DETEKSI TINGKAT KEMATANGAN BUAH NANAS MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) DENGAN ARSITEKTUR *VISUAL GEOMETRY GROUP* (VGG) 16

(Skripsi)

Oleh

Demila

JURUSAN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2023

ABSTRAK

DETEKSI KEMATANGAN BUAH NANAS MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) DENGAN ARSITEKTUR *VISUAL GEOMETRI GROUP* (VGG) 16

Oleh

DEMILA

Telah direalisasikan sebuah sistem identifikasi tingkat kematangan buah nanas yang dilakukan secara *non*-destruktif berbasis *convolutional neural network* (CNN). Penelitian ini dilakukan dengan tujuan untuk membuat sistem yang dapat mendeteksi identifikasi tingkat kematangan buah nanas. *Convolutional neural network* (CNN) digunakan untuk mengklasifikasi kematangan buah nanas dengan melewati arsitektur yang telah dirancang, pada penelitian ini menggunakan model arsitektur *Visual Geometry Group* (VGG) 16. Melakukan pelatihan dataset buah nanas (50% data latih, 20% data validasi dan 30% data uji) dengan menggunakan model arsitektur VGG16. *Input* citra dari CNN berupa buah nanas yang diambil dari hasil tangkapan kamera/*webcam* untuk dideteksi tingkat kematangannya. Pada penelitian ini menghasilkan sebuah sistem deteksi kematangan buah nanas yang terdiri dari perangkat lunak dan perangkat keras yang mampu mengidentifikasi tingkat kematangan buah nanas dengan akurasi pelatihan dataset sebesar 96% dan pada hasil akurasi sistem deteksi tingkat kematangan buah nanas sebesar 100%.

Kata kunci: Buah nanas, CNN, VGG16, citra, python

ABSTRACT

DETECTION OF PINEAPPLE MATURITY USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD WITH VISUAL GEOMETRY GROUP (VGG) ARCHITECTURE 16

By

DEMILA

A system for identifying the ripeness level of pineapple fruit has been realized which is carried out non-destructively based on a convolutional neural network (CNN). This research was carried out with the aim of creating a system that can identify the ripeness level of pineapple fruit. Convolutional neural network (CNN) is used to classify the ripeness of pineapple fruit by passing through the architecture that has been designed, in this study using the Visual Geometry Group (VGG) 16 architecture model. Carrying out training on the pineapple fruit dataset (50% training data, 20% validation data and 30 % test data) using the VGG16 architectural model. The input image from CNN is a pineapple taken from a camera/webcam capture to detect its ripeness level. This research produces a pineapple ripeness detection system consisting of software and hardware that is capable of identifying pineapple ripeness levels with a training dataset accuracy of 96% and the accuracy of the pineapple ripeness level detection system is 100%.

Keywords: pineapple, CNN, VGG16, image, python

DETEKSI TINGKAT KEMATANGAN BUAH NANAS MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) DENGAN ARSITEKTUR *VISUAL GEOMETRY GROUP* (VGG) 16

Oleh

Demila

Skripsi

Sebagai Salah satu Syarat untuk Mencapai Gelar

SARJANA SAINS

Pada

Jurusan Fisika

Fakultas Matematika dan Ilmu Pengetahuan Alam

JURUSAN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2023

Judul Skripsi	:	**DETEKSI TINGKAT KEMATANGAN BUAH NANAS MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN) DENGAN ARSITEKTUR *VISUAL GEOMETRY GROUP* (VGG) 16**
Nama Mahasiswa	:	Demila
Nomor Pokok Mahasiswa	:	1957041006
Jurusan	:	Fisika
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Arif Surtono, S.Si., M.Si., M.Eng
NIP. 197109092000121001

Gurum Ahmad Pauzi, S.Si., M.T.
NIP. 198010102005011002

2. Ketua Jurusan Fisika

Gurum Ahmad Pauzi, S.Si., M.T.
NIP. 198010102005011002

MENGESAHKAN

1. Tim Penguji

Ketua : **Arif Surtono, S.Si., M.Si., M.Eng.** .....

Sekretaris : **Gurum Ahmad Pauzi, S.Si., M.T.** .....

Penguji
Bukan Pembimbing : **Drs. Amir Supriyanto, M.Si.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi : **13 Desember 2023**

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepengetahuan saya tidak ada karya atau pendapat yang ditulis atau diterbitkan oleh orang lain kecuali yang secara tertulis diacu dalam naskah ini sebagaimana disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 14 Desember 2023

Demila
NPM. 1957041006

RIWAYAT HIDUP

Penulis bernama lengkap Demila lahir pada tanggal 08 November 2001 di Duri Riau. Dari pasangan Bapak Windra Effendi dan Ibu Risnawati sebagai anak kedua dari tiga bersaudara. Penulis ini menyelesaikan pendidikan di SDN Kubang Laban pada tahun 2014, SMPN 2 Bojonegara pada tahun 2016, SMAN 1 Bojonegara pada tahun 2019, tepatnya di daerah Kab. Serang, Banten. Pada tahun 2019, penulis ini diterima sebagai mahasiswa di Universitas Lampung Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam melalui jalur Mandiri.

Selama menjadi mahasiswa, penulis aktif dalam Himpunan Mahasiswa Fisika (HIMAFI) sebagai anggota biro Kesekretariatan (KRT) dan aktif juga dalam Badan Eksklusi Mahasiswa (BEM) FMIPA sebagai anggota Dinas Pemberdayaan Wanita (PW) dan Sekretaris Dinas Advokasi Kesejahteraan Mahasiswa (Adkesma). Penulis melaksanakan Praktek Kerja Lapang (PKL) di Badan Tenaga Nuklir Nasional (BATAN) tepatnya di bidang Pusat Rekayasa Fasilitas Nuklir (PRFN) dengan judul **"Penentuan *Object Counter* Pada Portal Monitor Radiasi (PMR) Dengan Menggunakan Sensor Okupansi *Proximity Sensor Infrared Photoelectric* di Pusat Rekayasa Fasilitas Nuklir"**. Selain itu, penulis melaksanakan Kuliah Kerja Nyata (KKN) di Desa Sinas Petir, Tanggamus pada tanggal 27 Juni sampai dengan 05 Agustus 2022. Penulis juga menyelesaikan penelitian skripsi di Jurusan fisika dengan judul **"Deteksi Tingkat Kematangan Buah Nanas Menggunakan Metode *Convolutional Neural Network* (CNN) dengan Algoritma *Visual Geometry Group* (VGG) 16"**.

MOTTO

"Dunia adalah sebuah perjalanan, cukup di jalani sesuai aturan yang sudah ditentukan, tidak perlu berlari tidak juga dengan berdiam, berjalan dengan hati-hati hingga sampai pada tujuannya nanti"

(Demila, 2023)

"Sesungguhnya Bersama Kesulitan Ada Kemudahan, maka Apabila Engkau telah Selesai, Tetaplah bekerja Keras dan Hanya Kepada Allah lah Engkau Berharap"

--- QS. Al Insyirah: 6-8 ---

"Sesungguhnya Allah tidak akan Mengubah Keadaan Suatu Kaum Sebelum Mereka Mengubah Keadaan Diri Mereka Sendiri"

--- QS. Ar Ra'd: 11 ---

PERSEMBAHAN

Dengan Mengaharapkan Ridho Allah SWT dan Syafaat Nabi Muhammad SAW

Karya ini saya persembahkan kepada Bapak dan Ibu ku Tercinta

WINDRA EFFENDI & RISNAWATI

Terimakasih atas Do'a yang tidak pernah putus untuk anakmu ini serta segala upaya untuk memberikan yang terbaik untukku.

Bapak Ibu Guru serta Bapak Ibu Dosen

Terimakasih atas segala didikan serta bimbingannya yang telah diberikan, semoga apa yang telah disampaikan bermanfaat kelak menjadi bekal untuk kesuksesan kedepannya.

Kakak dan adik tercinta, Dody Alfajar dan Gea Ramadhani

Terimakasih atas dukungan dan semangatnya, yang selalu memberikan motivasi

Rekan-rekan seperjuangan Fisika angkatan 2019

Almamater Tercinta
UNIVERSITAS LAMPUNG

KATA PENGANTAR

Assalamualaikum Warahmatullah Wabarakatuh.

Puji syukur penulis haturkan atas karunia Allah SWT, karena atas berkat, rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul **"Deteksi Kematangan Nanas Menggunakan Metode *Convolutional Neural Network* (CNN) dengan Arsitektur *Visual Geometry Group* (VGG) 16"**. Penulis menyadari dalam penulisan skripsi ini masih terdapat banyak kesalahan dan kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran demi perbaikan kekurangan tersebut. Semoga skripsi ini dapat bermanfaat dan menjadi literatur serta rujukan bagi penelitian berikutnya.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Bandar Lampung, 14 Desember 2023

Penulis,

Demila

SAWACANA

Alhamdulillahirobbil'alamin, puji syukur penulis haturkan atas karunia Allah SWT, karena atas berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam tak lupa penulis sampaikan kepada Nabi Muhammad SAW, karena dengan perantaranya kita semua dapat merasakan indahnya kehidupan ini.

Skripsi dengan judul **"Deteksi Kematangan Nanas Menggunakan Metode *Convolutional Neural Network* (CNN) dengan Algoritma *Visual Geometry Group* (VGG) 16"** ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Sains pada Jurusan Fisika FMIPA Universitas Lampung. Selama proses pengerjaan skripsi ini, penulis telah menerima banyak bantuan pemikiran, semangat serta dorongan dari berbagai pihak. Dengan segala kerendahan hati, penulis menghaturkan terima kasih kepada:

1. Bapak Arif Surtono, S.Si., M.Si., M.Eng. Selaku dosen pembimbing I dan Pembimbing Akademik yang telah memberikan bimbingan, saran, motivasi, serta ilmunya selama melakukan penelitian dan penulisan skripsi ini.
2. Bapak Gurum Ahmad Pauzi, S.Si., M.T. Selaku dosen pembimbing II dan Ketua Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan masukan, saran dan nasihat dalam melakukan penelitian dan penulisan skripsi ini.
3. Bapak Drs. Amir Supriyanto, M.Si. Selaku dosen pembahas yang telah memberikan saran dan masukan sehingga penulisan skripsi ini dapat lebih baik.
4. Bapak Dr. Eng Heri Satria, S.Si., M.Si. Selaku Dekan FMIPA Universitas Lampung.

5. Segenap dosen yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh pendidikan di Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
6. Orang tuaku, Bapak Windra Effendi dan Ibu Risnawati serta kakak dan adikku yang selalu memberikan semangat dan dukungan kepada penulis selama menempuh pendidikan di Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
7. Arman Juliansyah yang sudah memberikan dukungan, motivasi, semangat, menemani mengurus skripsi ini dan sebagai tempat cerita.
8. Wayan Risti Putri salah satu kakak dari Jurusan Fisika yang telah membantu penulis selama penyelesaian skripsi.
9. Teman-teman seperjuangan Sasmita Ningrum, Icha Arum Vicias, Muhamad Ridwan, Eka Fadhilah Irawan, Zakiyyah Nur Hafizhah, Dian Permatasari, Nur Tasya Febrianti, Adhito Dwi Danendra, Aulia Nofdizhar Baehaqi dan teman-teman seperjuangan Fisika 2019 yang telah membantu dan memberikan semangat dalam menyelesaikan skripsi ini.10.
10. Teman-teman pimpinan BEM FMIPA dan HIMAFI periode 2021.
11. Teman-teman di luar kampus Winanda Nurfadiyah, Azira Salsabila dan Eka Martiniyah.
12. Semua pihak yang tidak dapat disebutkan satu persatu, yang telah memberikan moril maupun material pada penulis.

Bandar Lampung, 14 Desember 2023

Penulis,

Demila

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

I. PENDAHULUAN

1.1. Latar Belakang

Menurut Kementrian Pertanian (2015) sektor pertanian merupakan sektor yang memiliki peran strategis dalam struktur pembangunan ekonomi negara, salah satunya tanaman perkebunan. Menurut Kementrian Pertanian (2016) nanas merupakan salah satu tanaman hortikultura unggulan Indonesia yang dikenal dunia dan termasuk dalam lima besar penghasil nanas di dunia. Menurut Badan Pusat Statistik (BPS), produksi nanas di Indonesia mencapai 2,89 juta ton pada tahun 2021. Jumlah tersebut tumbuh 17,95% dibandingkan pada tahun sebelumnya, yaitu sebesar 2,45 juta ton. Berdasarkan wilayahnya, Lampung menjadi penghasil nanas terbesar di Indonesia sebesar 705.883 ton pada tahun 2021. Jumlah tersebut setara dengan 24,46% dari total produksi nanas di Indonesia sepanjang tahun lalu (Sadya, 2022).

Buah nanas sangat banyak diminati masyarakat Indonesia karena memiliki rasa yang cukup manis dan segar (Lubis, 2020). Selain rasanya yang manis, buah nanas juga mengandung serat yang berguna untuk melancarkan proses pencernaan, menurunkan kolesterol darah dan mengurangi risiko diabetes dan penyakit jantung. Serat dari 150 gram potongan nanas setara dengan setengah buah jeruk. Menurut penelitian, nanas mengandung asam amino esensial dan tak tergantikan yang membantu memperkuat sistem imun tubuh pada manusia, mengatasi kelelahan serta meningkatkan daya tahan tubuh dan energi, valin dan leusin yang terdapat pada nanas juga diperlukan untuk pertumbuhan dan pemulihan otot tubuh. Nanas mengandung *cystine* yang berguna untuk pembentukan kulit dan rambut, penting untuk pembentukan formasi kulit dan mempercepat penyembuhan luka (Winastia, 2011).

Saat ini perkembangan teknologi telah maju dan mendorong manusia untuk mengembangkannya. Berbagai inovasi baru dalam teknologi diharapkan dapat membuat pekerjaan manusia menjadi lebih mudah dan efisien. Salah satunya, adalah teknologi yang dapat membantu manusia dalam penentuan warna, yang semula harus dilakukan secara manual menggunakan penglihatan manusia, berubah menjadi penentuan warna secara otomatis atau sering disebut sebagai sensor elektronik. Teknologi membantu manusia dalam penentuan berbagai warna, biasanya pada penentuan warna pada makhluk hidup atau benda mati sekalipun (Purba *et al*., 2022).

Penilaian kematangan buah nanas telah dilakukan dengan bantuan teknologi dan menggunakan berbagai teknik pengolahan citra dan metode kecerdasan buatan seperti logika fuzzy dan jaringan syaraf tiruan (Azman dan Ismail, 2017). Berbagai metode telah digunakan oleh para peneliti untuk mengukur pemutuan kematangan pada buah nanas seperti pengukuran parameter non-destruktif, kimia dan destruktif (Pathaveerat *et al*., 2008), ekstraksi fitur warna buah (Asnor *et al*., 2013; Bakar *et al*., 2013; Ullah *et al*., 2018), segmentasi (Nawawi dan Ismail, 2017), Viola-Jones (Rahman, 2016) dan model kulit nanas dengan *Active Shape Model* (ASM) (Kaewapichai *et al*., 2007). Beberapa metode tersebut masih memiliki kelemahan, seperti teknik yang dilakukan Pathaveera *et al*., (2008), yaitu pengukuran parameter destruktif, dengan pengukuran kematangan buah nanas merusak buah tersebut karena peneliti harus membelah untuk diidentifikasi. Pada teknik Viola-Jones yang dilakukan Rahman (2016), teknik tersebut membutuhkan banyak sampel yang digunakan selama proses pelatihan. Model *Active Shape Model* (ASM) yang dilakukan Kaewapichai *et al*., (2007), membutuhkan data set yang cukup banyak guna pemodelan buah nanas. Ekstraksi warna buah yang dilakukan Asnor *et al*., (2013); Bakar *et al*., (2013); Ullah *et al*., (2018) dan teknik segmentasi yang dilakukan Nawawi dan Ismail (2017), yang hanya mendeteksi kulit nanas secara umum sehingga warna dari setiap mata pada nanas tidak terdeteksi dengan maksimal (Yanti, *et al*., 2022). Oleh karena itu, diperlukan teknik yang baru untuk mengetahui kuantifikasi persentase kematangan buah nanas agar diperoleh hasil deteksi kematangan yang lebih akurat.

Penelitian ini diusulkan untuk mengembangkan teknik pendeteksi kematangan nanas yang telah dilakukan Yanti *et al*., (2022), ia menggunakan teknik *thresholding* untuk mengukur persentase kematangan pada nanas. Metode tersebut masih memiliki kelemahan dimana dapat diketahui *thresholding* atau dapat disebut juga dengan proses ambang batas nilai pixel pada citra namun untuk nilai ambangnya sendiri harus diberikan secara manual atau dilakukan secara coba-coba, sehingga dalam proses pengambilan citra memakan waktu yang cukup lama (Yanti *et al*., 2022).

