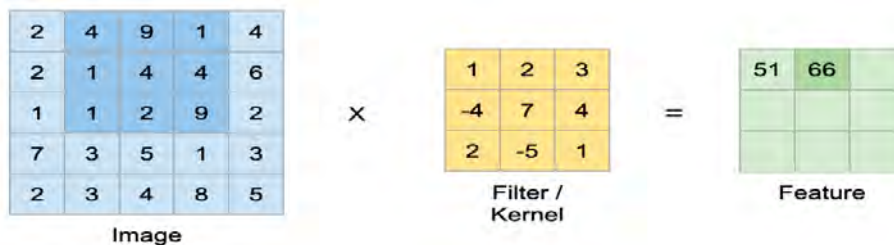
$$s(t) = (x * w)(t) \dots\dots\dots (2.2)$$

Keterangan :

S(t) = fungsi hasil operasi konvolusi

X = input

W = bobot (kernel)



Gambar 2. 6 Proses Konvolusi (Trisiawan et al., 2022)

2.5.2 Rectification Linear Unit (ReLU)

ReLU digunakan untuk mengenali non linearitas dan juga untuk meningkatkan kinerja suatu model ke dalam jaringan saraf (Rohim et al., 2019). Rumus ReLU dapat dilihat pada persamaan 2.3:

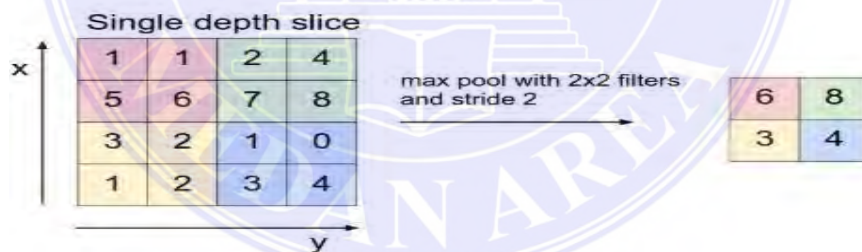
$$f(x) = \max(0, x) \dots\dots\dots (2.3)$$

2.5.3 Pooling Layer

Pooling Layer merupakan lapisan yang fungsinya untuk mengurangi mengurangi jumlah parameter dan komputasi, serta untuk menghindari kondisi *overfitting* dimana model sangat akurat dalam memprediksi data *training* tetapi gagal dalam mengenali data di luar data *training* (Hariyani et al., 2020).

Persamaan dan contoh operasi *max pooling* dapat dilihat pada persamaan 2.5 berikut:

$$y = \max_{i,j}^{h,w} x_{i,j} \dots\dots\dots (2.5)$$



Gambar 2. 7 Proses Max pooling (Peryanto et al., 2020)

2.5.4 Fully connected Layer

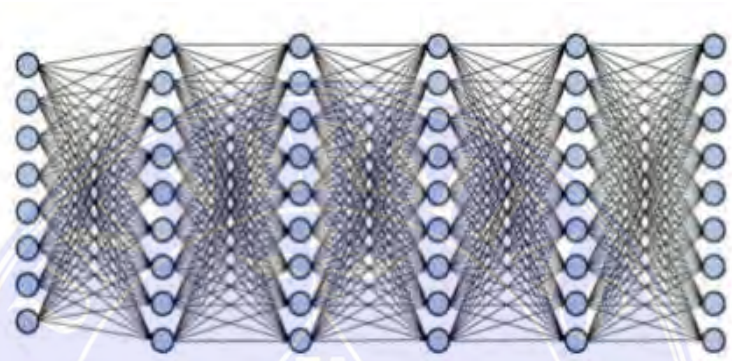
Fully connected layer adalah lapisan terakhir yang setiap neuron dari lapisan sebelumnya terhubung dengan neuron di lapisan selanjutnya yang digunakan sebagai output layer (Ilahi et al., 2022). Fully connected layer biasanya dipakai pada penerapan multilayer perceptron dengan tujuan untuk melakukan transformasi sehingga data dapat

diklasifikasikan (Yohannes & Al Rivan, 2022). Rumus persamaan dan ilustrasi dari *fully connected layer* dapat dilihat di bawah ini.

$$\sum_{i=1}^N I_i * V_{ij} = J_i \dots\dots\dots (2.6)$$

$$\sum_{i=1}^N J_i * W_{ij} = H_i \dots\dots\dots (2.7)$$

$$\sum_{i=1}^N H_i * X_{ij} = O_i \dots\dots\dots (2.8)$$



Gambar 2. 8 *Fully connected Layer* (Asrianda et al., 2021)

2.5.5 Softmax

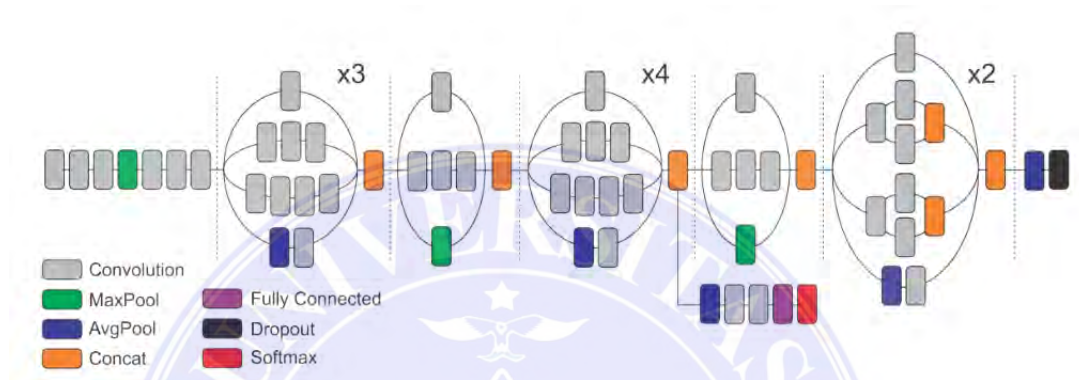
Softmax digunakan untuk mengorganisasi yang bertujuan untuk dapat mempunyai hasil klasifikasi. Point kelas didapat dengan menggunakan aktivasi *softmax*. Fungsi aktivasi *softmax* dapat ditulis sebagai berikut :

$$s(y) = \frac{e^y}{\sum e^y} \dots\dots\dots (2.9)$$

2.6 GoogLeNet

GoogLeNet adalah model yang diperkenalkan oleh google pada tahun 2014 dan menjadi peringkat pertama pada kompetisi ILSVRC yang mendapatkan predikat sebagai arsitektur dengan kinerja paling baik (Faizin et al., 2022). Arsitektur *GoogLeNet* merupakan salah satu modifikasi dari arsitektur *CNN* yang berhasil menjadi model terbaik karena mampu mendeteksi citra dengan 22 lapisan (27 lapisan termasuk lapisan *pooling*) dan

memiliki akurasi yang tinggi (Sa'idah et al., 2022). *GoogLeNet* menggunakan 3 filter dimensi yang berbeda untuk inputan gambarnya yaitu (1x1, 3x3, 5x5) dan menggabungkan karakteristik untuk mendapatkan hasil atau output (Almas et al., 2019). Arsitektur *GoogLeNet* dapat dikenali melalui penggunaan *layer Inception* yang lebih dari satu *layer* yang akan berguna untuk menaikkan nilai hasil.



Gambar 2. 9 Arsitektur *Inception-v3* (Ingrid, 2022)

GoogLeNet dapat mengurangi tingkat kesalahan hingga 6,67% dimana lebih unggul dari alexnet sebesar 15,3%. *GoogLeNet* menggunakan *batch normalization* agar inputan tersusun secara beruntun, dan distorsi citra digunakan untuk menyesuaikan ukuran citra sesuai keinginan, setelah melalui fase *Inception*, nantinya fitur akan melakukan *fully connected neuron* (Hafifah et al., 2021). Model arsitektur memiliki empat versi *Inception* yang disebut dengan *Inception v1*, *Inception v2*, *Inception v3*, dan *Inception v4*.

Arsitektur *Inception v3* juga mengikuti kompetisi ILSVRC dalam pengenalan dan klasifikasi image yang pada saat itu menggunakan sebanyak 1.4 juta image dengan 1000 lebih class, dibandingkan dengan *Inception v1*, model *Inception v3* memiliki kinerja yang lebih unggul dalam pengenalan objek (Lin et al., 2019). Secara khusus, model *Inception v3* mencakup tiga bagian dasar, yaitu *convolutional block*, *Inception module*, dan pengklasifikasian. *Convolutional block* yang diselingi dengan *max pooling* digunakan

sebagai ekstraksi fitur. *Inception* module biasanya memiliki tiga lapisan konvolusi yang berbeda dan satu *max pooling*. Setiap output pada lapisan model, dilakukan penyatuan setelah konvolusi, kemudian di padukan untuk mendapatkan struktur *Deep Learning*