Pada penelitian kali ini menggunakan metode *Convolutional Neural Network* (CNN). Dengan metode ini, akan diketahui persentase kematangan buah nanas dengan melihat warna dari buah nanas itu sendiri. Pada metode CNN terdapat dua tahapan utama, yaitu *feature learning* dan *classification*. Tahapan *feature learning* terdiri dari *convolution layer* yang digunakan untuk proses mengkonvolusi pada *output*, dan pada tahapan *feature learning* terdapat juga *pooling layer* yang berfungsi untuk mengurangi spasial dari fitur konvolusi. Tahapan *feature learning* ini akan dilakukan secara berulang-ulang, tergantung jenis arsitektur CNN apa yang akan digunakan pada saat proses dilakukan. Adapun tahap utama pada CNN yang kedua, dimana lanjutan dari tahapan *feature learning*, yaitu *classification* yang terdiri dari *fully connected layer* yang merupakan lapisan yang semua neuron aktivasi dari lapisan sebelumnya telah terhubung dengan neuron di lapisan selanjutnya (Purba *et* al., 2022). Dari dua tahapan utama tersebut, maka akan dihasilkan kuantifikasi persentase tingkat kematangan pada buah nanas.

1.2. Rumusan Masalah

Rumusan masalah pada penelitian ini yaitu, bagaimana cara mendeteksi identifikasi kematangan buah nanas dengan menggunakan metode *Convolutional Neural Network* (CNN)?

1.3. Tujuan Penelitian

Tujuan dilakukannya penelitian ini yaitu, dapat menggunakan metode deteksi kematangan buah nanas dengan mengaplikasikan teknik *Convolutional Neural Network* (CNN).

1.4. Manfaat Penelitian

Manfaat pada penelitian ini adalah sebagai berikut.

1. Dapat menghasilkan suatu cara untuk melakukan identifikasi kematangan buah nanas dengan bantuan teknologi.
2. Dihasilkan suatu metode untuk mengetahui kuantifikasi kematangan pada buah nanas dengan menggunakan teknik *Convolutional Neural Network* (CNN) dengan algoritma *Visual Geometry Group* (VGG) 16.

1.5. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut.

1. Pengambil sampel citra menggunakan *Webcam* Logitech C922 Pro HD Stream dengan resolusi kamera 1080p/30fps dan 720p/60fps.
2. Metode yang digunakan adalah *Convolutional Neural Network* (CNN).
3. Buah nanas yang digunakan jenis *Queen*.
4. Menggunakan aplikasi python sebagai bahasa pemrograman.
5. Arsitektur *Convolutional Neural Network* (CNN) yang digunakan adalah model *Visual Geometry Group* (VGG) 16.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terkait

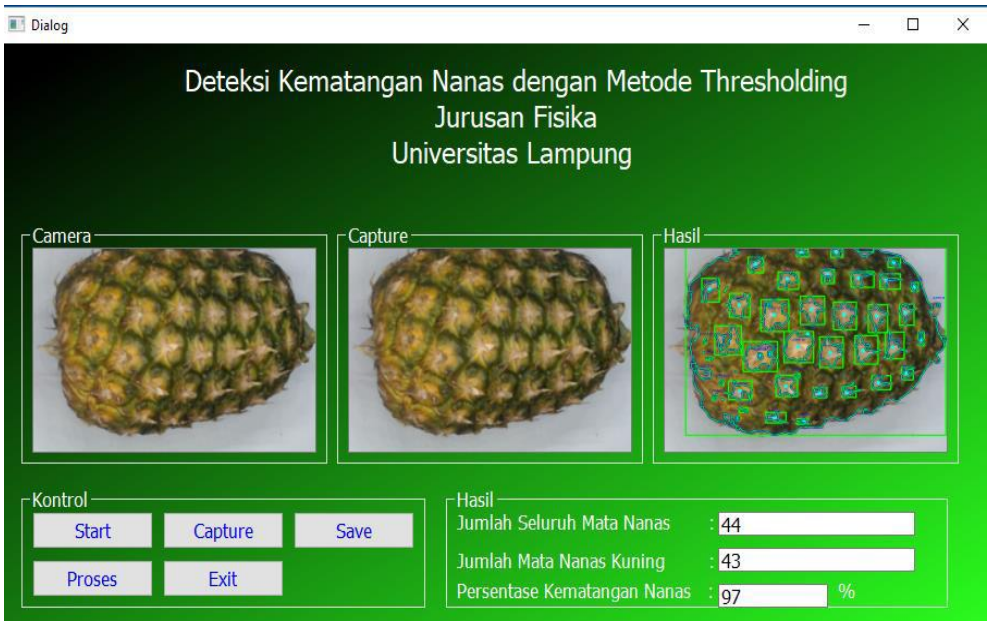
Penelitian mengenai rancang bangun alat deteksi kematangan buah nanas berdasarkan tingkat kekuningan mata nanas dengan menggunakan metode *thresholding* (Yanti *et al*., 2022). Peneliti menggunakan perangkat keras kotak deteksi kematangan buah nanas, yang terdiri dari beberapa alat lainnya, yaitu PC, *Webcam* dan *LED*. Adapun perangkat lunak menggunakan *python* sebagai bahasa pemrograman untuk mendeteksi kematangan buah nanas. Pada **Gambar 2.1** menunjukkan rancang bangun alat deteksi kematangan buah nanas.

Gambar 2.1 Alat deteksi kematangan buah nanas menggunakan metode *thresholding* (Yanti *et al*., 2021)

Perancangan perangkat keras tingkat kematangan buah nanas terdiri dari sebuah kotak yang terbuat dari duplex berwarna putih berukuran 45 cm x 45 cm x 50 cm sebagai tempat nanas, *LED* sebagai sumber penerangan, lalu ada *Webcam* ditempatkan di atas kotak yang berfungsi untuk mengambil citra buah nanas dan

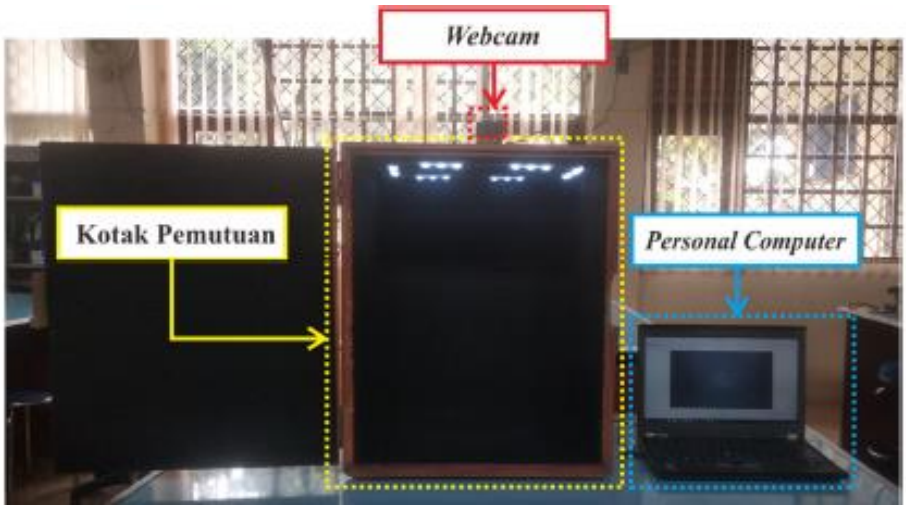
PC untuk mengolah citra serta mengidentifikasikan tingkat kematangan buah nanas. Pada percobaan ini peneliti menggunakan metode *thresholding* untuk mendeteksi kematangan nanas dengan menghitung jumlah seluruh mata nanas dan menghitung jumlah mata nanas berwarna kuning. Pengambilan sampel nanas yang didapat dari *Webcam* akan diproses menggunakan pengolahan citra. Proses ini dilakukan menggunakan *webcam* yang telah dikontrol menggunakan GUI, gambar yang dihasilkan memiliki format BGR, lalu akan dikonversi ke RGB, setelah itu gambar akan masuk tahap segmentasi citra.

Proses segmentasi citra berguna untuk mengkonversi dan me*resize* citra yang masih memiliki noise sehingga perlu di *smoothing* agar mengurangi noise tersebut, selanjutnya akan dilakukan *thresholding* untuk memisahkan antara mata nanas dan *background*nya. Mata nanas yang telah dipisahkan dari *background*nya ternyata masih ada yang menyatu atau tumpang tindih sehingga diperlukan pemisahan dengan dilakukannya *dilation.* Setelah itu, citra yang sudah melakukan beberapa tahap segmentasi citra maka akan dilakukan *preprocessing* selanjutnya, yaitu seleksi kontur. Citra akan di kontur pada bagian mata nanas, agar mata nanas yang terdeteksi dapat dilihat, lalu dilakukannya perhitungan jumlah seluruh mata nanas dan jumlah mata nanas berwarna kuning. Hasil deteksi akan ditampilan pada panel GUI, dapat dilihat pada **Gambar 2.2**.



Gambar 2.2 Tampilan pada GUI hasil deteksi kematangan nanas menggunakan metode *thresholding* (Yanti *et al*., 2022)

Prasetyo *et al*., (2021) melakukan rancang bangun sistem identifikasi tingkat kematangan buah nanas secara non-destruktif berbasiskan *computer vision.* Dengan metode ini peneliti melakukan pengumpulan data citra buah nanas terlebih dahulu dengan berbagai tingkat kematangan sebagai dataset. Setelah itu, akan dilanjutkan ke tahap segmentasi citra, tahap ini dilakukan untuk mengkonversi citra menjadi *grayscale,* lalu di *thresholding* dengan metode Otsu sehingga me*resize* bagian buah dan masuk ke tahap *masking* area buah dengan *ellipse mask* untuk mendapatkan hasil segmentasi citra. Setelah proses segmentasi cita maka dilanjutkan dengan mengekstrak citra citra untuk memisahkan citra ke bentuk *channel* RGB ke citra *HSV* dan akan menghitungkan nilai *mean, varian, standar deviasi, skewness,* dan *kurtosis* pada *channel* RGB, lalu citra akan disimpan ke ekstraksi fitur ke format csv. Terakhir dilakukannya seleksi ciri bertujuan untuk menggambarkan hubungan antara ciri warna dengan indeks tingkat kematangan nanas. Peneliti juga melakukan *preprocessing* yang dilanjutkan dengan melatih data menggunakan jaringan syaraf tiruan (JST), pelatihan jaringan syaraf tiruan dengan *input* berupa ciri yang diperoleh dari hasil seleksi citra yang telah dilakukan sebelumnya. Hasil yang diperoleh dari penelitian ini, yaitu tingkat keakuratan kematangan pada buah nanas adalah sebesar 98,4% dengan menggunakan metode Non-Destruktif Berbasis Computer Vision.

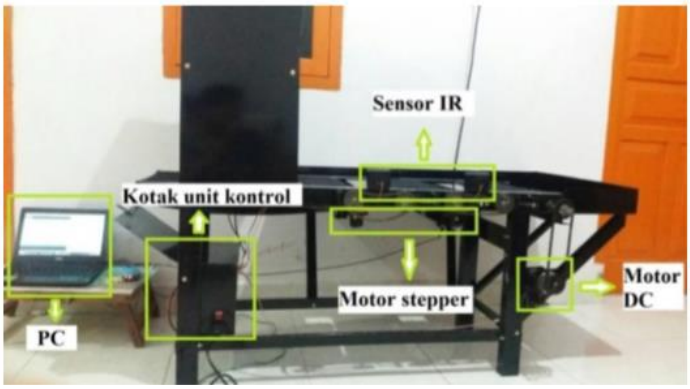


Gambar 2.3 Realisasi sistem pemutuan (Prasetyo *et al*., 2021)

Desi *et al*., (2020) melakukan rancang bangun alat purwarupa pemilah nanas yang berdasarkan tingkat kematangan menggunakan mikrokontroler *Blue Pill*

STM32F103C8T6 berbasis *computer vision*. Tujuan dari merancang bangun alat purwarupa ini, yaitu untuk memilih nanas dengan klasifikasi mentah, matang dan sangat matang. Metode yang dilakukan oleh peneliti ini untuk melakukan perancangan pemilahan buah nanas, yaitu menggunakan sistem mekanik dan juga rangkaian elektronika yang digunakan untuk menjalankan alat purwarupa tersebut secara otomatis (Yanti *et al*., 2022).

Peneliti menjalankan beberapa proses untuk pemilahan buah nanas, yaitu dengan proses simulasi menggunakan data serial indeks kematangan yang telah dikirimkan dari PC ke mikrokontroler melalui komunikasi serial. *Conveyor* yang digunakan untuk proses pemilahan nanas, yaitu menggunakan sistem transmisi rantai, roda gigi dan juga penggerak berupa motor DC (Yanti *et al*., 2022). Pada **Gambar 2.4** menunjukkan alat purwarupa pemilah buah nanas.



Gambar 2.4 Purwarupa pemilah nanas (Desy *et al*., 2020)

2.2. Buah Nanas

Buah nanas merupakan salah satu komoditas buah tropis di Indonesia. Indonesia merupakan wilayah yang beriklim tropis dan berada di daerah khatulistiwa. Indonesia memungkinkan tumbuhnya berbagai macam tumbuhan dengan subur seperti buah-buahan, salah satunya adalah buah nanas. Jenis nanas yang banyak tumbuhan di Indonesia adalah jenis nanas *Queen* dan *Cayenne* (Delfi, 2021).

Dalam famili nanas terdapat 60 genus dengan sekitar 1.500 spesies, berikut adalah klasifikasi nanas.

Kingdom : *Plantae* (tumbuh-tumbuhan)

Divisi : *Spermatophyta* (tumbuhan berbiji)
Sub divisi : *Angiospermae* (berbiji tertutup)
Ordo : *Farinosae* (*Bromeliales*)
Kelas : *Liliopsida* (*Monokotil* berdaun lembaga daun)
Famili : *Bromeliaceae*
Genus : *Ananas*
Spesies : *Ananas comosus* (L. *Merr*)

Nanas merupakan tanaman yang memiliki tinggi 1-2 m, memiliki diameter kurang lebih 1,5 m. Tanaman ini memiliki batang, tetapi berkayu. Batang utama nanas di bawah buah disebut juga dengan istilah *butt*. Batang tanaman nanas berbentuk gada dengan panjang kurang lebih 20-30 cm. Diameter bawah kisaran 2-3,5 cm, sedangkan bagian tengah nanas kurang lebih 5,5-6,5 cm. Pada bagian atas biasanya terlihat lebih kecil. Batang memiliki ruas pendek yang akan terlihat jika daunnya dilepas. Tetapi hal ini tergantung dari varietas nanas. Daun nanas berbentuk panjang dan sedikit sempit atau kecil. Ujung nanas memanjang, runcing, permukaan atas berwarna hijau tua, merah tua dan bergaris atau coklat kemerahan. Lebar daun kurang lebih 6 cm dan panjang bisa mencapai 90 cm. Sebagian daun nanas memiliki duri pada bagian ujung, tampak warna putih seperti ketombe pada daun yang disebabkan adanya rambut-rambut bersel yang disebut *trichome*. Hal ini tergantung dari varietasnya. Nanas memiliki akar serabut, dangkal dan tersebar luas. Akar akan tumbuh dari batang, lalu masuk ke ruang batang dan daun. Nanas juga memiliki bunga yang terletak tegak lurus pada tangkai buah dan berkembang menjadi buah majemuk. Sifat bunga adalah hermaprodit dengan jumlah 100-200 yang masing-masing berada di ketiak daun pelindung. Adapun biji nanas yang berukuran kecil dengan panjang sekitar 2-4 mm dan lebar 1-2 mm. Adapun tunas pada nanas, yaitu tangkai buah, tunas yang muncul dari ketiak daun batang dan tunas yang muncul karena anakan. Tunas-tunas inilah yang nantinya akan dijadikan sebagai perbanyakan vegetatif tanaman nanas (Lubis, 2020).