2.7 Penelitian Terkait

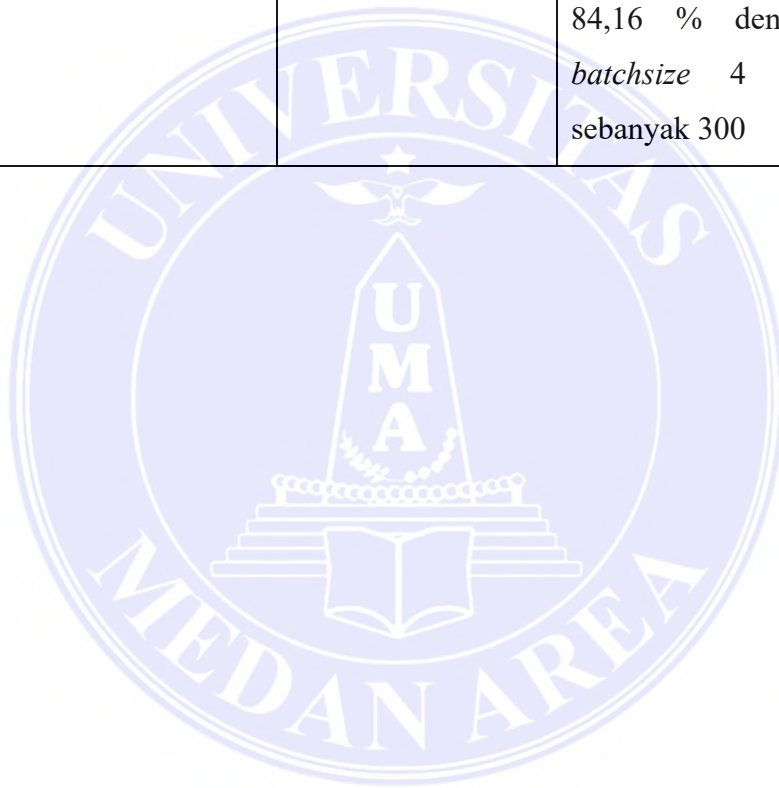
Penelitian sebelumnya berperan sebagai salah satu referensi utama dalam pelaksanaan sebuah penelitian. Hal ini disebabkan oleh peran penelitian sebelumnya dalam memperluas dan memperdalam teori yang akan digunakan pada kajian penelitian yang akan dilakukan. Secara umum, penelitian sebelumnya merupakan sumber dari hasil penelitian terdahulu yang dapat digunakan oleh peneliti untuk membandingkan penelitian yang akan dilakukan.

Tabel 2. 1 Ringkasan Penelitian Terkait

No.	Peneliti	Judul Penelitian	Keterangan
1.	Yuhang, Junru, Yuting, Xuemei, dll, 2023	<i>Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases</i>	Membandingkan dua arsitektur yaitu <i>Inception-v3</i> dan <i>resnet-50</i> dalam klasifikasi citra fundus, total data yang digunakan sebanyak 1032 yang terdiri dari dua kelas mata normal dan mata dengan fundus. Hasil akurasi dari kedua arsitektur masing-masing 91.76% dan 93.81%
2.	Mujahid M, dll, 2022	<i>Pneumonia Classification from X-ray Images with Inception-V3 and Convolutional Neural Network</i>	Melakukan klasifikasi penyakit pneumonia dengan menggunakan citra x-ray yang terdiri dari dua kelas yaitu paru-paru normal dan paru-paru pneumonia. Hasil

			dari <i>Inception-v3</i> mendapatkan akurasi sebesar 99.29%, <i>precision</i> 98.83%, <i>recall</i> 99.73%, dan <i>f1-score</i> 99.28%.
3.	Ung dan Galih, 2023	Klasifikasi Warna Pada Kematangan Buah Kopi Kuning Menggunakan Metode <i>CNN Inception-V3</i>	Klasifikasi Tingkat kematangan berdasarkan warna pada kopi kuning menggunakan <i>Inception-V3</i> mendapat hasil akurasi sebesar 92% dengan menggunakan <i>batch size SGD</i> , <i>learning rate</i> 20, <i>batch size</i> 32, dan <i>learning rate</i> 0.0001.
4.	B.Almas , K.Sathesh , dan S.Rajasekaran, 2019	<i>A Deep Analysis of GoogleNet and AlexNet for Lung Cancer Detection</i>	Menggunakan dua metode <i>CNN</i> yaitu <i>googlenet</i> dan <i>alexnet</i> dalam klasifikasi kanker paru-paru, penelitian ini menghasilkan nilai akurasi lebih dari 90%, yaitu masing-masing 99.05% dengan <i>running time</i> 46 jam 27 detik dan 99.91% dengan <i>running time</i> 20 jam 11 detik.
5.	Arif faizin, moh.lutfi, achmyatari, 2022	Perbandingan Arsitektur <i>Lenet</i> Dan <i>GoogLeNet</i> Dalam Klasifikasi <i>Diabetic Retinopathy</i> Pada Citra Retina <i>Fundus</i>	membandingkan dua metode <i>CNN</i> yaitu arsitektur <i>Lenet</i> dan <i>GoogLeNet</i> dalam mengklasifikasi <i>Diabetic Retinopathy</i> pada citra retina fundus, dataset yang digunakan bersumber dari

			<p>kaggle dengan jumlah data 2.500 citra, hasil yang didapatkan adalah arsitektur <i>Lenet</i> mendapatkan akurasi sebesar 95,40% dengan jumlah <i>batchsize</i> 16 dan <i>epoch</i> 100 sedangkan hasil dari arsitektur <i>GoogLeNet</i> mendapatkan akurasi sebesar 84,16 % dengan jumlah <i>batchsize</i> 4 dan <i>epoch</i> sebanyak 300</p>
--	--	--	--

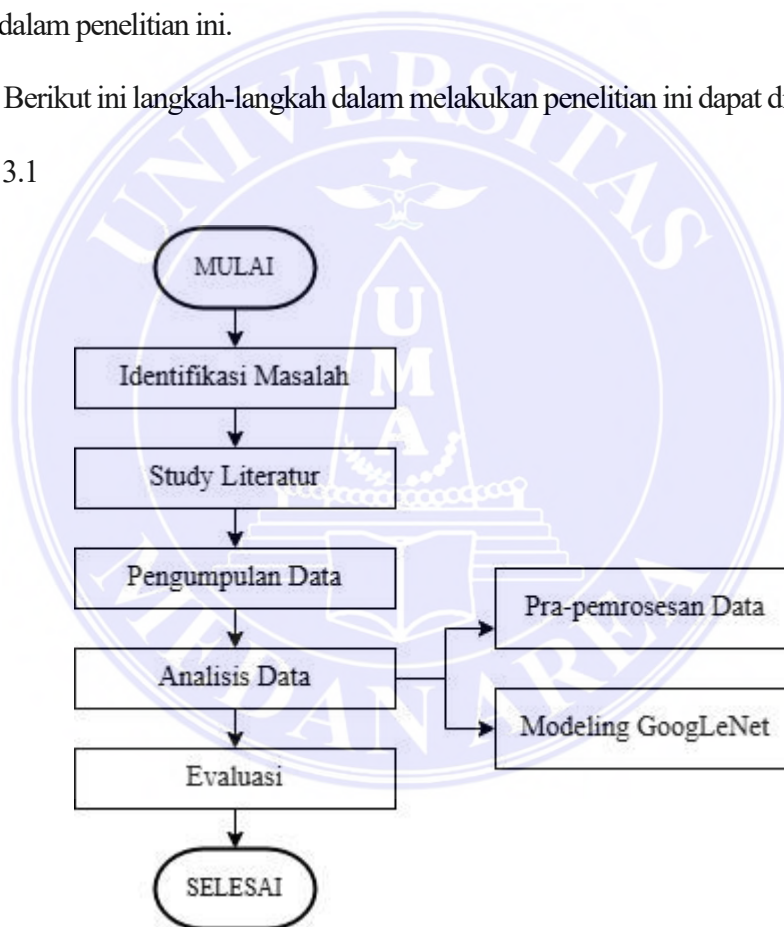


BAB III METODOLOGI PENELITIAN

3.1 Diagram Alur Penelitian

Diagram alur penelitian ini digunakan untuk mendukung jalannya penelitian, maka dibutuhkan langkah-langkah desain penelitian yang disusun secara terstruktur agar penelitian ini lebih terarah dan sistematis. Hal ini dilakukan untuk memberikan ketentuan dalam bentuk masalah dan sebagai pedoman dari langkah awal penelitian hingga ke langkah hasil yang dicapai dalam penelitian ini.

Berikut ini langkah-langkah dalam melakukan penelitian ini dapat ditampilkan pada gambar 3.1



Gambar 3. 1 Alur Penelitian Secara Umum

Berdasarkan diagram penelitian secara umum pada gambar 3.1 maka penelitian ini dapat diuraikan setiap langkah-langkah nya sebagai berikut:

3.2 Identifikasi Masalah

Identifikasi masalah merupakan langkah pertama yang harus dipersiapkan terlebih dahulu dalam melakukan sebuah penelitian. Langkah ini akan dijadikan topik penelitian untuk menemukan solusi yang akan menjadi tujuan dari penelitian tersebut. Masalah yang diangkat dalam penelitian ini adalah terkait dalam hal klasifikasi tahu yang mengandung formalin. Oleh karena itu diperlukan adanya pendekatan digital agar dapat mengenali tahu tersebut apakah mengandung formalin atau tidak, yang menggunakan metode yang dikembangkan dari *CNN* dengan arsitektur *GoogLeNet*.

3.3 Study Literatur

Studi literatur ini dilakukan agar mendapatkan landasan teori yang terkait dalam permasalahan penelitian yang akan diteliti. Pada tahap ini, pencarian dilakukan dengan mengakses mesin pencari di internet, membaca berbagai buku dan jurnal, meninjau hasil penelitian terdahulu, dan juga mencari sumber lain yang relevan dengan permasalahan utama dalam penelitian ini.



3.4 Pengumpulan Data

Pengumpulan data dilakukan dengan cara membeli susu kedelai sebanyak 2 kg, lalu diolah menjadi tahu secara mandiri. Prosesnya, susu kedelai dituang ke dalam dua panci yang berbeda yang berisikan 1 liter air panas, masing-masing berisi 1 kg susu kedelai. Wadah pertama yang akan menjadi tahu tanpa formalin, hanya menggunakan susu kedelai dan larutan asam (cuka dapur), larutan asam ditambahkan sedikit demi sedikit sampai terlihat adanya calon tahu (endapan tahu) lalu di didiamkan sampai endapan tahu banyak. Dan wadah kedua juga dicampurkan dengan larutan asam dan formalin sebanyak 1 sdt yang akan menjadi tahu formalin. Kemudian endapan tahu dipindahkan ke loyang yang sudah dilapisi

dengan kain, lalu ditutup dengan kain dan diberikan pemberat agar tahu menjadi padat dan berbentuk selama 2 jam.