Gambar 2.5 Buah nanas (Kompas.com, 2021)

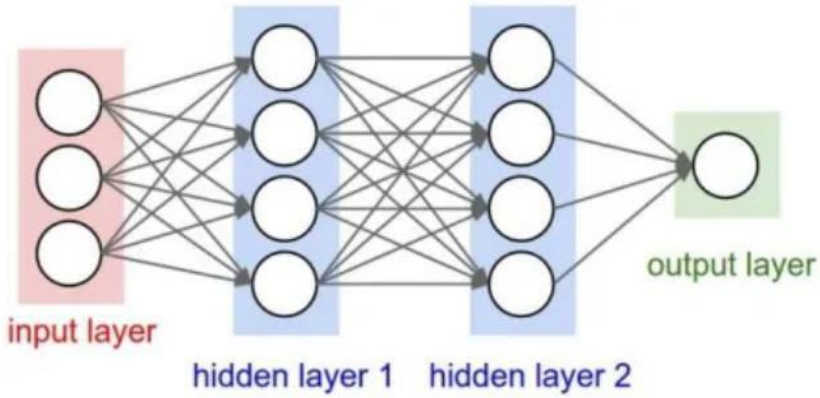
Nanas memiliki rasa manis yang unik dan segar, sehingga banyak dikonsumsi oleh masyarakat Indonesia, berupa jus buah dan buah-buahan kaleng. Komponen aroma utama buah nanas adalah terpen, keton, aldehid dan ester (Delfi, 2021). Nanas mengandung serat yang berguna untuk membantu proses pencernaan, sekitar setengah dari protein dalam nanas mengandung *protease* bromelin. Nanas memiliki gizi dan kandungan yang sangat baik dalam 100 gram diantaranya, yaitu mengandung zat besi 0,28 milligram, energi 50 kkal, fosfor 8 milligram, gula 9,26 gram, karbohidrat 12,63 gram, kalsium 13 milligram, lemak 0,12 gram, magnesium 12 milligram, serat 1,40 gram, protein 0,54 gram, potassium 115 milligram, thiamin (Vit. B1) 0,079 milligram, riboflavin (vit. B2) 0,031 milligram, niacin (Vit. B3) 0,489 milligram, pantothenic acid (B5) 0,205 milligram, vitamin B6 0,110 milligram, folate (Vit. B9) 15 milligram, vitamin C 36,2 milligram (Lubis, 2020).

Buah nanas termasuk salah satu jenis buah yang bernilai ekonomis tinggi. Selain dikonsumsi dalam bentuk segar (tanpa diolah), nanas juga bisa dikonsumsi dalam bentuk yang sudah dalam olahan menjadi berbagai bentuk olahan makanan dan minuman. Rasa manis dan asam yang dihasilkan oleh nanas sangat disukai oleh masyarakat dan juga nanas memiliki kandungan gizi yang sangat tinggi sehingga baik untuk tubuh dan kesehatan (Lubis, 2020). Kandungan yang ada di dalam nanas terdiri atas enzim bromelin. Enzim bromelin merupakan unsur pokok dari nanas yang sangat bermanfaat dalam bidang farmasi dan makanan. Enzim bromelin mirip dengan papain dan fisik, yang berfungsi sebagai pemecah protein (Silaban dan Rahmanisa, 2016).

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu metode klasifikasi yang termasuk ke dalam kelompok *deep learning* yang menggunakan *layer* konvolusi untuk mengkonvolusi suatu *input* dengan filter (Yusuf *et al*., 2019). CNN merupakan pengembangan dari *Multi Layer Perceptron* (MLP). CNN juga memiliki kemampuan untuk mempelajari fitur secara *unsupervised* (Azman *et al*., 2017).

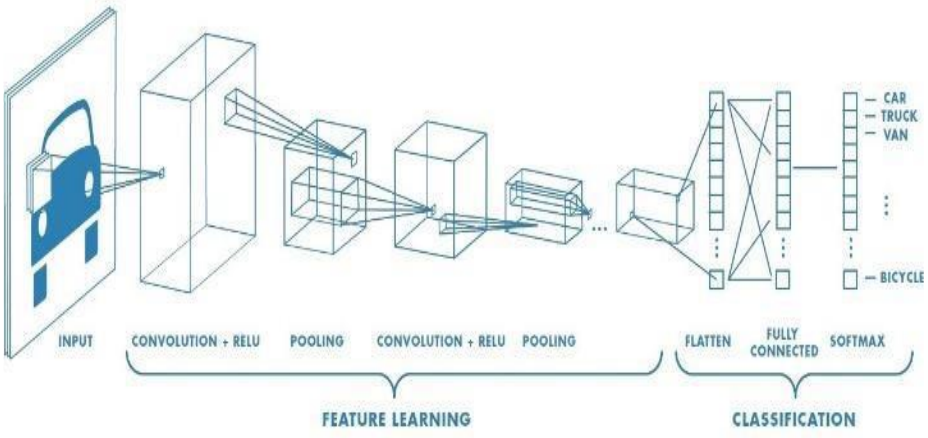
Sebuah MLP memiliki 1 layer dengan masing-masing layer berisi neuron. MLP menerima *input* data satu dimensi yang akan meneruskan data tersebut ke jaringan selanjutnya yang akan menghasilkan *output*. Operasi linear pada MLP dilakukan dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non-linear yang biasa disebut sebagai fungsi aktivasi. Sedangkan pada CNN, data yang diteruskan ke jaringan selanjutnya adalah data dua dimensi. Operasi linear pada CNN menggunakan operasi konvolusi, sedangkan bobot tidak satu dimensi, namun akan berbentuk empat dimensi yang merupakan kernel konvolusi (Putra, 2016). Seperti pada **Gambar 2.6**.



Gambar 2.6 Arsitektur MLP sederhana (Putra, 2016)

Teknik pengolahan citra telah dilakukan dengan menggunakan berbagai metode, contohnya jaringan syaraf tiruan, segmentasi citra maupun *thresholding* yang telah dilakukan penelitian sebelumnya Yanti *et al.*, (2022) dan sebagainya. Dari penelitian ini akan menggunakan metode *Convolutional Neural* Network (CNN). Yang membedakan metode ini dengan metode *machine learning* lainnya adalah,

yaitu membutuhkan fitur yang sebelumnya harus ditentukan (Azman *et al*., 2017). CNN sendiri terdiri dari dua tahapan utama, yaitu *feature learning* dan *classification.* Pada tahapan *feature learning* terdiri dari *convolution layer* dan *pooling layer* sedangkan pada tahap *classification* terdiri dari *fully connected layer* (Yusuf *et al*., 2019).

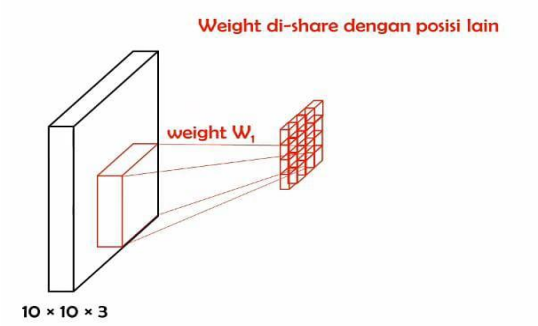


Gambar 2.7 Arsitektur CNN (Trivusi, 2022)

Input gambar pada model CNN menggunakan citra yang berukuran 64x64x3. Angka tiga yang dimaksud adalah 3 *channel* yaitu *red, green* dan *blue* (RGB). Citra masukan kemudian diproses pada tahapan *feature learning*, jumlah proses konvolusi pada rancangan ini memiliki dua lapisan konvolusi, dimana setiap konvolusi memiliki jumlah ukuran *kernel* yang berbeda (Saputra *et al*., 2020). Tahapan bagian dari *convolutional neural network* (CNN), diantaranya *convolution layer, pooling layer* dan *fully connected layer*.

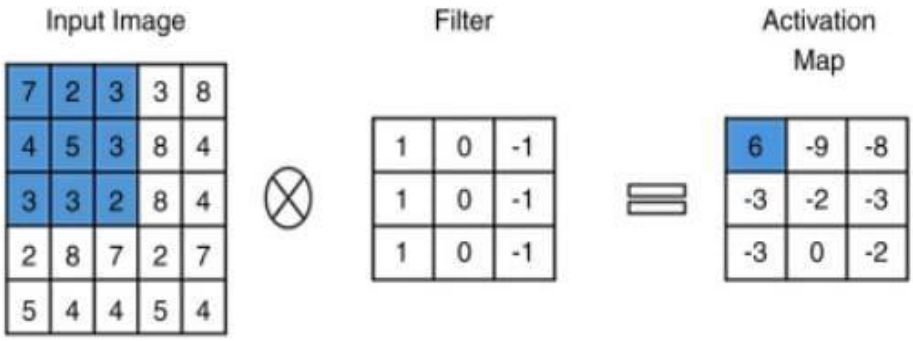
2.3.1. Convolution Layer

Tahap ini melakukan operasi konvolusi pada *output* dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari jaringan arsitektur CNN (Sena, 2017). *Convolution Layer* merupakan lapisan yang digunakan untuk melakukan operasi konvolusi dari *output* lapisan sebelumnya. **Gambar 2.8** menunjukkan input layer pada tahap *convolution layer*.



Gambar 2.8 *input* secara lokal (Astrid, 2019)

Convolution Layer adalah bangunan utama pada CNN, yang terdiri dari satu set *kernel*. Ukuran *kernel* biasanya lebih kecil dari gambar sebenarnya. Untuk konvolusi *kernel* meluncur melintasi tinggi dan lebar gambar, yang akan melakukan perkalian titik antara setiap elemen *kernel* dan pada *input* akan dihitung setiap posisi spasialnya (Mostafa dan Wu, 2021). Pada **Gambar 2.9** menunjukkan contoh proses konvolusi.



Gambar 2.9 Contoh grafis proses *convolution layer* (Mostafa dan Wu, 2021)

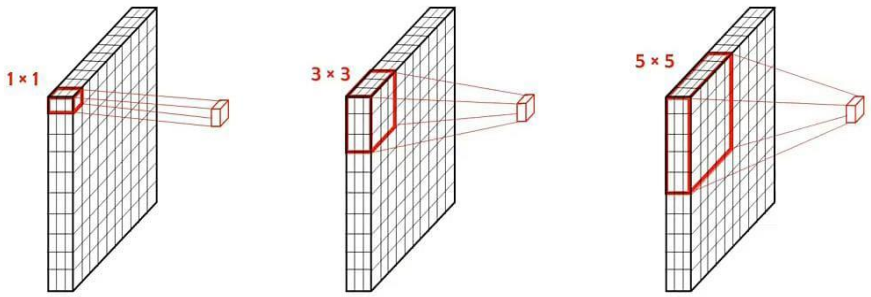
Adapun contoh perhitungannya pada gambar proses *convolutional layer* sebagai berikut.

$$7*1+2*0+3*(-1)+4*1+5*0+3*(-1)+3*1+3*0+2*(-1) = 6 \quad (2.1)$$

2.3.1.1. Kernel

Kernel biasa disebut juga sebagai konvolusi filter, kernel konvolusi dinyatakan dalam bentuk matriks (umumnya 1 x 1, 3 x 3, atau 5 x 5) ukuran matriks ini biasanya lebih kecil dari ukuran citra aslinya dan untuk nilai pada kernel biasa disebut

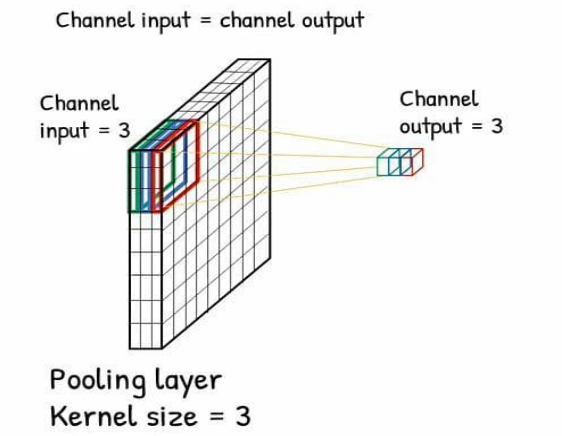
sebagai koefisien (Astrid, 2019). Pada **Gambar 2.10** menunjukkan proses perkalian *kernel* dengan *input* citra.



Gambar 2.10 Contoh ukuran *kernel* (Astrid, 2019)

2.3.2. Pooling Layer

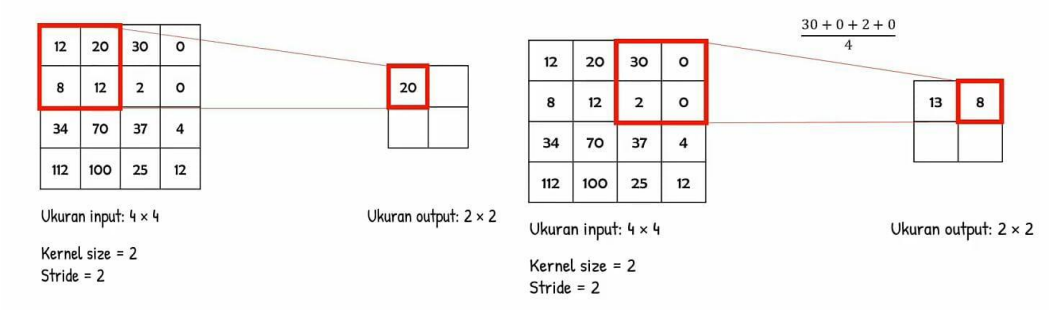
Pooling layer merupakan lapisan yang berfungsi untuk mengurangi *size* spasial dari fitur konvolusi (mengurangi dimensi), yang dapat mengurangi sumber daya komputasi yang terlalu banyak (Astrid, 2019). Pada **Gambar 2.11** menunjukkan proses pada *pooling layer*.



Gambar 2.11 *Pooling layer* (Astrid, 2019)

Pooling layer dapat memproses setiap *channel input* secara terpisah, maka pada *pooling* jumlah *channel input* dan *channel output* sama. *Pooling layer* terbagi menjadi dua macam , yaitu *max pooling* dan *average pooling*. *Max pooling* akan memilih nilai maksimum dari setiap area interlokal yang dilihat, sedangkan *average pooling* akan merata-rata semua nilai area lokal tersebut, sehingga akan

meneruskan semua *input* sama rata (Astrid, 2019). Pada **Gambar 2.12** menunjukkan proses *max pooling* dan *average pooling*.

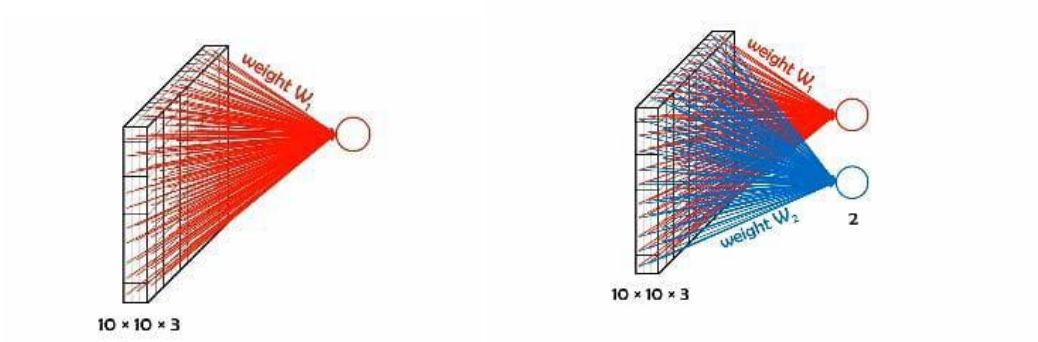


(a) **(b)**

Gambar 2.12 (a) *Max pooling*; (b) *average pooling* (Astrid, 2019)

2.3.3. Fully Connected Layer

Fully Connected Layer merupakan *layer*/lapisan yang dimana semua neuron aktivitas dari lapisan sebelumnya telah terhubung semua dengan neuron di lapisan selanjutnya seperti pada jaringan syaraf tiruan. *Fully Connected Layer* tersebut adalah layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear (Astrid, 2019). Pada **Gambar 2.13** menunjukkan proses dari *fully connected layer*.



(a) **(b)**

Gambar 2.13 (a) FC 1 *output*; (b) FC 2 *output* (Astrid, 2019)

Fully Connected Layer berfungsi untuk mengolah data sehingga bisa mengklasifikasi berdasarkan *future* yang diekstrak. *Fully Connected Layer* akan mengkoneksi setiap *pixel* pada gambar termasuk dari 3 *chanel* warna untuk setiap

output. Setiap neuron pada *Convolution Layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *Fully Connected Layer* (Astrid, 2019). *Fully Connected Layer* merupakan neuron buatan yang akan mengkomputasi perhitungan berdasarkan unit yang terhubung. Pada neuron buatan tunggal, neuron langsung terhubung ke deskripsi *input* dari objek yang ingin di ekstrak *input*nya. Berdasarkan nilainya, neuron akan memutuskan ada atau tidak ada beberapa karakteristik yang hadir dalam *input* tersebut (Malhotra, 2018).

Terdapat 2 langkah untuk perhitungan pada *fully connected*, yaitu sebagai berikut.

a. Pra-Aktivasi Neuron (aktivasi *input*)

Dalam bentuk skalar,

$$a(x) = b + \sum_i w_i x_i \tag{2.2}$$

Dalam bentuk vektor,

$$a(x) = b + w^T x \tag{2.3}$$

Dengan *a* adalah fungsi pra-aktivasi, *x* adalah *input* vektor, *w* adalah vektor bobot koneksi. Ini mewakili kekuatan antar koneksi, *w* adalah vektor bobot koneksi. Ini mewakili kekuatan antar koneksi dan *b* adalah bias.

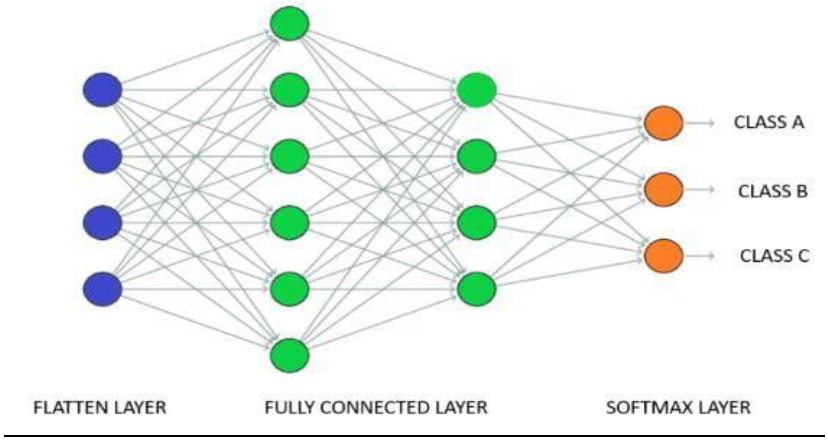
Jika tidak memiliki *input*, maka b akan menjadi *input* untuk neuron. Biar adalah nilai negatif dari ambang batas, ambang batas berarti nilai batas yang harus memiliki fungsi aktivasi yang akan berfungsi untuk masuk ke kategori lain.

b. Aktivasi Neuron (aktivasi *output*)

Menggunakan nilai-nilai dari fungsi pra-aktivasi untuk menghitung aktivasi (Malhotra, 2018).

$$H(x) = g(a(x)) = h(b+w^T x) \tag{2.4}$$

Dengan *(x)* adalah hasil aktivasi dan *g(x)* adalah fungsi aktivasi.

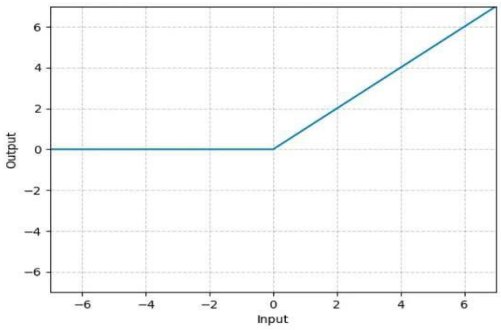


Gambar 2.14 Klasifikasi CNN (Indian, 2021)

Fully Connected Layer adalah lapisan terakhir dalam jaringan syaraf konvolusi. Lapisan *Fully Connected Layer* mengambil *input* dari *flatten layer* dimana lapisan satu dimensi. Data *flatten layer* terlebih dahulu melewati fungsi *affine* dan kemudian ke fungsi non-linear disebut sebagai 1 FC (*fully connected*) atau lapisan tersembunyi. Keluaran dari lapisan tersembunyi terakhir dikirim ke fungsi *softmax* atau *sigmoid* untuk distribusi probabilitas pada kumpulan akhir dari jumlah total kelas (Indian, 2021).

2.3.4. Rectified Linear Unit (ReLU)

ReLU adalah cara paling umum dan dasar untuk aktivasi memperkenalkan non-linearitas ke model jaringan syaraf. ReLU juga merupakan hasil dari penjumlahan dari konvolusi. Fungsi ReLU adalah fungsi linier sepotong yang akan mengeluarkan *input* secara langsung, jika positif maka *output*nya akan ">0", jika *input*nya negatif maka *output*nya akan menghasilkan "0" (Patel, 2019). Pada **Gambar 2.15** menunjukkan ilustrasi ReLU.

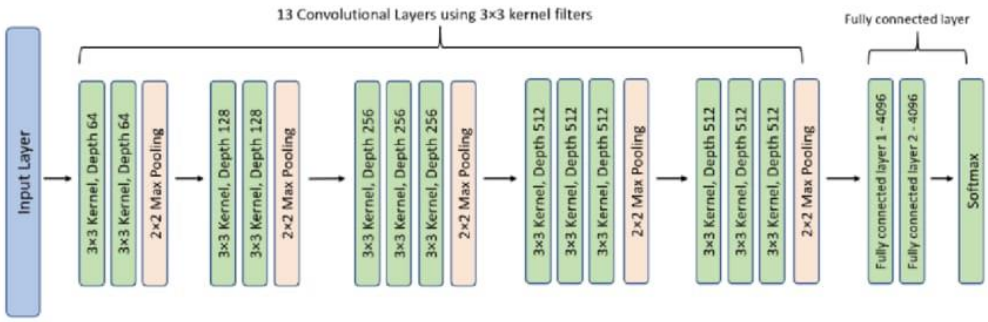


Gambar 2.15 Contoh ReLU (Patel, 2019)

2.4. Arsitektur Model *Visual Geometry Group* (VGG) 16

Arsitektur *deep learning* yang digunakan dalam klasifikasi citra salah satunya adalah *Visual Geometry Group* (VGG) dengan 16 lapisan terdiri dari kombinasi *convolutional layer*, *max pooling* dan *fully connected layer*. Klasifikasi citra adalah satu aplikasi penting dalam bidang pengolahan citra, yang bertujuan untuk mengklasifikasikan citra menjadi beberapa kelas berdasarkan ciri-ciri visual yang terdapat pada citra tersebut (Wicaksono dan Henri, 2023).

Arsitektur VGG16 terbukti sangat efektif dalam melakukan klasifikasi citra pada berbagai dataset citra seperti dataset *ImageNet*. Arsitektur VGG16 merupakan top ke 5 dalam tantangan *ImageNet* pada tahun 2014 yang memperoleh akurasi sebesar 92,6% dengan kumpulan data lebih dari 14 juta gambar yang termasuk dalam 1000 kelas (Simonyan dan Zisserman, 2015). Meskipun arsitektur VGG16 memiliki performa yang baik dalam klasifikasi citra, terdapat beberapa keterbatasan yang perlu diperhatikan. Seperti, arsitektur VGG16 memiliki jumlah parameter yang sangat banyak, sehingga membutuhkan waktu dan sumber daya komputasi yang besar dalam melakukan pelatihan dan yang diperhatikan juga pada arsitektur VGG16 mungkin mengalami *overfitting* apabila dataset pelatihan terlalu kecil (Wicaksono dan Henri, 2023). Pada **Gambar 2.16** menunjukkan tahapan arsitektur VGG16.



Gambar 2.16 Arsitektur model VGG16 (Rismiyati dan Ardytha, 2021)

Pada **Gambar 2.16** terdapat 13 konvolusi layer, kernel *size* 3x3, max pooling 2x2 dan *fully connected layer* menggunakan fungsi aktivasi *softmax*. Pada layer ke 1 dan 2 terdapat 64 bit, maksudnya adalah pada layer pertama dan kedua memiliki kernel citra input dengan matrix sebesar 64x64. Layer selanjutnya yaitu, layer 3 dan

4 memiliki 128 bit pada citra input. Layer 5 sampai layer 7 memiliki 256 bit. Layer 8 sampai layer 13 memiliki 512 bit (Rismiyati dan Ardytha, 2021).

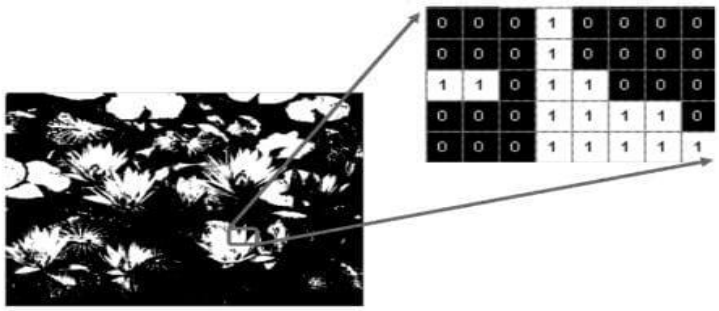
Hasil dari tahap konvolusi dilanjutkan dengan proses *fully connected layer* yang menggunakan fungsi aktivasi *softmax*. Umumnya *softmax layer* menggunakan generalisasi *multiclass* dari *regresi* jumlah data yang digunakan atau dikenal sebagai fungsi *softmax* yang terhubung setelah *fully connected layer. Softmax* juga bertujuan untuk meningkatkan probabilitas dalam klasifikasi kelas (jenis citra nanas) (Candra dan Aditya, 2020).

2.5. Citra

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue)* dari intensitas cahaya pada bidang dwimatra, sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini dilengkapi oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner)* dan sebagainya. Sehingga bayangan objek yang disebut terakam. Citra juga merupakan objek tiruan dari suatu benda atau gambar. Dapat diketahui citra sendiri terbagi menjadi dua bagian yaitu citra analog dan citra digital. Citra analog adalah citra yang dihasilkan oleh sinyal *kontinu*, misalnya pada cetakan foto pada kertas foto, termasuk juga citra yang biasa tampil dalam layar Televisi dan lain-lain (Irtawaty dan Jayanti *et al*., 2016). Sedangkan citra digital terbagi menjadi 3 bagian yaitu citra biner, citra *grayscale* dan citra warna.

2.5.1. Citra biner

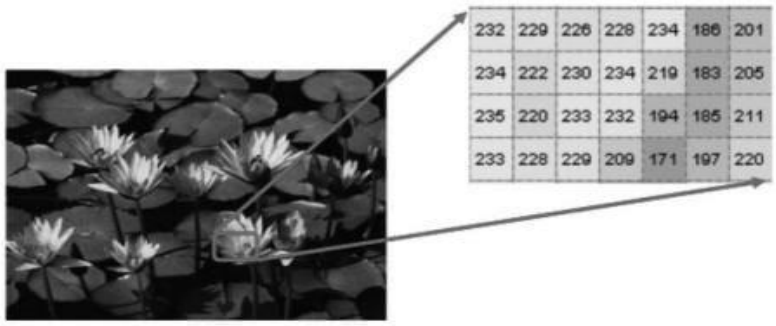
Citra biner adalah citra yang terdiri dari tiap-tiap piksel berwarna yang hanya membutuhkan 1 bit memori. Maka, setiap piksel hanya mempunyai dua nilai intensitas, yaitu 1 atau 0, atau dalam bentuk warna biasanya putih dan hitam (Andono dan Sutojo, 2017). **Gambar 2.17** menunjukkan gambar citra biner.



Gambar 2.17 Citra biner (Andono dan Sutojo, 2017)

2.5.2. Citra *grayscale*

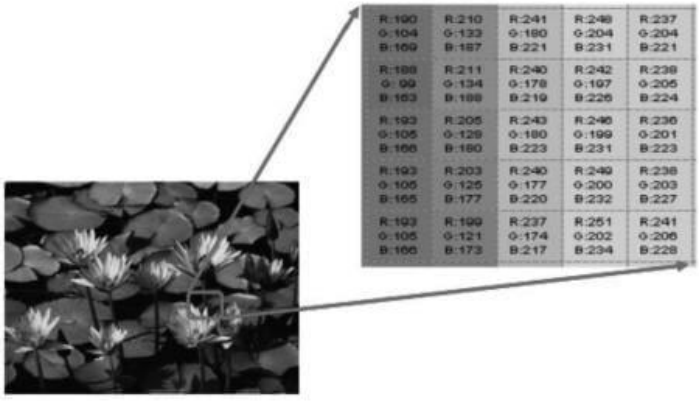
Citra *grayscale* adalah matriks data yang nilainya mewakili intensitas setiap piksel berkisar 0 sampai dengan 255. Setiap piksel membutuhkan 8 bit memori (Andono dan Sutojo, 2017). **Gambar 2.18** menunjukkan gambar citra *grayscale*.



Gambar 2.18 Citra *grayscale* (Andono dan Sutojo, 2017)

2.5.3. Citra warna

Citra warna adalah citra yang masing-masing pikselnya mempunyai tiga komponen warna yang spesifik, yaitu komponen merah (*red*), hijau (*green*) dan biru (*blue*). Warna setiap piksel ditentukan oleh kombinasi dari intensitas warna merah, hijau dan biru yang akan disimpan pada bidang warna di lokasi piksel. Pada format file grafis menyimpan citra warna sebanyak 24 bit, yang berasal dari komponen merah hijau dan biru yang masing-masing membutuhkan 8 bit. Karena pada citra warna mempunyai 24 juta kemungkinan warna (Andono dan Sutojo, 2017). **Gambar 2.19** menunjukkan gambar citra warna.



Gambar 2.19 Citra warna (Andono dan Sutojo, 2017)

Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran *M* baris dan *N* kolom, dengan *x* dan *y* adalah koordinat $f(x,y)$ dinamakan intensitas atau tingkat keabuan citra pada titik tersebut. Apabila nilai x,y dan nilai amplitudo *f* secara keseluruhan berhingga dan bernilai diskrit, maka dapat dikatakan bahwa citra tersebut adalah citra digital. Citra digital dapat dituliskan dalam bentuk matrik sesuai dengan Persamaan dibawah ini:

$$f(x,y) = \begin{pmatrix} f(0,0) & \cdots & f(0,M-1) \\ \vdots & \cdots & \vdots \\ f(N-1,0) & \cdots & f(N-1,M-1) \end{pmatrix} \qquad (2.5)$$

Nilai pada suatu irisan antara baris dan kolom (pada porsi *x*, *y*) disebut dengan *picture elements*, *images element*, *pixel*. Istilah piksel adalah yang paling sering digunakan (Kusumanto dan Tompunu, 2011)

2.6. RGB

Segmentasi warna, ada bermacam-macam model warna. Model warna RGB (*red*, *green* dan *blue)* merupakan model yang banyak digunakan, salah satunya adalah monitor. Pada model ini untuk mempresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain model RGB terdapat juga model normalisasi RGB dimana model ini terdapat 3 komponen, yaitu *r, g dan b* yang mempresentasikan persentase dari sebuah piksel pada citra digital (Kusumanto dan Tompunu, 2011). Nilai-nilai tersebut mengikuti persamaan dibawah ini:

$$r = \frac{R}{R+GB}, g = \frac{G}{R+GB}, b = \frac{B}{R+G+B} \qquad (2.6)$$

sehingga, $r+g+b=1$ (2.7)

Biasanya, warna RGB digunakan untuk grafis komputer. Ia terdiri dari tiga warna, yaitu merah, hijau dan biru. Tiga warna ini digambarkan lewat angka 0 hingga 1. Angka-angka ini bisa membentuk persamaan matematika. Oleh karena itu, gabungan ketiganya bisa digambarkan lewat garis tiga dimensi. Garis ini akan membentuk kubus. Warna awal dalam RGB adalah hitam, ia berubah dengan menambahkan merah, hijau dan biru (Kusumanto dan Tompunu, 2011)

2.7. Artificial Intelligence

Artificial Intelligence (AI) merupakan bagian dari ilmu komputer yang mempelajari bagaimana menjadikan komputer dapat melakukan pekerjaan sebaik yang dilakukan manusia, bahkan bisa lebih baik. Sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan dengan baik oleh manusia. Seiring dengan perkembangan zaman, peran komputer semakin mendominasi kehidupan manusia (Pannu dan Tech, 2015).

2.7.1 Machine Learning

Machine learning pertama kali didefinisikan oleh Arthur Samual pada tahun 1959. Menurut Arthur Samuel, *machine learning* adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrograman yang jelas. *Machine learning* dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat prediksi yang akurat (Mohri, 2012).

2.8. Python

Pada awalnya kita akan beranggapan bahwa penamaan bahasa pemrograman ini didasarkan pada nama binatang melata, anggapan tersebut dapat dibilang salah. Penamaan bahasa pemrograman ini diilhami ketika pembuatnya menonton acara komedi di televisi di BBcC yang bernama Monty Python’s Flying Circus. Dapat diketahui pembuatan pemrograman ini adalah Guido van Rossum dari Amsterdam, Belanda. Yang pada awalnya, motivasi pembuatan bahasa pemrograman ini

merupakan untuk bahasa skrip tingkat tinggi pada sistem operasi industri Amoeba (Clinton dan Sengkey, 2019).

Python memiliki fitur perpustakaan yang luas. dalam distribusi Python telah disediakan modul yang siap pakai untuk berbagai keperluan. Python memiliki aturan *layout* kode sumber yang memudahkan pengecekan pembacaan kembali dan penulisan ulang kode sumber. Berorientasi objek memiliki sistem pengelolaan memori *automatic* (*garbage collection,* seperti *java*) modular, mudah dikembangkan dengan menciptakan modul baru, modul tersebut dapat didasari dengan bahasa Python maupun *C/C++*. Python memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman *java* dan python juga memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para programmer tidak perlu melakukan pengaturan ingatan komputer kembali dan akan secara langsung komputer mengingatnya. IDE untuk Python sendiri bisa didapatkan gratis dari web resminya (*http://python.org/*) (Clinton dan Sengkey, 2019).