Kemudian kedua jenis tahu tersebut diambil citranya menggunakan kamera ponsel dengan jarak kurang lebih 15-20 cm, dengan jumlah data yang dikumpulkan sebanyak 2200 data, format file *JPG*. Dari 2.200 data yang digunakan akan dibagi menjadi data *training* dan data validasi dengan masing-masing 1.000 data untuk *training* dan validasi sebanyak 200 data. Setelah melakukan pelatihan (*training*) terhadap model, maka diperlukan pengujian (*testing*). Data testing yang digunakan adalah data citra tahu yang diambil dari beberapa pasar dengan jumlah data sebanyak 200 data citra tahu. Berikut beberapa citra tahu formalin dan tahu non formalin.

Tabel 3. 1 Deskripsi Citra dan Jumlah Data Citra

Citra Tahu	Jumlah Data
 <p>Tahu Formalin</p>	1.000
 <p>Tahu Non-formalin</p>	1.000

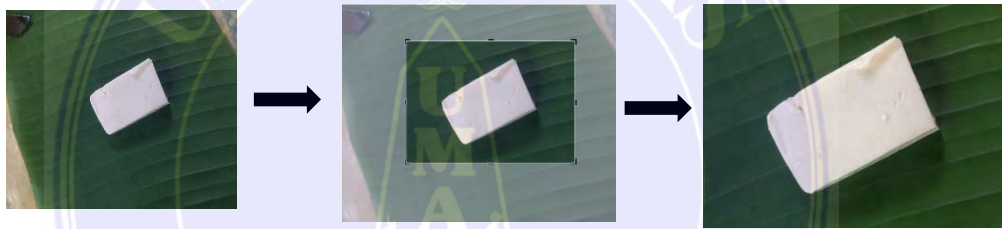
3.5 Analisis Data

Data yang telah dikumpulkan akan dianalisis yaitu dengan melakukan pra-pemrosesan. Pra-pemrosesan data adalah langkah yang dilakukan untuk menyiapkan data

mentah menjadi format yang sesuai untuk dilatih menggunakan model yang digunakan. Proses ini melibatkan beberapa langkah-langkah, yaitu sebagai berikut:

3.5.1 Cropping Data

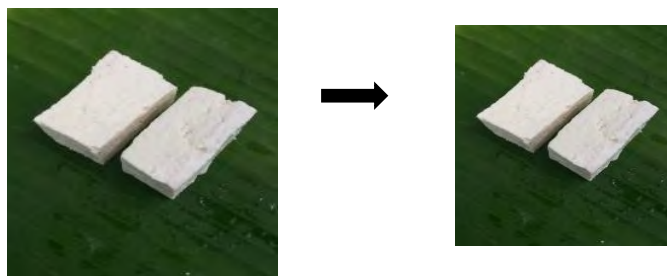
Cropping melibatkan pemotongan Sebagian dari gambar asli untuk membuat data gambar baru. Pemotongan yang dilakukan pada gambar yang diambil terlalu jauh, terlalu ke samping kanan atau kiri maupun ke bawah. Sehingga gambar yang baru akan mendapatkan hasil yang lebih bagus. Proses *Cropping* dilakukan menggunakan aplikasi *Microsoft office picture manager 2010*. Berikut merupakan contoh hasil *Cropping* atau perbandingan gambar asli dan gambar baru setelah dilakukan *Cropping*.



Gambar 3. 2 *Cropping* data gambar

3.5.2 Resizing Data

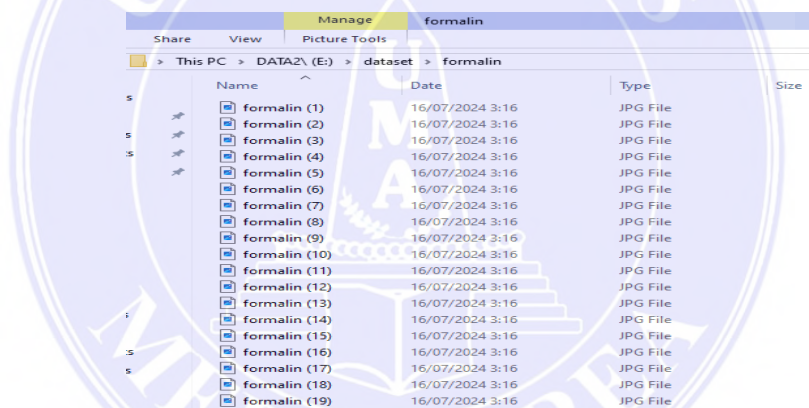
Resizing data adalah proses mengubah ukuran gambar atau data lain ke dimensi tertentu. Dalam penelitian ini semua gambar akan diubah ke ukuran pixel dengan Panjang dan lebar yang sama ukurannya yaitu 299 x 299 *pixel* berikut adalah contoh data gambar sebelum dan sesudah *Resizing* dilakukan, dapat dilihat pada gambar 3.3.



Gambar 3. 3 *Resizing* data gambar

3.5.3 Labeling Data

Tahapan selanjutnya yang dilakukan adalah melakukan *labelling* pada data gambar yang telah di *resize*. Proses *labelling* data ini merupakan proses memberikan label pada data mentah untuk mengklasifikasikan atau mendeskripsikan data tersebut. Pelabelan data dilakukan dengan cara manual, memilih semua *file* pada *folder* dataset di *file explorer* yang ingin diubah namanya dengan menekan tombol *ctrl + A* di *keyboard*, setelah itu klik kanan dan klik *rename* lalu mengetik nama sesuai dengan nama yang baru sesuai kelas datasetnya. Secara otomatis semua *file* yang dipilih namanya akan berubah, menjadi *formalin (1)*, *formalin (2)*, dan seterusnya sampai *formalin (1.200)*, dan *non formalin (1)*, *non formalin (2)*, dan seterusnya sampai *non formalin (1.200)*.



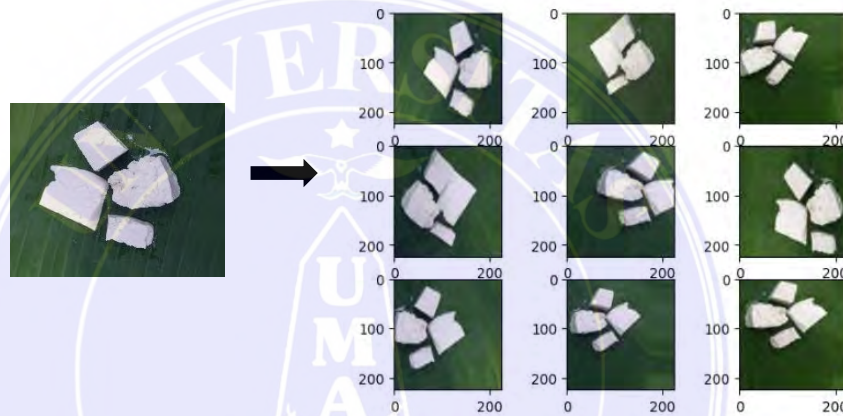
Gambar 3. 4 Proses *Labeling* data

3.5.4 Augmentasi Data

Pada tahap augmentasi data ini dilakukan agar data lebih bervariasi tanpa mengumpulkan data tambahan, hal ini penting dilakukan untuk mencegah terjadinya *overfitting* (memiliki kinerja yang baik saat proses *training* tetapi kinerja yang buruk pada saat proses validasi dan *testing*).

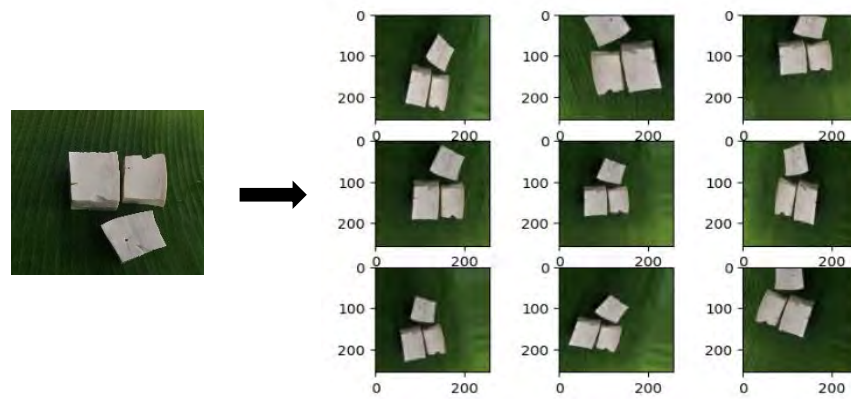
Augmentasi data dilakukan dengan cara *rotation* (memutar gambar tahu pada sudut tertentu) dengan kode *rotation_range = 20*, *width and height shift* (menggeser

gambar secara horizontal dan vertikal) dengan kode *width_shift_range* = 0.2 dan *height_shift_range* = 0.2, *rescale* (mengubah skala piksel pada gambar) dengan kode *rescale* = 1./255, *shear* (mengubah posisi gambar menjadi miring) dengan kode *shear_range* = 0.2, *zoom* (memperbesar citra) dengan kode *zoom_range* = 0.2, dan horizontal *flip* (membalik gambar secara horizontal) dengan kode *horizontal_flip* = 180. Seperti yang ditunjukkan pada gambar dibawah ini, merupakan hasil augmentasi dari salah satu data tahu formalin.



Gambar 3. 5 Hasil augmentasi tahu formalin

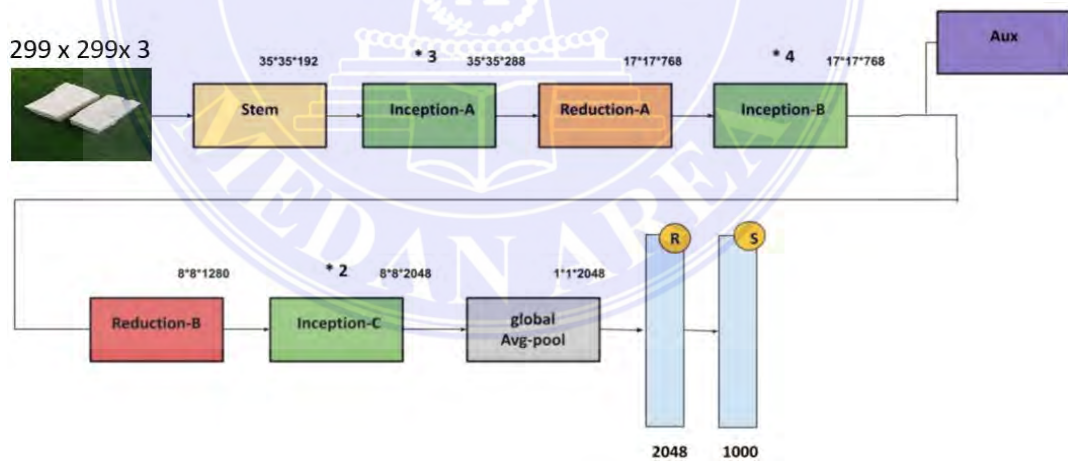
Untuk augmentasi data non formalin dilakukan juga hal yang sama seperti data formalin, yaitu dengan menggunakan Teknik augmentasi *rotation* dengan kode *rotation_range* = 20, *width and height shift* dengan kode *width_shift_range* = 0.2 dan *height_shift_range* = 0.2, *rescale* dengan kode *rescale* = 1./255, *shear* dengan kode *shear_range* = 0.2, *zoom* dengan kode *zoom_range* = 0.2, dan horizontal *flipping* dengan kode *horizontal_flip* = 180.



Gambar 3. 6 Hasil augmentasi data nonformalin

3.5.5 Modeling GoogLeNet

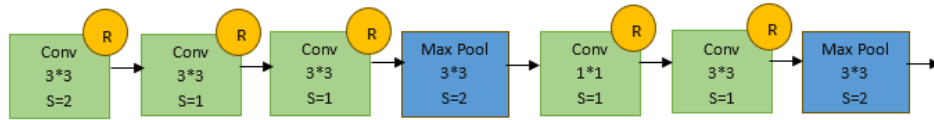
Tahap selanjutnya melakukan perancangan model untuk pelatihan yang dibuat. Perancangan model arsitektur *GoogLeNet* menggunakan imagenet dengan pixel $299 \times 299 \times 3$, angka tiga menandakan citra tersebut adalah chanel image *RGB* pada citra warna (true color). Modul *Inception* biasanya mencakup satu *max pooling* dan tiga konvolusi dengan ukuran yang berbeda. Berikut ini adalah gambaran arsitektur *GoogLeNet*.



Gambar 3. 7 Arsitektur *GoogLeNet Inception v3*

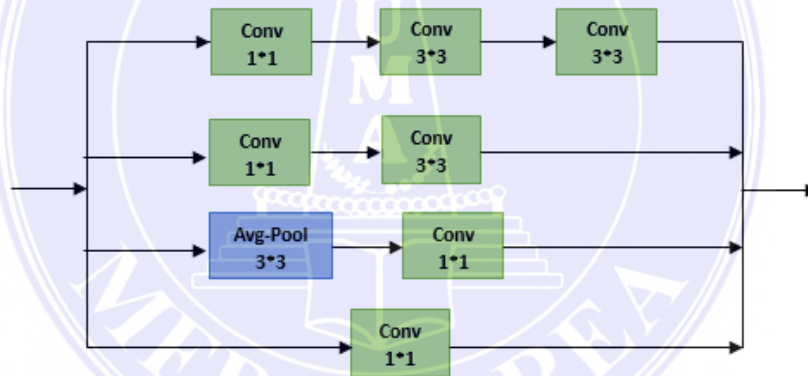
Pada dasarnya *Inception v3* dikembangkan berdasarkan *Convolutional Neural Network (CNN)*. Dalam arsitektur *Inception v3* yang dipaparkan pada gambar 3.3. Terdapat berbagai prosedur yang harus dilakukan diantaranya adalah konvolusi, *average pool*, *maxpool*, *dropout*, *fully connected*. Dan untuk fungsi aktivasi yang digunakan pada

penelitian ini adalah *softmax*, karena klasifikasi yang dilakukan merupakan klasifikasi dua kelas atau biasa disebut dengan *binary classification*. Stem block merupakan langkah awal yang dilalui oleh data inputan, yang terdiri dari konvolusi dan max pool dengan kernel dan stride yang berbeda.

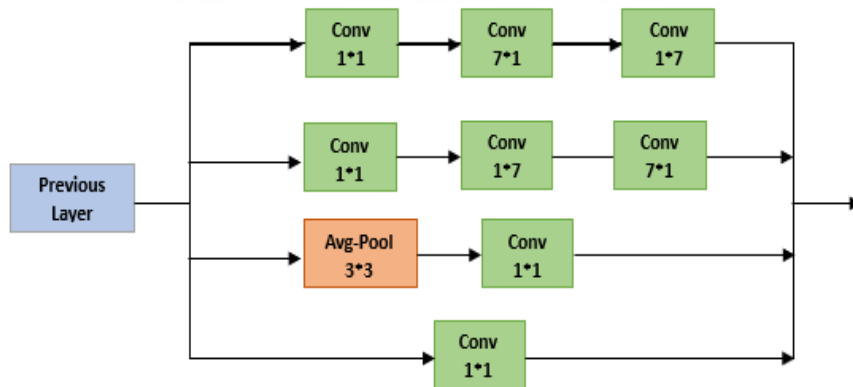


Gambar 3. 8 Stem Block

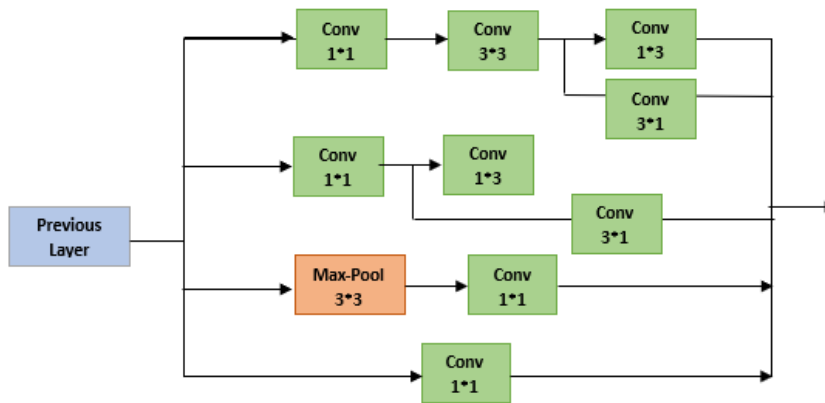
Arsitektur pada gambar 3.7 diatas menjelaskan langkah-langkah serta memaparkan kernel *default* yang digunakan di *Inception v3*. Proses *Inception* dibagi menjadi tiga yaitu *Inception A*, *Inception B* dan *Inception C*.



Gambar 3. 9 Inception A

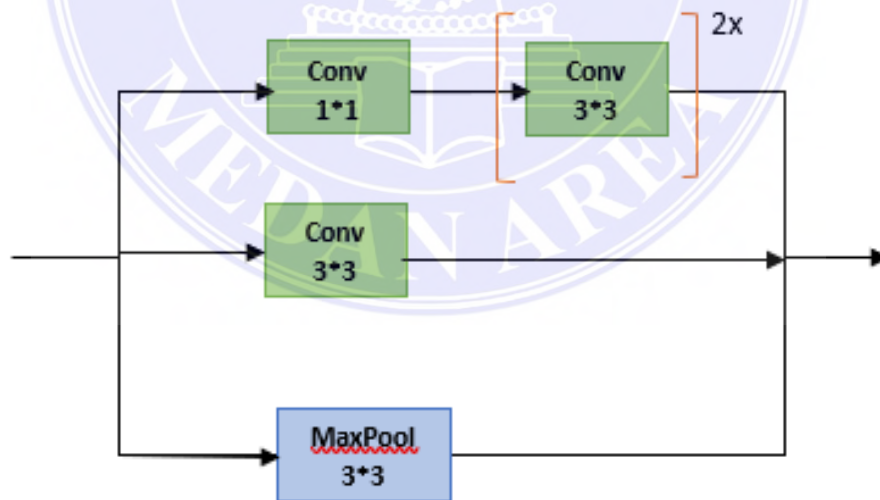


Gambar 3. 10 Inception B



Gambar 3. 11 *Inception C*

Pada umumnya *Inception* memiliki empat percabangan di setiap blok yang terdiri dari *pooling* dan konvolusi. *Inception A* dilakukan sebanyak tiga kali pengulangan, *Inception B* melakukan sebanyak lima kali pengulangan, dan *Inception C* melakukan sebanyak dua kali pengulangan. Inputan yang digunakan pada setiap proses atau tahap pada *Inception v3* ini adalah hasil output dari *layer* sebelumnya.