Gambar 2.20 Gambar aplikasi python (Petanicode.com, 2018)

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter, interactive, object-oriented* dan dapat beroperasi hampir di semua platform : *Mac, Linux dan* Windows. python termasuk bahasa pemrograman yang mudah untuk dipelajari dan dan dipahami karena sintaks yang digunakan jelas dan juga dapat dikombinasikan dengan penggunaan modul siap pakai dan struktur data tingkat tinggi yang efisien. Distribusi *Python* dilengkapi dengan suatu fasilitas seperti *shell* di *Linux.* Lokasi penginstalan *Python* biasa terletak di "/usr/bin/Python". Menjalankan Python, cukup dengan mengetikan "*Python*", tunggu *Python* telah siap menerima perintah (Nurviyanto, 2012).

II. METODE PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Elektronika Dasar dan Instrumentasi, Jurusan Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung pada 20 November 2022 sampai dengan 13 Oktober 2023.

3.2. Alat dan Bahan

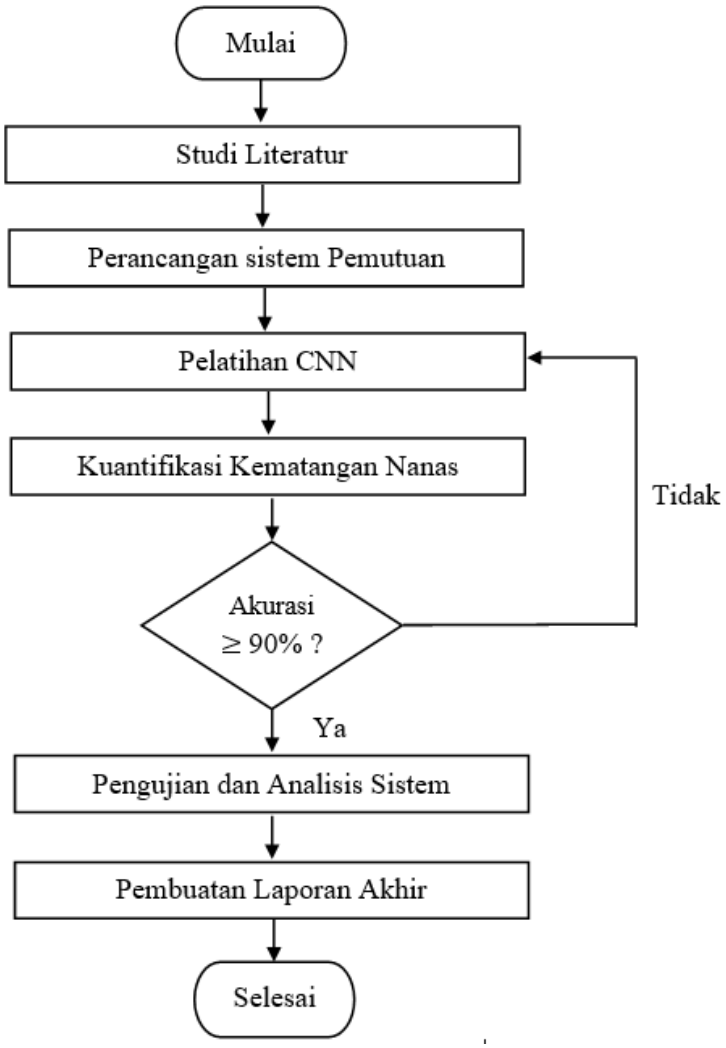
Alat dan bahan yang digunakan pada penelitian ini dapat dilihat pada **Tabel 3.1**.

Tabel 3.1 Alat dan Bahan Penelitian

Alat dan bahan	Fungsi
Personal Computer (PC)	Untuk mengolah program mendeteksi kematangan buah nanas.
Perangkat lunak *Python*	Sebagai program untuk mendeteksi kematangan buah nanas.
Webcam Logitech C922Pro	Untuk pengambilan gambar sampel buah nanas.
Kotak pengambilan citra	Tempat pengambilan gambar sampel buah nanas.
Lampu *LED* strip 117 SMD	Sebagai penerangan pada kotak pendeteksi kematangan nanas.
Protoboard	Sebagai konduktor sekaligus untuk meletakkan komponen-komponen.
Kabel jumper	Sebagai penghubung dari satu komponen ke komponen lainnya.
Potensiometer	Sebagai pembagi tegangan yang bisa disesuaikan.
Power supply (model S-60-12)	Untuk memberikan atau menyuplai arus listrik, yang tadinya arus berlawanan menjadi arus searah.
Colokan penghubung listrik	Untuk membantu menghidupkan *LED* yang akan disambungkan ke arus listrik.
Nanas Queen	Sebagai sampel uji kematangan nanas.

3.3. Prosedur Penelitian

Prosedur penelitian secara umum dapat dilihat pada diagram alir penelitian pada **Gambar 3.1**.



Gambar 3.1 Diagram alir penelitian

3.3.1. Studi Literatur

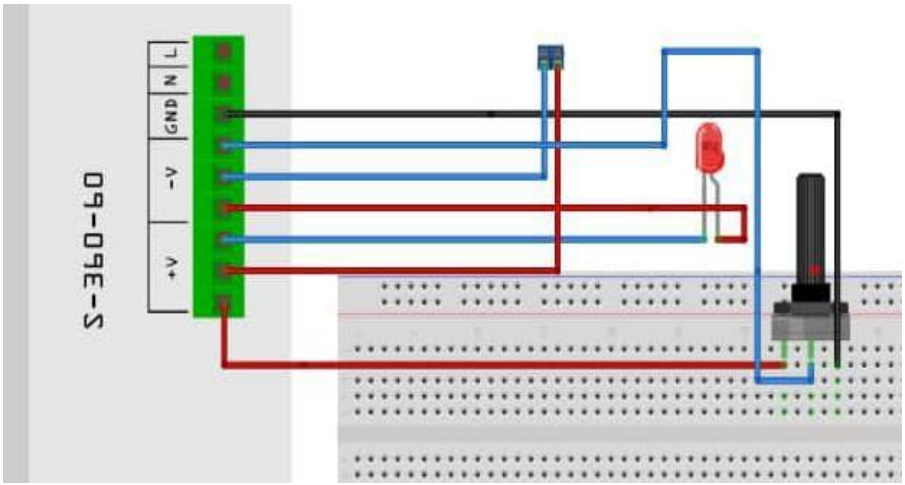
Studi literatur dilakukan untuk mengetahui konsep dasar dalam penelitian ini. Literatur yang dikaji pada penelitian ini yaitu dapat mengelompokkan tingkat kematangan pada buah nanas, teknik pengolahan citra digital, analisis ciri citra dan konsep dasar menggunakan metode CNN.

3.3.2. Perancangan Sistem Deteksi

Pada sistem deteksi buah nanas dilakukan perancangan perangkat keras *(hardware)* dan perangkat lunak *(software)* sistem identifikasi tingkat kematangan buah nanas.

3.3.2.1. Perancangan Rangkaian *LED*

Pada kotak pendeteksi kematangan buah nanas terdapat *LED* di dalamnya yang berfungsi sebagai penerangan pada kotak deteksi selama proses dilakukan, untuk rangkaiannya dapat dilihat pada **Gambar 3.2**.

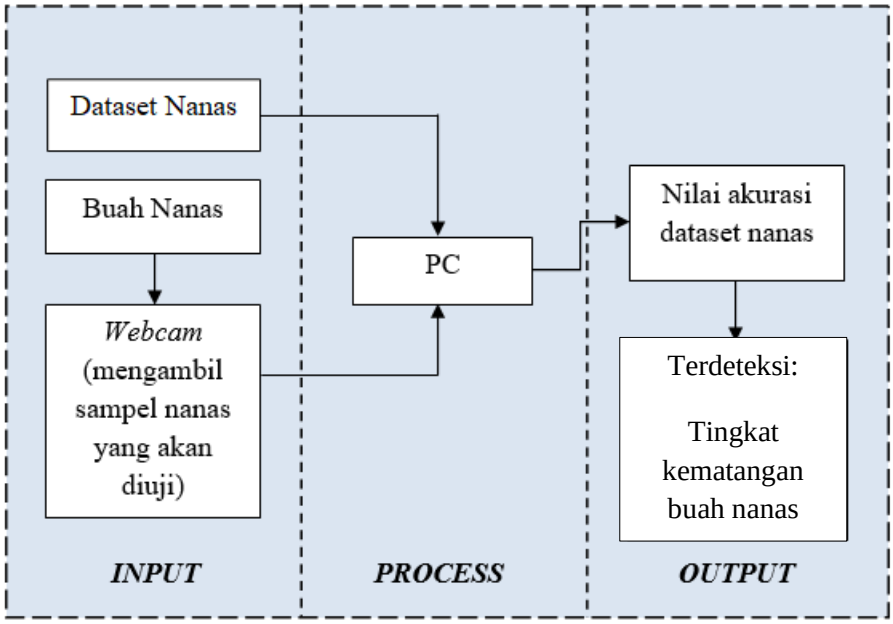


Gambar 3.2 Rangkaian *LED*

Pada rangkaian diatas terdapat sebuah potensiometer, *power supply* 20V, *LED*, *breadboard*, kabel jumper dan terminal. Pada potensiometer terdapat 3 dataset yaitu *ground*, *input* dan *output*. Pada *LED* memiliki sumber positif (+) dan negatif (-). Pada *power supply* memiliki dataset v+, v-, *ground*, N dan L. Terakhir pada terminal terdapat 2 pin. Untuk merangkai rangkaian tersebut, langkah pertama yaitu *ground* pada potensiometer disambungkan ke *ground* yang ada pada *power supply*, untuk *output* dan *input* disambungkan ke pin *power supply* v- dan v-, pada *LED* pin (+) dan pin (-) disambungkan ke v+ dan v- yang terdapat di *power supply* dan yang terakhir pada terminal terdapat 2 pin masing-masing disambungkan ke v+ dan v- pada *power supply*.

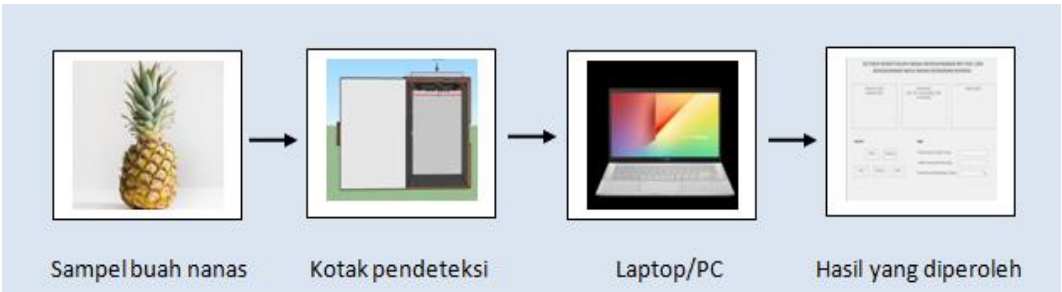
3.3.2.2. Perancangan Rangkaian Keseluruhan

Adapun perancangan perangkat lunak pada penelitian ini dapat dilihat pada **Gambar 3.3**.



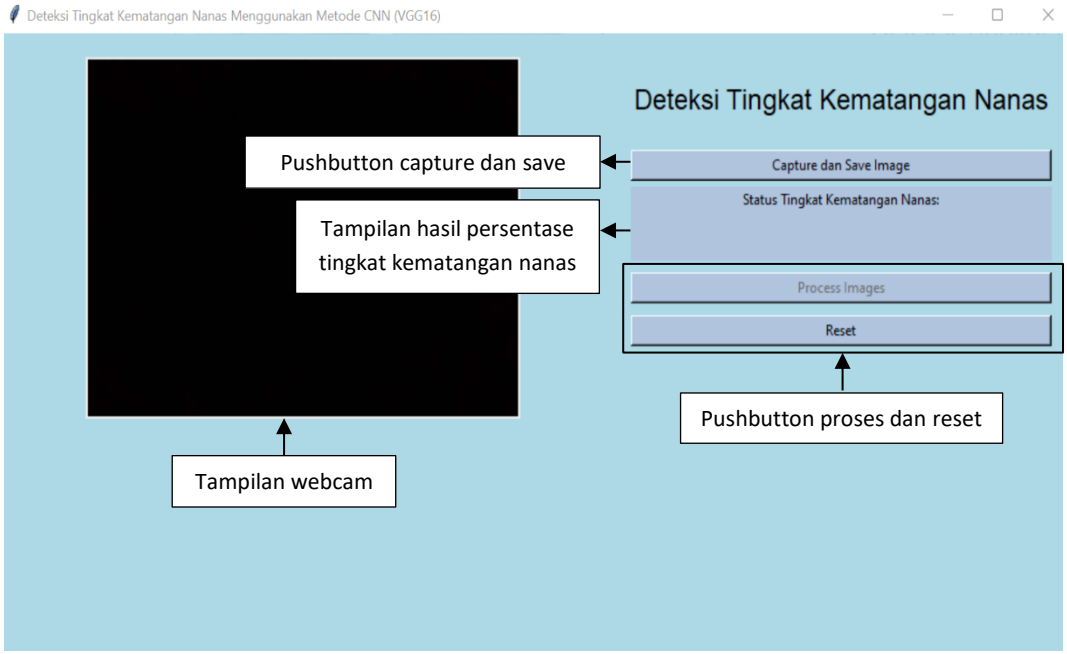
Gambar 3.3 Diagram blok sistem identifikasi kematangan nanas

Pada penelitian ini dirancang alat pendeteksi kematangan buah nanas yang terbuat dari sebuah kotak dari bahan kayu dengan ukuran 40x40x42 cm dan sebagai tempat sampel nanas diproses. Pada kotak deteksi tersebut terdapat juga sebuah lampu *LED* sebagai sumber penerangan dan *webcam* berfungsi untuk pengambilan citra sampel pada buah nanas dan PC untuk pengelola program citra yang sudah diperoleh dari *Webcam* yang akan diidentifikasikan tingkat kematangan buah nanas tersebut. Adapun diagram blok pada perancangan perangkat keras pada penelitian ini dapat dilihat pada **Gambar 3.4**.



Gambar 3.4 Diagram Blok deteksi kematangan buah nanas

Pada **Gambar 3.4** terdapat sampel buah nanas sebagai sampel untuk pengujian yang dilakukan pada penelitian ini. Pengambilan sampel dilakukan pada kotak pendeteksi yang di dalamnya terdapat satu buah nanas dan lampu LED sebagai sumber cahaya pada kotak. *Webcam* yang terletak tepat diatas kotak pendeteksi akan menangkap atau men *capture* buah nanas yang berada di dalam kotak. Setelah pengambilan sampel selesai, dilanjutkan dengan proses deteksi tingkat kematangan nanas yang dilakukan laptop atau PC dengan menggunakan program *python.* Output dari proses yang dilakukan berupa *interface* GUI yang telah di *design* menggunakan tkinter. Pada **Gambar 3.5** menunjukkan *design interface* deteksi tingkat kematangan nanas.



Gambar 3.5 *Design interface* deteksi kematangan nanas

Design interface dirancang pada program python menggunakan Tkinter. Tkinter merupakan salah satu fungsi pada python yang mampu membuat tampilan GUI dengan dimasukkannya program secara *default* ke dalam instalasi python. Pada tampilannya terdapat 1 kolom utama yang menampilkan buah nanas dari *webcam* (aktif) dan akan siap untuk di *capture.* Adapun kontrol panel pada GUI diantaranya button untuk meng *capture* dan *save*, jika nanas sudah diambil gambarnya sebanyak 5 *layer* atau penampakan keseluruhan dari gambar buah maka akan secara otomatis tersimpan pada dokumen yang bernama "dataset_gambar" yang berformat "model.h5". Setelah melakukan *capture* dan *save* maka sampel yang sudah diambil

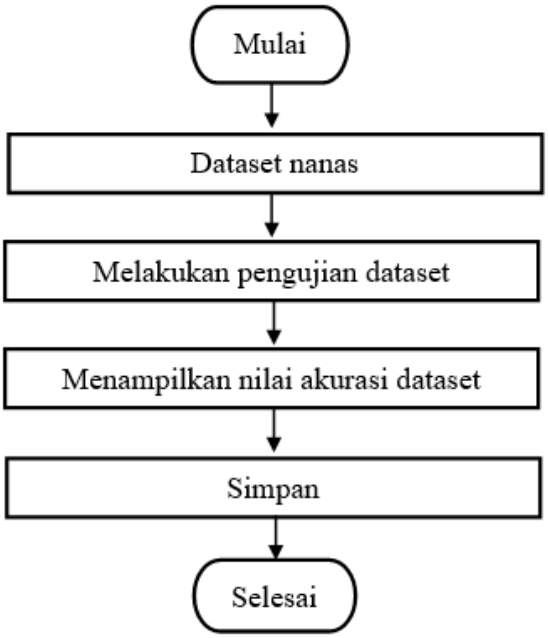
akan diproses setelah mengklik button proses. Pada proses ini menggunakan algoritma VGG16 yaitu, salah satu algoritma yang terdapat pada CNN yang akan menghasilkan nilai kuantifikasi atau persentase kematangan pada buah nanas. Adapun button reset yang akan mereset sampel gambar nanas yang telah diuji sebelumnya dan akan memulai menguji sampel nanas berikutnya.