Gambar 3. 12 *Reduction A*

1	0	-1
1	0	-1
1	0	-1

Gambar 3. 15 Kernel 3x3

Langkah selanjutnya dilakukan proses perhitungan di setiap *channel* dengan dikalikan dengan kernel ukuran pada gambar 3.14. tahapan ini dilakukan berulang dengan pergeseran kernel sebanyak 2 *strides* di setiap *channel*nya. Kotak hijau pada ilustrasi perhitungan konvolusi di bawah akan bergeser sebanyak 2 *pixel* secara horizontal dan vertical. Sehingga didapatkan perhitungan di setiap *channel* dengan nilai sebagai berikut:

Posisi 1 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(0*1) + (0*0) + (0*(-1)) +$$

$$(0*1) + (42*0) + (40*(-1)) +$$

$$(0*1) + (45*0) + (39*(-1)) = -79$$

Posisi 2 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(0*1) + (0*0) + (0*(-1)) +$$

$$(40*1) + (49*0) + (36*(-1)) +$$

$$(39*1) + (43*0) + (40*(-1)) = 3$$

Posisi 3 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(0*1) + (0*0) + (0*(-1)) +$$

$$(36*1) + (40*0) + (59*(-1)) +$$

$$(40*1) + (49*0) + (46*(-1)) = -29$$

Posisi 4 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(0*1) + (0*0) + (0*(-1)) +$$

$$(59*1) + (39*0) + (0*(-1)) +$$

$$(46*1) + (48*0) + (0*(-1)) = 105$$

Posisi 5 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(0*1) + (45*0) + (39*(-1)) +$$

$$(0*1) + (48*0) + (36*(-1)) +$$

$$(0*1) + (44*0) + (41*(-1)) = -116$$

Posisi 6 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(39*1) + (43*0) + (40*(-1)) +$$

$$(36*1) + (44*0) + (39*(-1)) +$$

$$(41*1) + (47*0) + (34*(-1)) = 3$$

Posisi 7 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(40*1) + (49*0) + (46*(-1)) +$$

$$(39*1) + (43*0) + (36*(-1)) +$$

$$(34*1) + (41*0) + (47*(-1)) = -16$$

Posisi 8 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(46*1) + (48*0) + (0*(-1)) +$$

$$(36*1) + (43*0) + (0*(-1)) +$$

$$(47*1) + (35*0) + (0*(-1)) = 129$$

Posisi 9 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$(0*1) + (44*0) + (41*(-1)) +$$

$$(0*1) + (43*0) + (40*(-1)) +$$

$$(0*1) + (50*0) + (47*(-1)) = -128$$

Posisi 10 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(41*1) + (45*0) + (34*(-1)) + (40*1) + (39*0) + (38*(-1)) + (47*1) + (46*0) + (43*(-1)) = 13$$

Posisi 11 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(34*1) + (41*0) + (47*(-1)) + (38*1) + (45*0) + (50*(-1)) + (43*1) + (45*0) + (51*(-1)) = -33$$

Posisi 12 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(47*1) + (35*0) + (0*(-1)) + (50*1) + (51*0) + (0*(-1)) + (51*1) + (52*0) + (0*(-1)) = 148$$

Posisi 13 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(0*1) + (50*0) + (47*(-1)) + (0*1) + (50*0) + (49*(-1)) + (0*1) + (0*0) + (0*(-1)) = -96$$

Posisi 14 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(47*1) + (45*0) + (51*(-1)) + (48*1) + (53*0) + (54*(-1)) + (0*1) + (0*0) + (0*(-1)) = -10$$

Posisi 15 di *channel red*:

0	0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0	0
0	45	39	43	40	49	46	48	0	0
0	48	36	44	39	43	36	43	0	0
0	44	41	47	34	41	47	35	0	0
0	43	40	39	38	45	50	51	0	0
0	50	47	46	43	45	51	52	0	0
0	50	49	52	48	53	54	51	0	0
0	0	0	0	0	0	0	0	0	0

$$(43*1) + (45*0) + (51*(-1)) + (48*1) + (53*0) + (54*(-1)) + (0*1) + (0*0) + (0*(-1)) = -54$$

Posisi 16 di *channel red*:

0	0	0	0	0	0	0	0	0
0	42	40	49	36	40	59	39	0
0	45	39	43	40	49	46	48	0
0	48	36	44	39	43	36	43	0
0	44	41	47	34	41	47	35	0
0	43	40	39	38	45	50	51	0
0	50	47	46	43	45	51	52	0
0	50	49	52	48	53	54	51	0
0	0	0	0	0	0	0	0	0

$$\begin{aligned}
 &(51*1) + (52*0) + (0*(-1)) + \\
 &(54*1) + (51*0) + (0*(-1)) + \\
 &(0*1) + (0*0) + (0*(-1)) = 105
 \end{aligned}$$

Setelah perhitungan di *channel red* dengan ukuran 7x7, maka didapatkan matriks ukuran 4x4 dengan nilai seperti dibawah ini:

-79	3	105
-116	3	129
-128	13	148
-96	-10	105

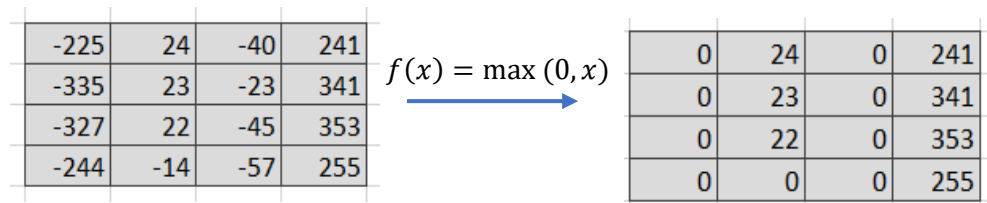
Selanjutnya dilakukan perhitungan dengan tahapan yang sama pada *channel green* dan *channel blue*, yaitu mengalikan kernel 3x3 pada gambar 3.14. Sehingga hasil yang didapatkan adalah sebagai berikut:

Green				Blue			
-70	15	0	55	-76	6	-11	81
-97	15	11	77	-122	5	-18	135
-89	6	7	77	-110	3	-19	128
-52	-5	-2	59	-96	1	-1	91

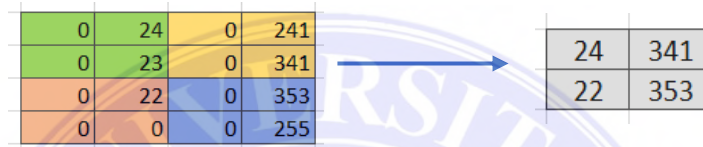
Setiap nilai masing-masing *channel* yang diperoleh melalui tahap perhitungan konvolusi, selanjutnya akan dijumlahkan sehingga menghasilkan output seperti berikut:

-225	24	-40	241
-335	23	-23	341
-327	22	-45	353
-244	-14	-57	255

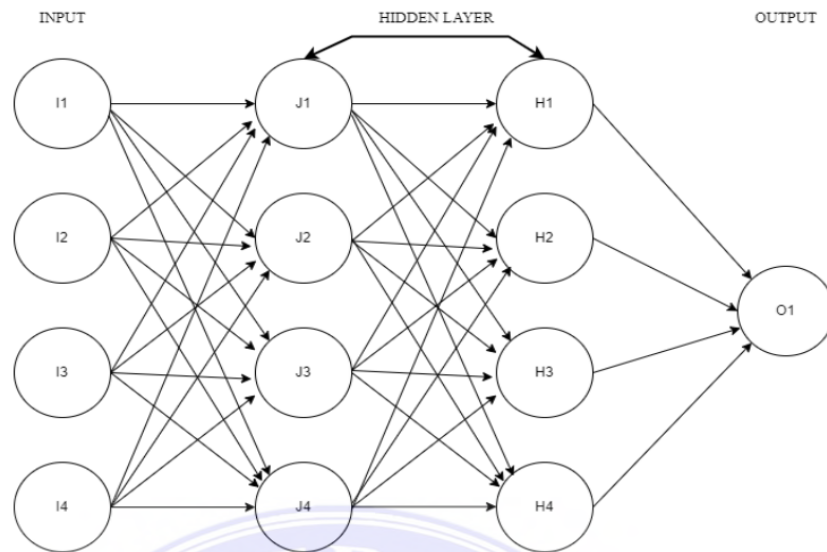
Dengan *activation ReLU*, maka setiap nilai yang negative akan diubah menjadi angka 0. Sehingga didapatkan hasil konvolusi seperti berikut:



Hasil *activation ReLU* diatas akan dijadikan inputan untuk tahap selanjutnya, yaitu *max pooling* menggunakan ukuran kernel 2x2 dengan 2 *stride*. Dapat dilihat pada proses *max pooling* berikut:



Tahapan *maxpooling* dilakukan dengan cara mengambil nilai tertinggi dari kernel 2x2 dan akan bergeser sebanyak 2 *strides*. Sehingga didapatkan output pada *layer maxpooling* yaitu 2x2x32 dengan nilai 24, 341, 22, dan 353. Output ini akan dijadikan sebagai inputan pada *global average pooling (GAP)*. GAP akan mengubah matrix 2x2 menjadi 4x1 atau vector 1D yang akan dijadikan sebagai input *fully connected layer (dense)*. Berikut ilustrasi *fully connected layer* seperti gambar 3.15 di bawah ini.