3.3.3. Convolutional Neural Network (CNN)

Tingkat kematangan buah nanas dapat dilakukan salah satunya dengan menggunakan algoritma CNN model arsitektur VGG16. Sebelum melakukan klasifikasi kematangan pada buah nanas untuk mendapatkan nilai persentase kematangan pada buah nanas, terlebih dahulu dapat dilakukan pelatihan dataset nanas untuk mengetahui keakuratan dataset yang dimiliki.

3.3.3.1. Pelatihan CNN

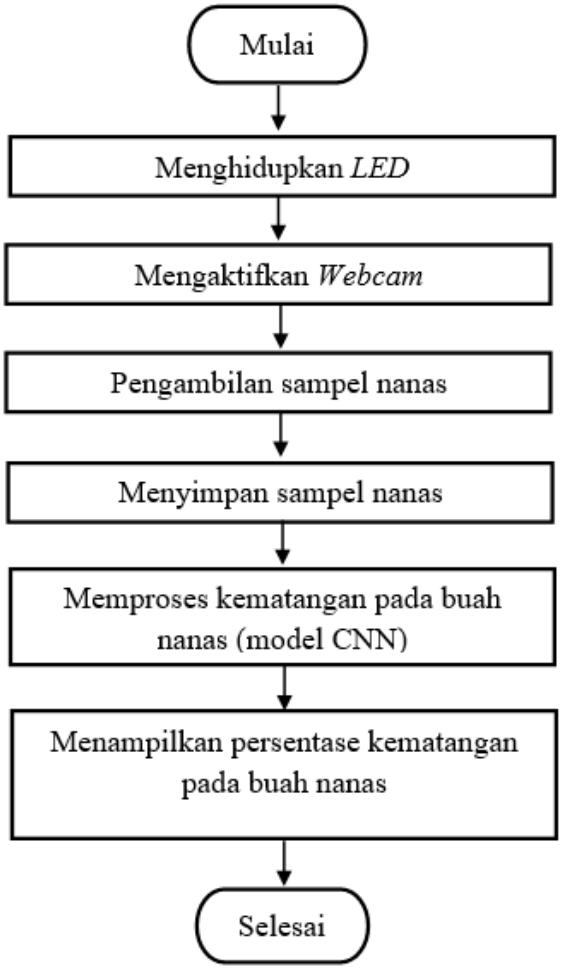
Pada pelatihan ini digunakan dataset nanas untuk mengetahui keakuratan dataset dengan sampel nanas yang akan diuji selanjutnya. Dataset nanas dibagi menjadi tiga bagian diantaranya yaitu data *train*, validasi dan *test*, masing-masing data terdapat dua macam buah nanas yaitu, nanas matang dan mentah. Pada hasil pelatihan ini akan diperlihatkan di terminal pada python yang akan menghasilkan nilai akurasi dataset. Pada metode CNN melakukan pelatihan dengan membagi beberapa data yaitu *train*, validasi dan *test* dengan epoch 10 kali. Pada **Gambar 3.6** menunjukkan diagram alir menampilkan nilai akurasi pelatihan.



Gambar 3.6 Diagram alir menampilkan nilai akurasi pelatihan CNN

3.3.3.2. Kuantifikasi Pemodelan CNN

Secara garis besar arsitektur model CNN terdiri dari beberapa layer seperti *convolutional layer, pooling layer, dropout layer, flatten layer* dan *fully connected layer.* Setelah melakukan pelatihan sebelumnya akan dilanjutkan dengan pemodelan menggunakan metode CNN, pada dataset nanas yang telah dilatih sebelumnya dijadikan bahan keakuratan data yang akan digunakan pada pelatihan selanjutnya yaitu mendapatkan nilai kuantifikasi tingkat kematangan nanas yang akan diuji menggunakan sampel nanas sesungguhnya. Pada **Gambar 3.7** menunjukkan diagram alir deteksi tingkat kematangan nanas.



Gambar 3.7 Diagram alir deteksi tingkat kematangan nanas

Pada metode *Convolutional Neural Network* (CNN) menggunakan algoritma arsitektur model *Visual Geometri Group* (VGG16) yang telah dilatih sebelumnya melalui *transfer learning* akan digunakan untuk dataset nanas *input channel* sebanyak tiga (*red, green* dan *blue*). *Output* pada konvolusi pertama akan dijadikan *input* untuk tahapan konvolusi kedua dan seterusnya. Adapun tahapan-tahapan pada metode CNN sebagai berikut.

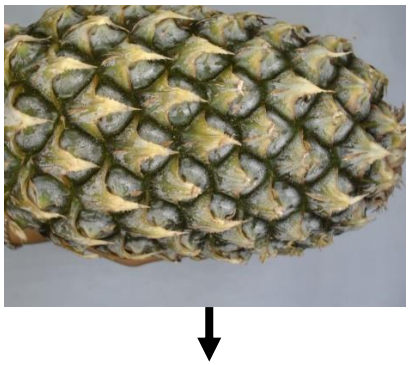
3.3.3.3. *Input layer*

Citra input berdimensi 64x64x3. Data citra yang telah di *capture* dimasukkan kedalam *input layer*. Ukuran *input* dan *output* dari layer adalah 64x64 dengan ukuran matriks 3x3 dan *channel* RGB.

3.3.3.4. *Convolutional layer*

Setelah menjalani *input* maka masuk ke tahap selanjutnya yaitu konvolusi. Tahapan ini akan memanfaatkan inputan yang didapatkan sebelumnya. Konvolusi pada tahapan ini menggunakan kernel 3x3, *padding same* (0), fungsi aktivasi ReLU dan *strides* 1.

Berikut adalah proses konvolusi dengan memberikan nilai filter pada matrix. Pixel citra dengan tiga *channel* warna yaitu *red*, *green* dan *blue* diambil data pixelnya masing-masing dan ditambahkan padding (0) disetiap pixelnya sehingga didapatkan pixel seperti **Gambar 3.8**.



Red

0	0	0	0	0
0	138	142	150	0
0	120	125	128	0
0	128	132	141	0
0	0	0	0	0

(a)

Green

0	0	0	0	0
0	136	141	153	0
0	112	118	122	0
0	122	126	141	0
0	0	0	0	0

(b)

Blue

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(c)

Gambar 3.8 (a) pixel red; (b) pixel green; (c) pixel blue

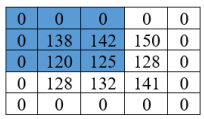
Misalkan pada kasus ini digunakan kernel 3x3 dengan nilai seperti pada **Gambar 3.9**.

0	1	2
-1	0	1
-2	-1	0

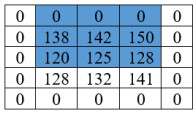
Gambar 3.9 kernel 3x3 (Astrid, 2019)

Langkah selanjutnya dilakukan proses perhitungan konvolusi di setiap *channel* dengan dikalikan dengan ukuran 3x3 pada **Gambar 3.9**. tahapan ini dilakukan berulang kali dengan pergeseran kernel sebanyak 1 *stides* di setiap *channel* nya sehingga didapatkan perhitungan konvolusi di setiap *channel* dengan nilai sebagai berikut.

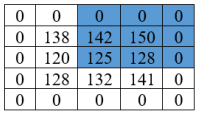
Proses konvolusi pada *channel red*



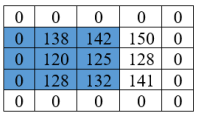
(0*0) + (1*0) + (2*0) + ((-1)*0) + (0*138) + (1*142) + ((-2)*0) + ((-1)*120) + (0*125) = 22



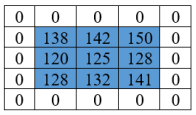
(0*0) + (1*0) + (2*0) + ((-1)*138) + (0*142) + (1*150) + ((-2)*120) + ((-1)*125) + (0*128) = -353



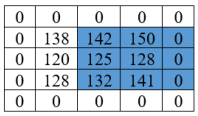
(0*0) + (1*0) + (2*0) + ((-1)*142) + (0*150) + (1*0) + ((-2)*0) + ((-1)*125) + (0*0) = -520



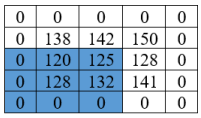
(0*0) + (1*138) + (2*142) + ((-1)*0) + (0*120) + (1*125) + ((-2)*0) + ((-1)*128) + (0*132) = 419



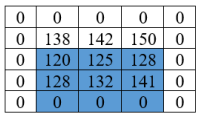
(0*138) + (1*0142) + (2*150) + ((-1)*120) + (0*125) + (1*128) + ((-2)*128) + ((-1)*132) + (0*141) = 62



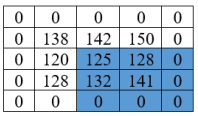
(0*142) + (1*150) + (2*0) + ((-1)*125) + (0*128) + (1*0) + ((-2)*132) + ((-1)*141) + (0*0) = -380



(0*0) + (1*120) + (2*125) + ((-1)*0) + (0*128) + (1*132) + ((-2)*0) + ((-1)*0) + (0*0) = 502



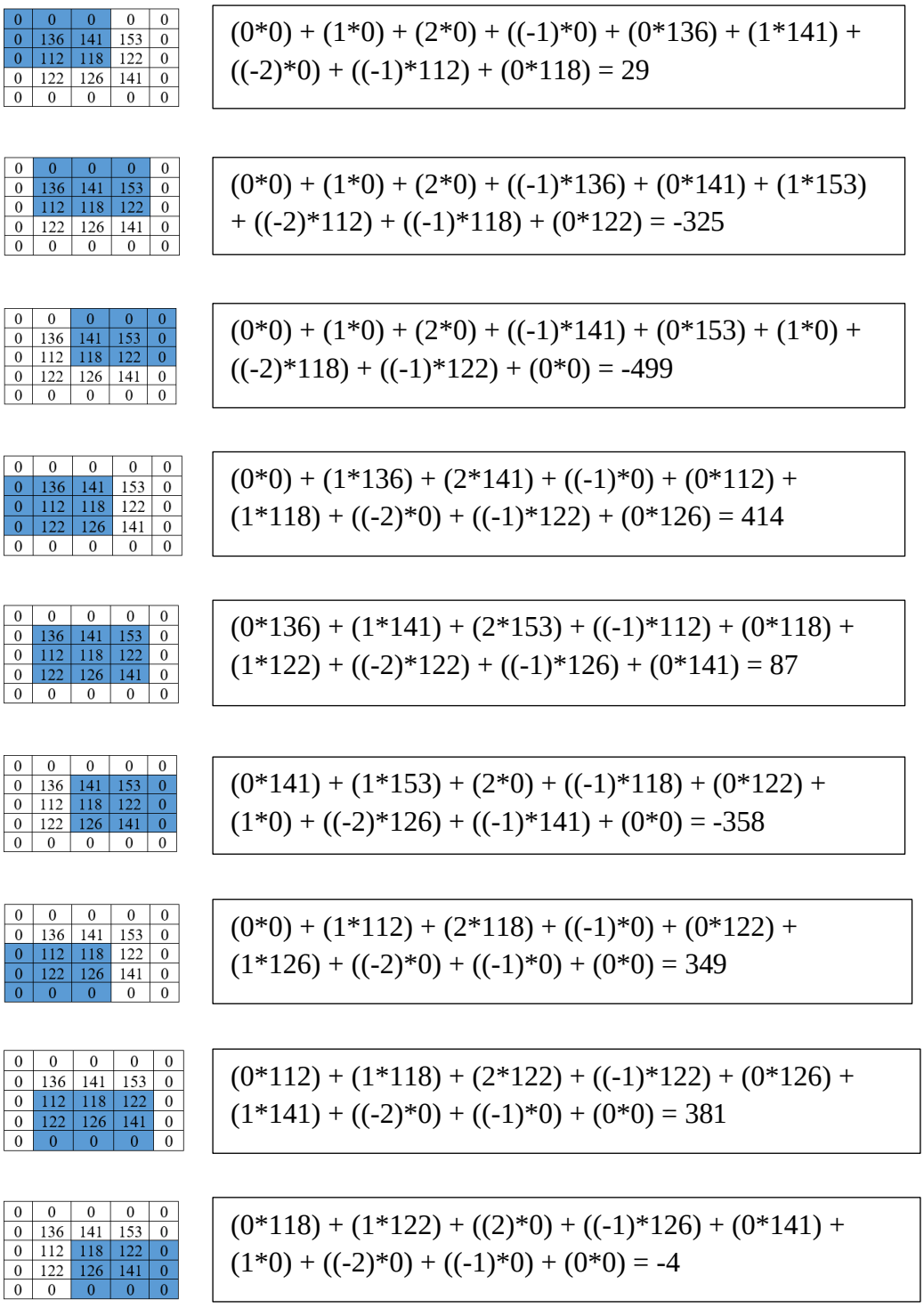
(0*120) + (1*125) +((2*128) + ((-1)*0) + (0*132) + (1*141) + ((-2)*0) + ((-1)*0) + (0*) = 394



(0*125) + (1*128) + (2*0) + ((-1)*132) + (0*141) + (1*0) + ((-2)*0) + ((-1)*0) + (0*0) = -4

Gambar 3. 10 Ilustrasi konvolusi *channel red*

Proses konvolusi pada *channel green*



Gambar 3. 11 Ilustrasi konvolusi *channel green*

Proses konvolusi pada *channel blue*

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*0) + (1*0) + (2*0) + ((-1)*0) + (0*128) + (1*133) + ((-2)*0) + ((-1)*79) + (0*88) = 54

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*0) + (1*0) + (2*0) + ((-1)*128) + (0*133) + (1*157) + ((-2)*79) + ((-1)*88) + (0*99) = -217

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*0) + (1*0) + (2*0) + ((-1)*0133) + (0*157) + (1*0) + ((-2)*88) + ((-1)*99) + (0*0) = -408

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*0) + (1*128) + (2*133) + ((-1)*0) + (0*79) + (1*88) + ((-2)*0) + ((-1)*90) + (0*94) = 392

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*128) + (1*133) + (2*157) + ((-1)*79) + (0*88) + (1*99) + ((-2)*90) + ((-1)*94) + (0*130) = 193

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*33) + (1*157) + (2*0) + ((-1)*88) + (0*99) + (1*0) + ((-2)*94) + ((-1)*130) + (0*0) = -294

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*0) + (1*79) + (2*88) + ((-1)*0) + (0*90) + (1*94) + ((-2)*0) + ((-1)*0) + (0*0) = 474

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*79) + (1*88) + (2*99) + ((-1)*90) + (0*94) + (1*142) + ((-2)*0) + ((-1)*0) + (0*0) = 326

0	0	0	0	0
0	128	133	157	0
0	79	88	99	0
0	90	94	130	0
0	0	0	0	0

(0*88) + (1*99) + (2*0) + ((-1)*94) + (0*130) + (1*0) + ((-2)*0) + ((-1)*0) + (0*0) = -5

Gambar 3. 12 Ilustrasi konvolusi *channel blue*

Hasil konvolusi pada ketiga *channel red, green* dan *blue* dihasilkan dalam bentuk matrix seperti pada **Gambar 3.13**.

Red

22	-353	-520
419	62	-380
502	394	-4

Green

29	-325	-499
414	87	-358
349	381	-4

Blue

54	-217	-408
392	193	-294
474	326	-5

Gambar 3.13 Hasil pada *channel red, green dan blue* perhitungan konvolusi

Setiap nilai dari *channel layer* konvolusi yang sudah melalui tahapan perhitungan pada **Gambar 3.10, Gambar 3.11** dan **Gambar 3.12**, selanjutnya dilakukan penjumlahan pada tiap *channel* untuk mengetahui total dari perhitungan RGB pada proses konvolusi.

Red

22	-353	-520
419	62	-380
502	398	-4

Green

29	-325	-499
414	87	-358
349	381	-4

Blue

54	-217	-408
392	193	-294
474	326	-5

Gambar 3.14 Ilustrasi penjumlahan total nilai RGB

Untuk mengetahui nilai total RGB pada perhitungan konvolusi dapat menggunakan persamaan 3.1.

$$\text{Total} = \text{red}+\text{green}+\text{blue} \quad (3.1)$$

Misal, penjumlahan pada kolom pertama baris pertama dan kolom terakhir dan baris terakhir.