Gambar 3. 16 Ilustrasi fully connected *layer* dan *softmax*

Langkah terakhir dari model *CNN* yang dibangun ini merupakan dense dengan fungsi aktivasi *softmax*. Berikut proses perhitungan dari setiap *hidden layer* menggunakan persamaan 2.6 pada bab 2.

$$J1 = (24 * 0,2) + (341 * 0,2) + (22 * 0,2) + (353 * 0,2) = 148$$

$$J2 = (24 * 0,4) + (341 * 0,4) + (22 * 0,4) + (353 * 0,4) = 296$$

$$J3 = (24 * 0,3) + (341 * 0,3) + (22 * 0,3) + (353 * 0,3) = 222$$

$$J4 = (24 * 0,1) + (341 * 0,1) + (22 * 0,1) + (353 * 0,1) = 74$$

Hidden layer di ilustrasikan dengan J1, J2, J3, dan J4 dimana setiap neuron dari I1, I2, I3, dan I4 dikalikan dengan nilai *weight* (W) yang berbeda-beda. Sehingga menghasilkan nilai J1=148, J2=296, J3=222, dan J4=74. Hasil dari setiap neuron akan dikalikan lagi dengan *weight* yang berbeda-beda untuk menghasilkan nilai H1, H2, H3, dan H4 sebagai *hidden layer* berikutnya, proses perhitungan menggunakan persamaan 2.7 pada bab 2.

$$H1 = (148 * 0,4) + (296 * 0,4) + (222 * 0,4) + (74 * 0,4) = 296$$

$$H2 = (148 * 0,5) + (296 * 0,5) + (222 * 0,5) + (74 * 0,5) = 370$$

$$H3 = (148 * 0,3) + (296 * 0,3) + (222 * 0,3) + (74 * 0,3) = 325,6$$

$$H4 = (148 * 0,2) + (296 * 0,2) + (222 * 0,2) + (74 * 0,2) = 148$$

Setiap neuron dari H1, H2, H3, dan H4 akan dikalikan lagi dengan *weight* yang berbeda untuk menghasilkan nilai O1 dan O2 sebagai hidden *layer* berikutnya. Proses perhitungan pada tahap ini menggunakan persamaan 2.8 pada bab 2.

$$O1 = (296 * 0,1) + (370 * 0,1) + (325,6 * 0,1) + (148 * 0,1) = 113,96$$

$$O2 = (296 * 0,3) + (370 * 0,3) + (325,6 * 0,3) + (148 * 0,3) = 341,88$$

Selanjutnya adalah melakukan perhitungan *softmax* menggunakan nilai neuron O1 dan O2 serta menerapkan persamaan *softmax* 2.9 pada bab 2.

$$s(O_i) = \frac{e^{O_i}}{\sum_{j=1}^N e^{O_j}}$$

$$s(O_1) = \frac{e^{O_1}}{\sum_{j=1}^N e^{O_j}} = \frac{e^{113,96}}{e^{113,96} + e^{341,88}} = 0,2$$

$$s(O_2) = \frac{e^{O_2}}{\sum_{j=1}^N e^{O_j}} = \frac{e^{341,88}}{e^{113,96} + e^{341,88}} = 0,8$$

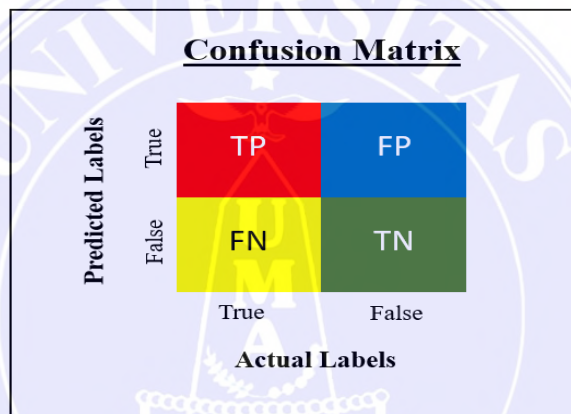
Sehingga didapatkan bobot nilai probabilitas yang lebih besar adalah O₂ yaitu 0.8 yang berarti input citra yang dimasukkan diprediksi adalah tahu.

3.6 Evaluasi

Tahap terakhir, yaitu evaluasi dilakukan untuk mengetahui apakah model yang digunakan dapat memberikan hasil yang akurat dalam mengklasifikasi citra tahu formalin.

Terdapat dua Teknik yang umum digunakan dalam tahap evaluasi ini yaitu *Confusion Matrix* dan *classification report*.

1. *Confusion Matrix* menyajikan informasi mengenai jumlah prediksi yang benar dari suatu model klasifikasi, yang sangat berguna untuk membandingkan nilai *actual* dengan nilai prediksi. Setiap baris dalam matriks merepresentasikan kelas yang sebenarnya, sementara setiap kolom merepresentasikan kelas yang diprediksi. Nilai *confusion* ini dapat menghitung *accuracy*, *precision*, *recall*, dan *F1-score* yang terdapat pada *classification report*.



Gambar 3. 17 Model *Confusion Matrix*

Berikut ini penjelasan lebih detailnya dari sebuah gambar 3.17 yaitu :

- a) *True positive* (TP): Prediksi nilai positif & nilai sebenarnya positif.
 - b) *True Negative* (TN): Prediksi nilai negatif & nilai sebenarnya negatif.
 - c) *False Positive* (FP): Prediksi nilai positif & nilai sebenarnya negative.
 - d) *False Negative* (FN): Prediksi nilai negatif & nilai sebenarnya positif.
2. *Classification report*, meskipun *confusion matrix* memberikan *detail* mengenai kinerja suatu model klasifikasi, sering kali masih sulit dipahami oleh pengguna dalam menilai seberapa baik kinerja tersebut. Oleh karena itu, *confusion matrix* dapat digunakan untuk menghitung metrik yang mengukur kinerja model, yang kemudian disajikan dalam bentuk *classification report* menggunakan berbagai metrik yaitu :

a) *Accuracy*

Merupakan parameter untuk membandingkan kelas yang diprediksi benar dengan total seluruh data yang digunakan. Akurasi digunakan untuk mengetahui efektifitas dari kelas dalam klasifikasi, dengan rumus sebagai berikut :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots (3.1)$$

b) *Precision*

Merupakan nilai yang menggambarkan ketepatan akurasi dari metode yang digunakan dalam melakukan klasifikasi di kelas dengan hasil prediksi model yang benar, dengan rumus sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (3.2)$$

c) *Recall*

keberhasilan model dalam menemukan kembali sebuah informasi untuk mendapatkan nilai yang dapat diukur hasilnya dalam bentuk persentase data yang terklasifikasi dengan benar, dengan rumus sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (3.3)$$

d) *F1-score*

Merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan, dengan rumus sebagai berikut:

$$F1 = 2x \frac{precision+recall}{precision+recall} \dots\dots\dots (3.4)$$

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diperoleh berdasarkan hasil pengujian klasifikasi tahu yang mengandung formalin menggunakan model arsitektur *GoogLeNet* adalah sebagai berikut:

1. Hasil *training* pada 8 model skenario yang diuji pada arsitektur *GoogLeNet*, diperoleh model terbaik yaitu pada model skenario ke-6 dan ke-7. dengan tingkat akurasi 99% menggunakan *hyperparameter* dengan jumlah *epoch* 30, *batch size* 32, *optimizer* *SGD*, dan *learning rate* 0.001, dan pada model skenario ke-7 dengan tingkat akurasi sebesar 99% dengan menggunakan *hyperparameter* de
2. Hasil evaluasi performa model skenario ke-6 menggunakan *confusion matrix* dengan data testing sebanyak 200 data citra menghasilkan nilai *accuracy* 99%, nilai *precision* 100%, *recall* 99%, dan *f1-score* 99%.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, saran yang dapat dipertimbangkan pada penelitian selanjutnya adalah menggunakan model arsitektur lain seperti *EffecientNet* sehingga dapat digunakan sebagai acuan penilaian performa model dari sudut pandang yang lain.

DAFTAR PUSTAKA

- Admojo, F. T., & Sulistyia, Y. I. (2022). Analisis Performa Algoritma Stochastic Gradient Descent (SGD) Dalam Mengklasifikasi Tahu Berformalin. *Indonesian Journal Of Data And Science (IJODAS)*, 3(1), 1–8.
- Almas, B., Sathesh, K., & Rajasekaran, S. (2019). A Deep Analysis Of Google Net And Alexnet For Lung Cancer Detection. *International Journal Of Engineering And Advanced Technology*, 9(2), 395–399.
- Andarwulan, N., Nuraida, L., Adawiyah, D. R., Triani, R. N., Agustin, D., & Gitapratwi, D. (2019). Pengaruh Perbedaan Jenis Kedelai Terhadap Kualitas Mutu Tahu. *Jurnal Mutu Pangan*, 5(2), 66–72.
- Arziyah, D., Yusmita, L., & Ariyetti. (2019). Analisis Mutu Tahu Dari Beberapa Produsen Tahu Di Kota Padang.
- Asrianda, A., Aidilof, H. A. K., & Pangestu, Y. (2021). Machine Learning For Detection Of Palm Oil Leaf Disease Visually Using Convolutional Neural Network Algorithm. *Journal Of Informatics And Telecommunication Engineering*, 4(2), 286–293.
- Asyfiradayati, R., Ningtyas, A., Lizansari, M., Purwati, Y., & Winarsih. (2019). Identifikasi Kandungan Formalin Pada Bahan Pangan (Mie Basah, Bandeng Segar Dan Presto, Ikan Asin, Tahu) Di Pasar Gede Kota Surakarta. In Online (Vol. 11, Issue 2). Desember.
- Barus, T., Wulandari, Y., Hutagalung, R. A., & Gunawan, A. W. (2019). Pelatihan Membuat Tahu Yang Sehat Pada Wkri Paroki St. Maria Fatima, Sentul City, Bogor. *Jurnal Bakti Masyarakat Indonesia*, 2(1), 145–154.
- Berliana, A., Abidin, J., Salsabila, N., Maulidia, N. S., Adiyaksa, R., & Siahaan, V. F. (2021). Penggunaan Bahan Tambahan Makanan Berbahaya Boraks Dan Formalin Dalam Makanan Jajanan. *Jurnal Sanitasi Lingkungan*, 1(2), 64–71.
- Dedy Kasingku, J., & Lumoindong, B. (2023). Peran Penting Pendidikan Lewat Makanan Bergizi Dalam Meningkatkan Kesadaran Masyarakat Akan Kesehatan Tubuh Dan Pikiran: Studi Literatur. *Journal On Education*, 05(04), 16071–16080.
- Dewi, S. R. (2019). Identifikasi Formalin Pada Makanan Menggunakan Ekstrak Kulit Buah Naga (Vol. 2).
- Faizin, A., Lutfi, M., & Achmyatari. (2022). Perbandingan Arsitektur Lenet Dan Googlenet Dalam Klasifikasi Diabetic Retinopathy Pada Citra Retina Fundus. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Issue 1).