Total = 22+29+54 = 105

Total = (-4)+(-4)+(-5) = -13

sehingga menghasilkan total penjumlahan dari ketiga *channel red, green* dan *blue* dapat dilihat seperti **gambar 3.15**.

Total = Red + Green + Blue

105	-895	-1.427
952	342	-1.032
1.370	1.101	-13

Gam bar 3.15 Total hasil perhitungan konvolusi

Dengan aktivasi ReLU maka setiap nilai yang negatif akan diubah menjadi angka nol (0) sehingga didapatkan hasil konvolusi seperti pada **Gambar 3.16**.

Total = Red + Green + Blue

105	-895	-1.427
952	342	-1.032
1.370	1.101	-13

ReLu

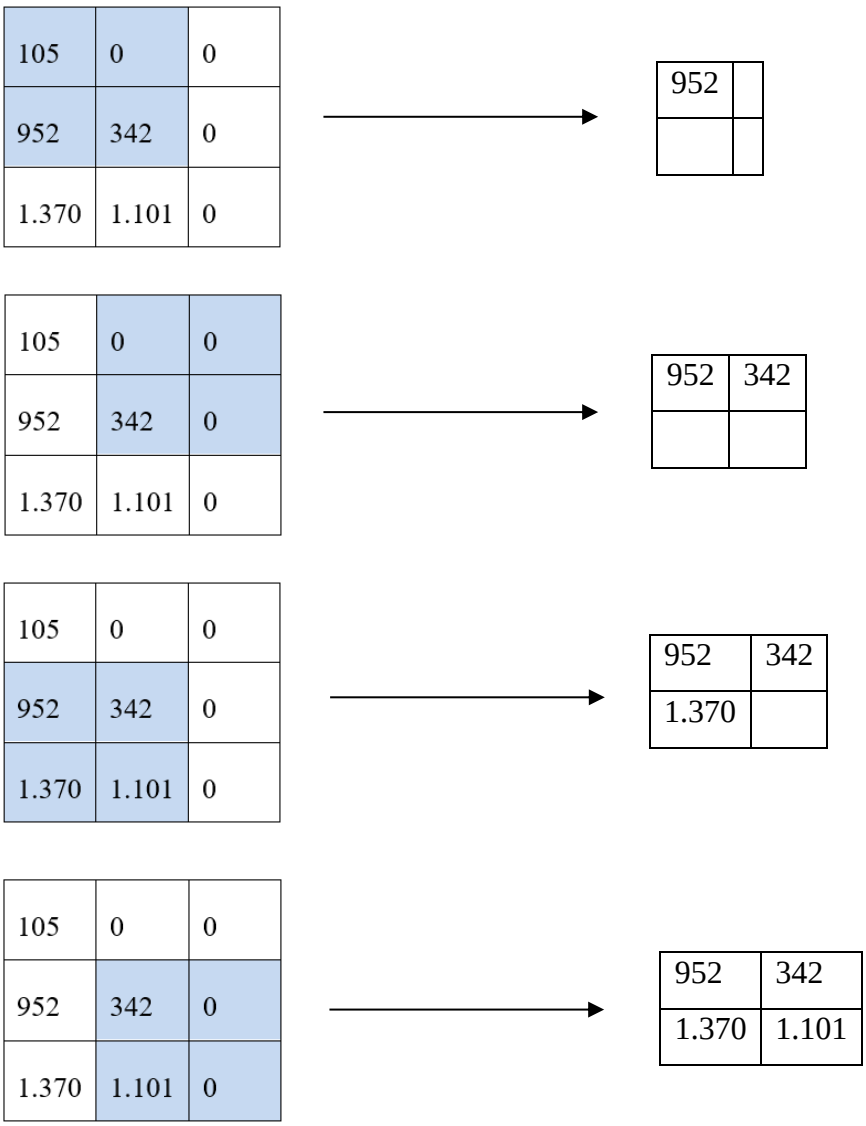
105	0	0
952	342	0
1.370	1.101	0

Gambar 3.16 Hasil konvolusi

Proses ini dilakukan secara berulang sampai semua citra dengan jenis filter yang berbeda sehingga *output* dari tahapan konvolusi ini akan menghasilkan banyak *feature map*, **Gambar 3.16** ini merupakan contoh *output* dari salah satu *feature map*.

3.3.3.5. *Pooling layer*

Pada hasil perhitungan konvolusi pada **Gambar 3.16** akan dijadikan sebagai *input* pada *pooling layer*. Pada tahapan ini menggunakan *max pooling*, kernel *size* 2x2 dan *strides* 1. Ilustrasi proses *max pooling* ditunjukkan seperti pada **Gambar 3.17**.



Gambar 3.17 Ilustrasi perhitungan layer (*max pooling*)

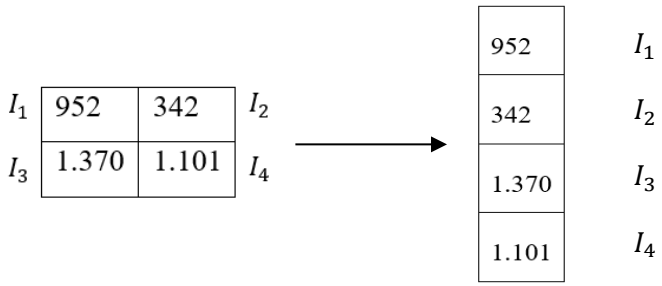
Pada **Gambar 3.17** merupakan tahapan *max pooling* dimana tahapan ini dilakukan dengan kernel size 2x2, dari kernel ini akan diambil nilai tertinggi sehingga didapatkan nilai pertama yaitu 952, selanjutnya bergeser 1 strides sehingga didapatkan nilai 342, dan begitu seterusnya untuk proses *max pooling.* Sehingga didapatkan nilai *output* pada *max pooling* dalam bentuk matrix 2x2 sebesar 952, 342, 1.370 dan 1.011. Matrix ini akan dijadikan *input*an pada *dropout layer.*

3.3.3.6. *Dropout layer*

Dropout berfungsi untuk mengurangi kompleksitas model yang telah dibangun dengan membuang *neural network* yang tidak terpakai.

3.3.3.7. *Flatten layer*

Flatten merupakan tahapan mengubah matriks ke bentuk vektor 1 dimensi. Dari *output dropout layer* akan didapatkan bentuk vektor baru, gambar ilustrasi dapat dilihat pada **Gambar 3.18**.

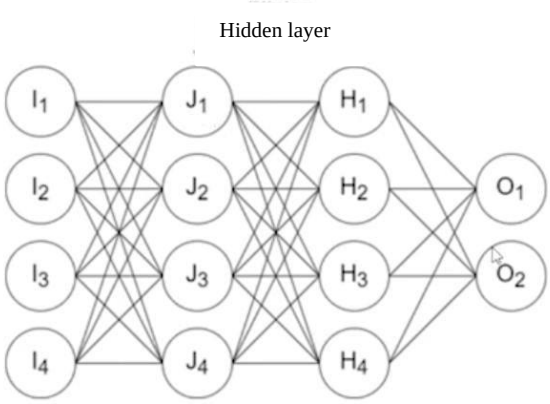


Gambar 3.18 Ilustrasi proses *flatten layer*

Proses *flatten* dimana matriks 2x2 diubah menjadi 4x1 atau vektor. *Output* pada tahapan ini adalah vektor 1 dimensi yang akan digunakan sebagai input di *fully connected layer* (*dense*).

3.3.3.8. *Fully connected layer*

Output dari tahapan *flatten* yaitu 1 vektor 1 dimensi, yang akan menjadi *input*an pada tahapan ini, berikut ilustrasi *fully connected layer* seperti pada **Gambar 3.19**.



Gambar 3.19 Ilustrasi proses *dense* + *softmax* (Luthfieta, 2023)

Tahapan akhir dari model CNN adalah *dense* dengan fungsi aktivasi *softmax*. Berikut perhitungan dari setiap *hidden layer*.

$$\sum_{i=1}^{N} l_i \times v_{ij} = J_i \quad (3.2)$$

$$\sum_{i=1}^{N} J_i \times w_{ij} = H_i \quad (3.3)$$

$$\sum_{i=1}^{N} H_i \times x_{ij} = O_i \quad (3.4)$$

Dengan diketahui l_i adalah nilai inputan pertama yang dihasilkan dari proses *flatten layer*, J_i adalah nilai output dari *hidden layer yang pertama*, H_i adalah nilai output dari proses *hidden layer* yang kedua dan v_{ij}, w_{ij} dan x_{ij} adalah nilai bobot yang nilainya harus berbeda-beda pada proses *hidden layer* dan O_i adalah output terakhir pada proses *hidden layer*.

Dari nilai output *flatten layer* pada **Gambar 3.18** yang kemudian dilakukan *fully connected layer* dengan perhitungan pada setiap hidden layer sebagai berikut.

$J_1 = (952x0{,}1) + (342x0{,}1) + (1.370x01), +(1.032x0{,}1) = 129{,}6$

$J_2 = (952x0{,}2) + (342x0{,}2) + (1.370x0{,}2) + (1.032x0{,}2) = 259{,}2$

$J_3 = (952x0{,}3) + (342x0{,}3) + (1.370x0{,}3) + (1.032x0{,}3) = 388{,}9$

$J_4 = (952x0{,}4) + (342x0{,}4) + (1.370x0{,}4) + (1.032x0{,}4) = 518{,}5$

Hidden layer pertama di ilustrasikan dengan j1, j2, j3 dan j4 dimana setiap *neuron* dari 387, 458, 387 dan 458 dikalikan dengan nilai *weight* yang berbeda-beda sehingga menghasilkan nilai J_1=169, J_2=507, J_3=676 dan J_4=338. setiap *neuron* dari J_1, J_2, J_3 dan J_4 ini akan dikalikan Kembali dengan *weight* yang berbeda-beda untuk menghasilkan nilaiH_1, H_2, H_3 dan H_4 sebagai *hidden layer* selanjutnya.

$H_1 = (129{,}2x0{,}2) + (388{,}9x0{,}2) + (\ 518{,}50{,}2) + (129{,}6x0{,}2) = 261{,}4$

$H_2 = (129{,}2x0{,}3) + (388{,}9x0{,}3) + (\ 518{,}5x0{,}3) + (129{,}6x0{,}3) = 392{,}1$

$H_3 = (129{,}2x0{,}4) + (\ 388{,}9x0{,}4) + (\ 518{,}5x0{,}4) + (129{,}6x0{,}4) = 522{,}8$

$H_4 = (129{,}2x0{,}5) + (388{,}9x0{,}5) + (\ 518{,}5x0{,}5) + (129{,}6x0{,}5) = 653{,}6$

Setiap *neuron* dari H_1, H_2, H_3 dan H_4 ini akan dikalikan kembali dengan *weight* yang berbeda-beda untuk menghasilkan nilai O_1 dan O_2 sebagai *hidden layer* selanjutnya.

$O_1 = (261{,}4x0{,}1) + (392{,}1x0{,}1) + (522{,}8x0{,}1) + (653{,}6x0{,}1) = 182{,}9$

$O_2 = (261{,}4x0{,}2) + (392{,}1x0{,}2) + (522{,}8x0{,}2) + (653{,}6x0{,}2) = 365{,}9$

Langkah selanjutnya adalah perhitungan *softmax* dengan persamaan eksponensial O_1 dibagi dengan total eksponensial O_1 dan O_2. Tahapan ini dilakukan dua langkah baik pada O_1 dan O_2 dengan persamaan (3.4).

$$s(o_i) = \frac{e^{o_1}}{e^{o_1} + e^{o_2}} \tag{3.5}$$

(Luthfieta, 2023).

Dengan diketahui e^{o_1} adalah nilai eksponensial dari output pertama dan e^{o_2} adalah nilai eksponensial dari output kedua, maka dari perhitungan menggunakan persamaan (3.5) didapatkan hasil:

$s(o_1) = \frac{e^{182{,}9}}{e^{182{,}9} + e^{365{,}9}}$ $\quad s(o_1) = \frac{2{,}70}{10{,}79}$ $\quad s(o_1) = 0.2$

$ss(o_2) = \frac{e^{365{,}9}}{e^{182{,}9} + e^{365{,}9}}$ $\quad s(o_2) = \frac{8{,}09}{10{,}79}$ $\quad s(o_2) = 0.7$

Sehingga didapatkan bobot nilai probabilitas yang lebih rendah pada O_1 yaitu 0.2 yang berarti input citra yang dimasukkan diprediksi adalah mentah, dengan nilai keakuratan gambar dengan nilai persentasenya dengan menggunakan persamaan 3.5, karena dapat dilihat gambar yang digunakan adalah buah nanas yang masih mentah atau belum matang.

Status persentase = nilai probabilitas terendah atau tertinggi x 100% (3.6)

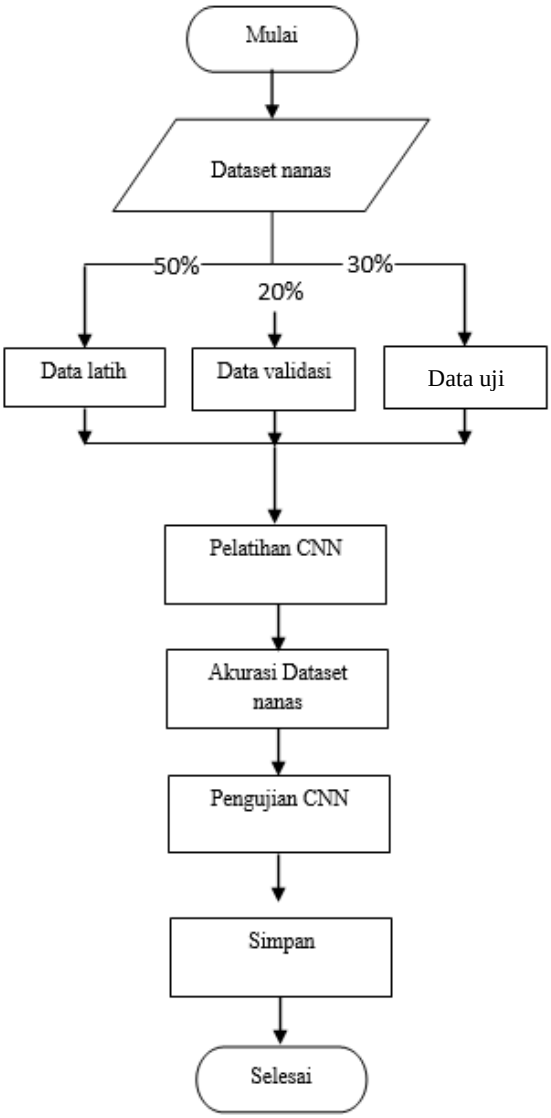
Tingkat kematangan buah nanas dapat dihitung dengan menggunakan persamaan 3.1 sampai dengan persamaan 3.6. setelah didapatkan hasil dari persentase

kematangan buah nanas maka akan digunakan logika if else untuk mengklasifikasi kematangan nanas. Untuk membandingkan antara perhitungan manual dengan perhitungan yang diperoleh oleh sistem yang telah dirancang yaitu menggunakan metode *Convolutional Neural Network* (CNN), dapat dilihat pada **Tabel 3.2**.

Tabel 3.2 Rancang tabel data nilai kuantifikasi deteksi kematangan nanas.

Gambar	Sistem		Manual		Persentase Akurasi Kematangan Nanas
	Persentase Kematangan Nanas	Tingkat Kematangan Nanas	Persentase Kematangan Nanas	Tingkat Kematangan Nanas	
1					
2					
3					
…					
Rata-rata					

Kematang buah nanas diklasifikasikan berdasarkan persentase tingkat kematangan buah nanas tersebut. Pada penelitian ini untuk mengetahui tingkat kematangan buah nanas dengan menggunakan metode CNN. Mengingat bahwa metode CNN juga terdapat arsitektur jaringan per *layer*, dimana untuk memudahkan *preprocessing* dilakukan pada proses penelitian ini. Berikut diagram alir proses menggunakan metode CNN, dapat dilihat pada **Gambar 3.20**.



Gambar 3.20 Diagram alir sistem

Pengambilan data hasil pada penelitian ini dilakukan dengan menggunakan metode *convolutional neural network* (CNN) dan pada tampilan hasilnya akan menggunakan panel GUI dengan tkinter sekaligus sebagai pengontrol pada PC yang akan menampilkan hasil. Untuk tahap proses deteksi kematangan buah nanas dapat dilihat, sebagai berikut.

a. Menyiapkan bahan utama yang akan diuji yaitu buah nanas dengan tingkat kematangan yang berbeda.

b. Menyiapkan kotak deteksi kematangan buah nanas, sekaligus mengaktifkan atau menghidupkan *LED* serta menghubungkan *Webcam* ke PC yang akan menampilkan panel GUI.
c. Setelah pada bagian (b) sudah siap, maka akan dilakukan *processing* pada buah nanas, dengan memasukkan buah nanas ke dalam kotak deteksi.
d. Setelah nanas sudah siap diletakkan ke dalam kotak deteksi, maka selanjutnya mengklik *button capture* dan *save* pada tampilan panel GUI. Lalu akan muncul gambar nanas pada layar dan otomatis tersimpan pada dokumen yang sudah disediakan.
e. Selanjutnya gambar nanas telah diambil sebagai sampel yang akan diuji maka akan dilakukan proses dengan mengklik *button* proses.
f. Setelah itu maka hasil akan terlihat pada tampilan panel GUI nilai kuantifikasi atau persentase tingkat kematangan buah nanas.