- Fitriani Sammulia, S., Poluan, T., & Friscia Yusri, Y. (2020). Analisis Kualitatif Kandungan Formalin Pada Tahu Di Pasar Jodoh Kota Batam. *Jurnal Endurance*, 5(1), 144.
- Gifran, N. A., Magdalena, I. R., Yunendah, R., & Fuadah, N. (2019). Classification Of Cataract Using Discrete Wavelet Transform And Support Vector Machine.
- Gradian Landang, A., Bahantwelu, M., A Manu, A. K., & Studi Arsitektur, P. (2023). Perancangan Sentra Industri Tahu Terpadu Di Kota Kupang Dengan Pendekatan Arsitektur Hijau (Vol. 5, Issue 1).
- Hafifah, F., Rahman, S., & Asih, M. S. (2021). Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN). *TIN : Terapan Informatika Nusantar*, 2(5), 292–301.
- Haris, M., Pustaka, T., Diponegoro, M. H., Kusumawardani, S., & Hidayah, I. (2021). Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning Pada Prediksi Kinerja Murid (Implementation Of Deep Learning Methods In Predicting Student Performance: A Systematic Literature Review). In *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* | (Vol. 10, Issue 2).
- Hariyani, Y. S., Hadiyoso, S., & Siadari, T. S. (2020). Deteksi Penyakit Covid-19 Berdasarkan Citra X-Ray Menggunakan Deep Residual Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 8(2), 443–453.
- Hayat, F., & Darusmin. (2022). Analisis Faktor Penggunaan Formalin Pada Pedagang Tahu Di Pasar Tradisional Kota Serang. *Surya Muda*, 3(2), 121–132.
- Herdhiansyah, D., Reza, Sakir, & Asriani. (2022). Kajian Proses Pengolahan Tahu: Studi Kasus Industri Tahu Di Kecamatan Kabangka Kabupaten Muna. In *Agritech* (Issue 2).
- Ilahi, P. P., Fauzi, H., & Siadari, T. S. (2022). Klasifikasi Penyakit Pneumonia Dan Covid-19 Berbasis Citra X-Ray Menggunakan Arsitektur Deep Residual Network Classification Of Pneumonia And Covid-19 Based On X-Ray Images Using The Deep Residual Network Architecture.
- Lin, C., Li, L., Luo, W., Wang, K. C. P., & Guo, J. (2019). Transfer Learning Based Traffic Sign Recognition Using Inception-V3 Model. *Periodica Polytechnica Transportation Engineering*, 47(3), 242–250.
- Mujahid, M., Rustam, F., Álvarez, R., Luis Vidal Mazón, J., Díez, I. De La T., & Ashraf, I. (2022). Pneumonia Classification From X-Ray Images With Inception-V3 And Convolutional Neural Network. *Diagnostics*, 12(5).
- Pan, Y., Liu, J., Cai, Y., Yang, X., Zhang, Z., Long, H., Zhao, K., Yu, X., Zeng, C., Duan, J., Xiao, P., Li, J., Cai, F., Yang, X., & Tan, Z. (2023). Fundus Image

Classification Using Inception V3 And Resnet-50 For The Early Diagnostics Of Fundus Diseases. *Frontiers In Physiology*, 14.

Pardede, E., Novarika, W., & Sibuea, S. R. (2022). Analisis Pendapatan Industri Tahu Di Kelurahan Tanjung Gusta. *Factory Jurnal Industri, Manajemen Dan Rekayasa Sistem Industri*, 1(2), 59–66.

Peryanto, A., Yudhana, A., & Umar, R. (2019). Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network. In *Jurnal* (Vol. 8).

Ratna, S. (2020). Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm. In *Technologia* (Vol. 11, Issue 3).

Restiani, D. D., Sutiningsih, D., & Hestinationsih, R. (2020). Studi Keberadaan Cemaran Formalin Dan Timbal (Pb) Pada Tahu Yang Dijual Pedagang Gorengan Tahu Petis Di Sekitar Kampus Universitas Diponegoro.

Ristiawanto, S. P., Irawan, B., & Setianingsih, C. (2021). Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network Menggunakan Arsitektur Residual Network-50 Facial Expression Recognition Based On Convolutional Neural Network Using Residual Network 50 Architecture.

Rohim, A., Sari, Y., & Tibyani. (2019). Convolution Neural Network (CNN) Untuk Pengklasifikasian Citra Makanan Tradisional (Vol. 3, Issue 7).

Rosita, B., Chairani, Suraini, Erawati, Renowati, & Sophia, A. (2019). Penyuluhan Kesehatan Bahaya Makanan Yang Mengandung Zat Kimia Pada Murid Sdn 06 Balai Talang Kecamatan Guguak Kabupaten 50 Kota. *Abdimas Kesehatan Perintis*, 1(1), 31–34.

Rottie, R. (2020). Pengendalian Kualitas Tahu Om Melky Dengan Pendekatan Statistical Process Control. *Jurnal Realtech*, 15(1), 53–58.

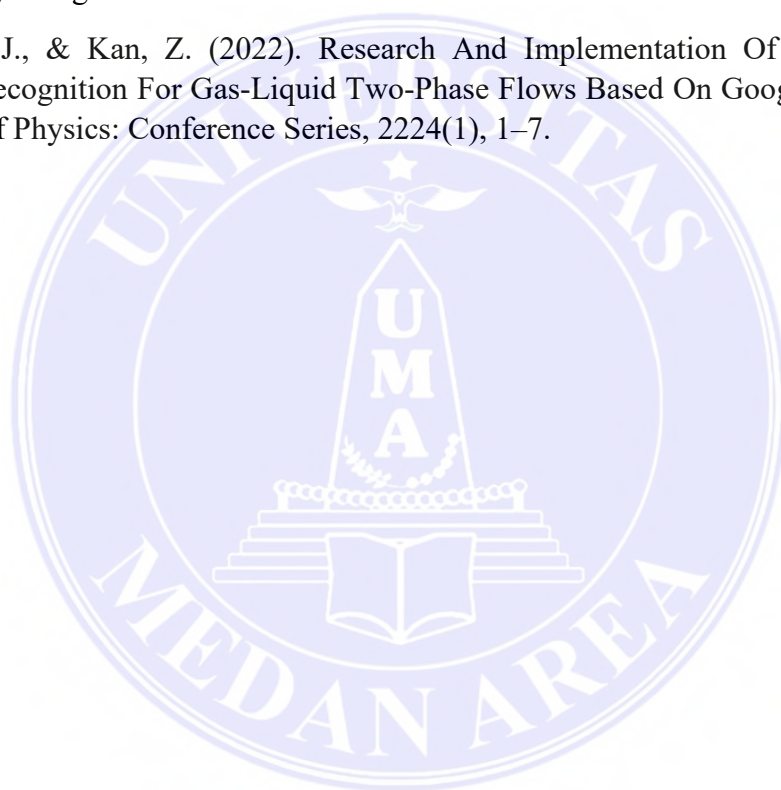
Sa'idah, S., Suparta, I., & Suhartono, E. (2022). Modification Of Convolutional Neural Network Googlenet Architecture With Dull Razor Filtering For Classifying Skin Cancer. In *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi* | (Vol. 11, Issue 2).

Saiman, L., & Satra, R. (2021). Analisis Performa Metode Support Vector Machine Untuk Klasifikasi Dataset Aroma Tahu Berformalin. *Indonesian Journal Of Data And Science (IJODAS)*, 2(2), 50–61.

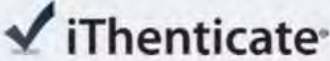
Setiawan, D. (2020). Tugas Dan Wewenang Balai Besar Pengawasan Obat Dan Makanan Dalam Mengawasi Makanan Yang Mengandung Zat Berbahaya. *Hukum Bisnis*, 4(2), 420–435.

Ungkawa, U., & Galih Al, H. (2023). Klasifikasi Warna Pada Kematangan Buah Kopi Kuning Menggunakan Metode CNN Inception V3. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(3).

- Waryat, Sudolar, N. R., Miskiyah, & Juniawati. (2020). Aplikasi Vinegar Sebagai Pengawet Alami Untuk Meningkatkan Umur Simpan Tahu. *Jurnal Ilmiah Respati*, 10(1), 41–48.
- Yanto, B., Fimawahib, L., Supriyanto, A., Hayadi, B., & Pratama, R. (2021). Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna Dengan Metode Deep Learning Convolutional Neural Network. *Jurnal Inovtek Polbeng*, 6(2), 259–268.
- Yohannes, R., & Al Rivian, M. E. (2022). Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM. *Jurnal Algoritme*, 2(2), 133–144.
- Yu, X., Wang, S.-H., Zhang, X., & Zhang, Y.-D. (2020). Detection Of Covid-19 By Googlenet-COD.
- Yuan, J., & Kan, Z. (2022). Research And Implementation Of Flow Pattern Recognition For Gas-Liquid Two-Phase Flows Based On Googlenet. *Journal Of Physics: Conference Series*, 2224(1), 1–7.



LAMPIRAN

Similarity Report ID: oid:29477:68118455

PAPER NAME	AUTHOR
AINUN FADHILLAH.pdf	AINUN FADHILLAH

WORD COUNT	CHARACTER COUNT
9628 Words	56089 Characters

PAGE COUNT	FILE SIZE
57 Pages	2.0MB

SUBMISSION DATE	REPORT DATE
Oct 5, 2024 9:06 AM GMT+7	Oct 5, 2024 9:07 AM GMT+7

- **11% Overall Similarity**
The combined total of all matches, including overlapping sources, for each database.
 - 7% Internet database
 - Crossref database
 - 6% Submitted Works database
 - 1% Publications database
 - Crossref Posted Content database
- **Excluded from Similarity Report**
 - Bibliographic material
 - Abstract
 - Cited material
 - Small Matches (Less than 15 words)



UNIVERSITAS MEDAN AREA

FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎ (061) 7366878, 7360168, 7364348, 7366781, Fax (061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 803/FT.6/01.10/XII/2023
Lamp : -
Hal : Perpanjang SK Pembimbing Tugas Akhir

8 Desember 2023

Yth. Pembimbing Tugas Akhir
Susilawati, S. Kom, M. Kom
di
Tempat

Dengan hormat,
Sehubungan telah berakhirnya waktu masa berlaku SK pembimbing nomor 389/FT.6/01.10/V/2023 tertanggal 29 Mei 2023 maka perlu diterbitkan kembali SK Pembimbing Skripsi baru atas nama mahasiswa berikut :

Nama : Ainun Fadhillah
N P M : 198160014
Jurusan : Teknik Informatika

Oleh karena itu kami mengharapkan kesediaan saudara :

Susilawati, S. Kom, M. Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

"Analisis Klasifikasi Tahu yang Mengandung Formalin menggunakan Model Arsitektur GoogleNet"

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,

Dr. Eng. Supriatno, ST, MT



UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 191 (061) 7366878, 7360168, 7364348, 7366781, Fax. (061) 7366998 Medan 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Seraya Nomor 70 A, TIR (061) 8225882, Fax. (061) 8226331 Medan 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 831 /FT.6/01.10/XII/2023
Lamp : -
Hal : **Penelitian Dan Pengambilan Data Tugas Akhir**

29 Desember 2023

Yth. Pimpinan Pabrik Tahu dan Tempe Saudara Kembar
Jln. Tanah Garapan No. 48, Percut Sei Tuan
Di
Deli Serdang

Dengan hormat,
Kami mohon kesediaan Bapak/Ibu berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	NAMA	NPM	PRODI
1	Ainun Fadhillah	198160014	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir pada perusahaan/Instansi yang Bapak/Ibu Pimpin.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :

Analisis Klasifikasi Tahu yang Mengandung Formalin menggunakan Model Arsitektur *GoogleNet*

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Tembusan :
1. Ka. BAMAI
2. Mahasiswa
3. File

Dekan,

Dr. Eng. Supriatno, ST, MT



PABRIK TAHU DAN TEMPE SAUDARA KEMBAR
Jl. Tanah Garapan No. 48
Deli Serdang

Deli Serdang, 29 Januari 2024

Hal : Selesai Penelitian dan Pengambilan Data Tugas Akhir

Kepada Yth. Dekan Fakultas Teknik
Universitas Medan Area
Di
Tempat

Dengan hormat,

Berdasarkan surat Dekan Fakultas Teknik Universitas Medan Area Nomor 831 /FT.6/01.10/XII/2023 tanggal 29 Desember 2023, perihal penelitian dan pengambilan data tugas akhir di Pabrik Tahu dan Tempe Saudara Kembar, maka bersamaan ini kami nyatakan bahwa mahasiswa di bawah ini:

Nama : Ainun Fadhillah
NPM : 198160034
Fakultas : Teknik
Program Studi : Teknik Informatika

Dinyatakan benar telah selesai melakukan penelitian dan pengambilan data di Pabrik Tahu dan Tempe Saudara Kembar, sehubungan dengan penyusunan skripsi "Analisis Klasifikasi Tahu Yang Mengandung Formalin Menggunakan Model Arsitektur GoogleNet".

Demikian surat keterangan ini dibuat dengan sebenarnya dan dapat dipergunakan sebagaimana mestinya.

Deli Serdang, 29 Desember 2024

Hormat Kami,


Dian Mery

Source Code

a) Impor Pustaka

```

import random
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf

from sklearn.preprocessing import LabelBinarizer
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
BatchNormalization, GlobalAveragePooling2D, Flatten, Dense, Dropout,
Concatenate, AveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.models import load_model
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras import layers, models
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint,
LearningRateScheduler
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, fbeta_score, confusion_matrix, classification_report
from PIL import Image

import glob
import cv2 as cv
import seaborn as sns

```

b) Melihat daftar file dan direktori dalam folder dataset

```

dataset_path = 'E:/dataset1'
os.listdir(dataset_path)

```

c) Pemanggilan Dataset

```

train_data = "E:/dataset1/train"
test_data = "E:/dataset1/test"
val_data = "E:/dataset1/val"

image_size = (299,299)
batch_size = 32

```


d) Menampilkan image masing-masing class

```

classes = ['formalin','nonformalin']
fig, axes = plt.subplots(1, len(classes), figsize=(12, 4))
for i, cls in enumerate(classes):
    img_path = os.path.join(train_data, cls,
        random.choice(os.listdir(os.path.join(train_data, cls))))
    img = plt.imread(img_path)
    axes[i].imshow(img)
    axes[i].set_title(cls)
    axes[i].axis('off')
plt.show()

```

e) Augmentasi data

```

#training
train_datagen = ImageDataGenerator(
    rescale=1/255,
    shear_range=0.2,
    zoom_range=0.3,
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=60,
    fill_mode='nearest')

#validasi dan testing
val_test_datagen = ImageDataGenerator(
    rescale=1/255,
    shear_range=0.2,
    zoom_range=0.3,
    horizontal_flip=True,
    vertical_flip=True,
    rotation_range=60,
    fill_mode='nearest')

#menyimpan data training, testing dan validasi
train_generator = train_datagen.flow_from_directory(
    train_data,
    target_size = (299,299),
    batch_size = batch_size,
    class_mode = 'categorical')

test_generator = val_test_datagen.flow_from_directory(
    test_data,
    target_size = (299,299),
    batch_size = batch_size,
    class_mode = 'categorical')

validation_generator = val_test_datagen.flow_from_directory(

```

```
val_data,
    target_size = (299,299),
    batch_size = batch_size,
    class_mode = 'categorical')
```

f) Parameter yang digunakan pada model

```
input_shape = (299, 299, 3)
n_classes = 2
batch_size = 32
epoch = 30
```

g) Membangun model GoogleNet menggunakan pre-trained imagenet

```
base_model = InceptionV3(weights='imagenet', include_top=False,
input_shape=input_shape)

# penambahan custom layer
x = base_model.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(512, activation='relu')(x)
predictions = layers.Dense(n_classes, activation='softmax')(x)

# Menggabungkan model dasar dengan model custom
model = models.Model(inputs=base_model.input, outputs=predictions)

model.summary()
```

```
for layer in base_model.layers[:50]:
    layer.trainable = False
```

h) Kompilasi model

```
model.compile(optimizer=SGD(learning_rate=0.001),
loss='categorical_crossentropy', metrics=['accuracy'])

checkpoint = ModelCheckpoint("best_modelGooglenet.h5",
monitor='val accuracy', save_best_only=True, mode='max')
```

i) Training data

```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=30,
    validation_data=validation_generator,
```



```
validation_steps=validation_generator.samples // batch_size,  
callbacks=[checkpoint]  
)
```

j) Akurasi data testing

```
test_loss, test_accuracy = model.evaluate(  
    test_generator  
)  
  
print('Test Loss:', test_loss)  
print(f'Test Accuracy: {test_accuracy*100:.2f}%")
```

k) Grafik akurasi training dan validasi

```
train_accuracy = history.history['accuracy']  
train_loss = history.history['loss']  
  
plt.figure(figsize=(12, 6))  
  
plt.subplot(1, 2, 1)  
plt.plot(history.history['accuracy'], label='GoogleNet Training Accuracy')  
plt.plot(history.history['val_accuracy'], label='GoogleNet Validation  
Accuracy')  
plt.title('Training and Validation Accuracy')  
plt.xlabel('Epoch')  
plt.ylabel('Accuracy')  
plt.legend()  
  
plt.tight_layout()  
plt.show()
```

l) Grafik loss training dan validasi

```
plt.figure(figsize=(12, 6))  
  
plt.subplot(1, 2, 2)  
plt.plot(history.history['loss'], label='GoogleNet Training Loss')  
plt.plot(history.history['val_loss'], label='GoogleNet Validation Loss')  
plt.title('Training and Validation Loss')  
plt.xlabel('Epoch')  
plt.ylabel('Loss')  
plt.legend()  
  
plt.tight_layout()  
plt.show()
```

m) Classification report dan confusion matrix

```
model_prediksi = load_model("best_modelGooglenet.h5")

y_pred = model_prediksi.predict(test_generator, verbose=1)
y_pred = np.argmax(y_pred, axis=1)

#confusion matrix dan classification report
batch_size=32
target_size=(299,299)
test_path = "E:/dataset1/test"

test_generator = val_test_datagen.flow_from_directory(
    test_data,
    target_size=target_size,
    batch_size=batch_size,
    class_mode=None,
    shuffle=False)
test_generator.reset()

model_klasifikasi = load_model("best_modelGooglenet.h5")
test_labels = test_generator.classes
pred = model_klasifikasi.predict(test_generator, verbose=1)
predictions=np.argmax(pred,axis=1)

cm = confusion_matrix(test_labels, predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='magma', xticklabels=classes,
yticklabels=classes)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()

cr = classification_report(test_labels, predictions, digits=4)

print(cr)
```