Setelah mengetahui nilai status persentase pada kematangan nanas, selanjutnya yaitu menentukan persentase keakuratan dan kesalahan dari gambar yang terdeteksi dengan perhitungan pixel gambar yang sebenarnya.

$$\% \text{ Kesalahan} = \frac{|nilai\ sesungguhnya - nilai\ percobaan|}{|nilai\ sesungguhnya|} \text{ x } 100\% \qquad (3.7)$$

$$\text{Akurasi} = 100\% - \% \text{ Kesalahan} \qquad (3.8)$$

(Sulandari *et al*., 2014).

3.4. Analisis Data Peningkatan Hasil Deteksi Kematangan Nanas

Pada tahap ini dilakukan pengujian dan analisis sistem deteksi tingkat kematangan buah nanas, yang telah dirancang dengan metode CNN. Secara umum, sistem yang dirancang terdiri dari tiga bagian utama yaitu, *input*, proses dan *output*. Maksudnya adalah pada citra nanas prosesnya melibatkan klasifikasi oleh metode CNN dan keluarannya adalah tingkat kematangan pada buah nanas. Sebelum melakukan proses validasi pada sampel nanas yang akan diuji maka terlebih dahulu akan dilakukan pelatihan pada dataset nanas yang terdiri dari beberapa data yaitu *train*, validasi dan *test*. Dataset yang telah dilatih untuk mengetahui seberapa akurat dataset yang akan digunakan untuk pengujian sampel nanas selanjutnya. Untuk

melakukan pengujian sampel nanas disiapkan tiga buah nanas dengan tingkat kematangan yang berbeda (mentah, setengah matang dan matang), untuk pengambilan data sampel dilakukan selama delapan hari berturut-turut supaya mendapatkan hasil yang lebih fleksibel. Sampel kemudian digunakan untuk validasi kinerja dari metode CNN yang menggunakan algoritma model arsitektur VGG16. Sepanjang penyelesaian penelitian, python digunakan untuk bahasa pemrograman sebagai perpustakaan pembelajaran yang mendalam. Kematangan buah nanas dapat dibedakan berdasarkan tampak luar warna kulit buah nanas yang berwarna kuning. Berikut tabel rancang nilai tingkat kematangan nanas dapat dilihat pada **Tabel 3.3**.

Tabel 3.3 Data tingkat persentase hasil sistem deteksi

Gambar	Pengambilan ke	Persentase Tingkat Kematangan Nanas (%)
1		
2		
3		

IV. HASIL DAN PEMBAHASAN

4.1. Implementasi Perangkat Keras Alat Deteksi Kematangan Buah Nanas

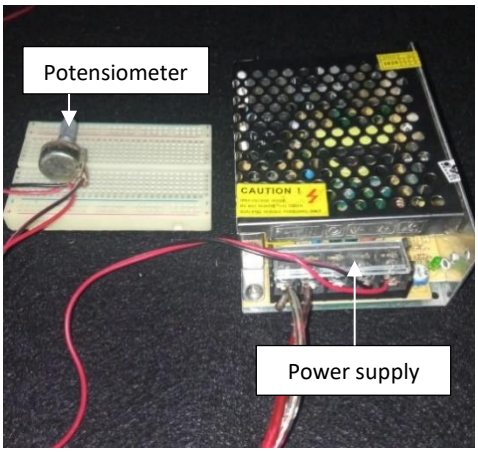
Perancangan perangkat keras alat pendeteksi kematangan buah nanas telah direalisasikan dengan hasil yang ditunjukkan pada **Gambar 4.1**. Sistem pendeteksi kematangan buah nanas terdiri dari *webcam*, kotak deteksi dan *personal computer* (PC). Kotak deteksi dibuat dari bahan multiplek dengan dimensi 40x40x40 cm. Bagian dalam dinding kotak dilapisi dengan kain panel berwarna hitam dan pada bagian atas kotak diberi lampu *led* sebagai sumber penerangan.

(a) Tampak luar (b) Tampak dalam

Gambar 4.1 Tempat atau kotak deteksi nanas

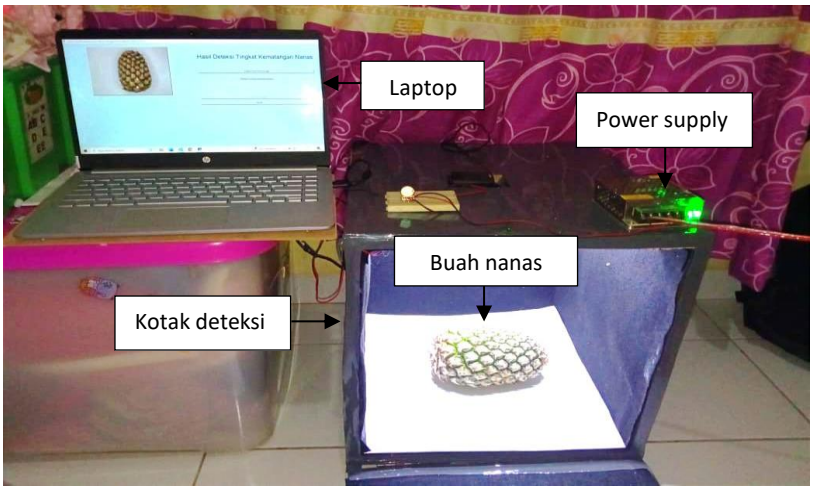
Sistem penerangan dalam kotak deteksi terdapat 8 buah lampu *led* (paralel) strip masing-masing diberi daya 7,5 watt yang diberi tegangan sebesar 12 volt dan arus 5 ampere. Adapun potensiometer yaitu untuk mengatur tingkat penerangan pada kotak deteksi dan juga menggunakan *power supply* 12V 5A sebagai sumber daya pada *led* stripnya. Intensitas cahaya pada kotak deteksi nanas yaitu sebesar 477 lux (Yanti *et al.,* 2022).

Pada **Gambar 4.2**. menunjukkan komponen potensiometer dan p*ower supply* 12 volt dan 5 ampere.



Gambar 4.2 Potensiometer dan *power supply* 12V 5A

Alat untuk mendeteksi kematangan buah nanas dengan metode *convolutional neural network* (CNN) dengan menggunakan algoritma model VGG16, yang terdiri dari kotak deteksi dan laptop atau PC sebagai perangkat keras untuk pengolahan citra pada program *python*. Alat pendeteksi ini berfungsi untuk mengakuisisi gambar nanas dengan *webcam*, dari hasil yang diperoleh dari *webcam* selanjutnya diproses dengan program yang telah dibuat pada laptop menggunakan bahasa pemrograman *python*. *Output* dari alat ini yaitu, dapat menampilkan gambar dan tingkat kematangan buah nanas pada GUI yang telah didesain langsung dari *python*. Pada **Gambar 4.3** menunjukkan gambar alat deteksi tingkat kematangan nanas.



Gambar 4.3 Alat deteksi tingkat kematangan nanas dengan metode CNN

4.2. Implementasi Perancangan *(Software)* Deteksi Kematangan Nanas

Sistem *software* yang digunakan untuk mendeteksi jumlah mata nanas yaitu menggunakan *Python* dan openCV. Sistem dari *software* ini menggunakan *Graphical User Interface* (GUI) sebagai program deteksi kematangan nanas.

4.2.1. Hasil Perancangan *Graphical User Interface* (GUI)

Perancangan *Graphical User Interface* (GUI) digunakan untuk memudahkan dalam mengontrol atau mengendalikan sistem deskripsi kematangan nanas. Pada **Gambar 4.4** menunjukkan tampilan GUI yang telah di desain dengan tkinter.

Gambar 4.4 Tampilan GUI

Tampilan GUI terdiri dari *display camera*, *button capture* dan *save*, *button process images*, *button reset*, blok hasil persentase kematangan nanas, untuk tabel *display* dan *button* dapat dilihat pada **Tabel 4.1**.

Tabel 4.1 *Display* dan *button* pada GUI

No.	***Display*/*button***	**Fungsi**
1.	*Camera*	Berfungsi untuk menampilkan video hasil tangkapan *webcam*.
2.	*Capture* dan *save image*	Berfungsi untuk menampilkan gambar yang telah ditangkap oleh sistem deteksi sekaligus menyimpan otomatis pada dokumen yang telah disediakan**.**
3.	*Process images*	Berfungsi untuk memproses image yang telah di capture menggunakan metode CNN.
4.	*Reset*	Berfungsi untuk me*reset* ulang.
5.	Blok hasil kematangan buah nanas	Berfungsi untuk menampilkan nilai kuantifikasi tingkat kematangan buah nanas.

4.2.2. Kontrol Graphical User Interface (GUI)

GUI yang telah dibuat mempunyai program kontrol untuk mengakuisisi citra nanas seperti pengambilan citra, menyimpan citra dan memproses citra. Beberapa tahapan akuisisi pada GUI dibuat dengan fungsi sebagai berikut:

4.2.2.1. Fungsi Program *Import Library* pada Python

Memproses citra pada program python juga membutuhkan beberapa *library* yang di*import* diantaranya dapat dilihat pada program *library* dibawah ini:

```
import cv2
import os
import tensorflow as tf
from keras.applications import VGG16
from keras.models import Model
from keras.layers import Dense, GlobalAveragePooling2D
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
import numpy as np
import tkinter as tk
from tkinter import Label, Button, Canvas, Frame
from PIL import Image, ImageTk
```

import cv2 digunakan untuk pengolahan citra digital secara real-time atau untuk penggunaan pengolahan citra secara waktu nyata missal untuk proses *input*, simpan dan menampilkan citra. import os yang memungkinkan untuk melakukan operasi

yang terkait dengan sistem operasi, sedangkan import tensorflow as tf dan from keras.preprocessing import image untuk memudahkan proses debugging dan bertujuan untuk menyederhanakan implementasi algoritma-algoritma *deep learning* dan dapat mengolah dataset agar pelatihan mesin dapat berjalan dengan baik. from keras.applications import VGG16 merupakan modul dalam Pustaka keras yang menyediakan arsitektur yaitu VGG16. from keras.models import model merupakan mengimport model deep learning yang digunakan. from keras.layers import Dense, GlobalAveragePooling2D digunakan untuk mengimport layer yang digunakan seperti dense dan global average pooling 2D. from keras.preprocessing. image import ImageDataGenerator digunakan untuk memproses kumpulan dataset untuk melatih model. import numpy as np digunakan untuk menyediakan fungsi bawaan untuk aljabar linier dan pembuatan bilang acak dan dapat memudahkan untuk mengolah data. import tkinter as tk digunakan untuk membuat tampilan aplikasi GUI dengan komponen yang ada di modul tkinter, from tkinter import Label, Button, Canvas, Frame adalah komponen-komponen yang dipakai pada aplikasi GUI pada penelitian ini. from PIL import Image dan Image Tk merupakan perpustakaan pencitraan python yang menyediakan penerjemahan python.

4.2.2.2. Fungsi Program Latih Dataset

Setelah semua *library* yang digunakan telah di*import* pada *python* maka selanjutnya akan dilakukan percobaan pelatihan dataset nanas sebelum memasuki atau memulai mendeteksi tingkat kematangan buah nanas, dapat dilihat pada program dibawah ini:

```
dataset_dir = "C:\\Users\\HP 14\\Documents\\SKRIPSI 2023\\HASIL\\CNN Nanas
(epoch 10)\\VGG16\\dataset_nanas"
num_classes = 2

base_model = VGG16(weights="imagenet", include_top=False)
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation="relu")(x)
predictions = Dense(num_classes, activation="softmax")(x)
model.compile(optimizer="adam", eror="categorical_crossentropy",
metrics=["accuracy"])
```

```
train_datagen = ImageDataGenerator(
    rescale=1.0 / 255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    fill_mode="nearest",
)
train_generator = train_datagen.flow_from_directory(
    dataset_dir + "/train",
    target_size=(224, 224),
    batch_size=32,
    class_mode="categorical",
)
validation_datagen = ImageDataGenerator(rescale=1.0 / 255)
validation_generator = validation_datagen.flow_from_directory(
    dataset_dir + "/validation",
    target_size=(224, 224),
    batch_size=32,
    class_mode="categorical",
)
model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size,
    epochs=10,
)
model.save("trained_vgg16_model.h5")
uji_datagen = ImageDataGenerator(rescale=1.0 / 255)
uji_generator = uji_datagen.flow_from_directory(
    dataset_dir + "/uji",
    target_size=(224, 224),
    batch_size=1,
    class_mode="categorical",
    shuffle=False,
)
accuracy = model.evaluate(uji_generator)
print(f"Uji accuracy: {accuracy[1]}")
```

dataset_dir = "C:\\Users\\HP 14\\Documents\\SKRIPSI 2023\\HASIL\\CNN Nanas (epoch 20)\\VGG16\\dataset_nanas" merupakan alamat atau *path* dataset nanas yang akan dilatih, sedangkan num_classes = 2 yaitu terdapat dua kelas atau dua macam yang terdapat pada dataset yaitu, matang dan mentah. base_model = VGG16(weights="imagenet", include_top=False) merupakan model VGG16 yang akan digunakan. model. Compile (optimizer= "adam", eror = "categorical_

crossentropy", metrics = ["accuracy"]) merupakan *compile* model. train_generator = train_datagen. flow_ from_ directory (dataset_dir + "/train" merupakan dataset nanas yang akan dilatih pada *directory train.* validation_datagen = Image Data Generator (rescale = 1.0 /255) validation_ generator = alidation_ datagen. flow_ from_ directory (dataset_ dir +255) validation_ generator = alidation_ datagen. flow_from_directory (dataset_dir + "/validation") merupakan dataset nanas yang akan dilatih pada *directory* validasi. epochs=10 yaitu pelatihan akan dilakukan sebanyak dua puluh kali pada data *train* dan validasi. model. Save ("trained_ vgg16_ model.h5") setelah pelatihan *train* dan validasi selesai dilakukan maka akan secara otomatis tersimpan dengan format file "model.h5". Setelah itu akan dilanjutkan Kembali dengan data *test* uji_datagen = Image Data Generator (rescale = 1.0 / 255) uji_ generator = uji_ datagen. flow_ from_ directory (dataset_ dir + "/uji",). accuracy = model. Evaluate (uji_generator) print (f"Uji accuracy: {accuracy[1]}") akan menampilkan hasil pelatihan.

4.2.2.3. Fungsi Program Kendali *Pushbutton*

Setelah melakukan pelatihan dataset nanas pada program, maka akan dilanjutkan dengan program deteksi yang akan ditampilkan oleh GUI yang sudah dirancang, untuk pemrogramannya dapat dilihat dibawah ini.

Fungsi menampilkan GUI dan mengaktifkan *webcam*:

```
# Inisialisasi GUI
root = tk.Tk()
root.title("Deteksi Tingkat Kematangan Nanas Menggunakan Metode CNN (VGG16)")
root.geometry("1000x600")
root.configure(bg="light steel blue")

cap = cv2.VideoCapture(0)

# Frame utama
main_frame = Frame(root, bg="light blue")
main_frame.pack(fill=tk.BOTH, expand=True)

# Frame untuk menampilkan video kamera
video_frame = Frame(main_frame, bg="light blue")
video_frame.pack(side=tk.LEFT, padx=20, pady=20, fill=tk.BOTH, expand=True)

# Canvas untuk tampilan video kamera
```

```
video_canvas = Canvas(video_frame, width=400, height=300, bg="black")
video_canvas.pack()

# Frame untuk hasil tangkapan gambar
captured_images_frame = Frame(video_frame, bg="light blue")
captured_images_frame.pack(pady=10)

# Label judul
title_label = Label(
   action_frame,
   text="Hasil Tingkat Kematangan Nanas",
   bg="light blue",
   fg="black",
   font=("Helvetica", 25),
)
title_label.pack(pady=30, fill=tk.X)

#Frame untuk tombol proses dan reset
action_frame = Frame(main_frame, bg="light blue")
action_frame.pack(side=tk.RIGHT, padx=20, pady=20, fill=tk.Y)

# Tombol untuk menangkap gambar (berbentuk card hijau)
capture_button = Button(
   action_frame,
   text="Capture dan Save Image",
   command=capture_image,
   bg="light steel blue",
   fg="black",
)
capture_button.pack(pady=5, fill=tk.X)

detect_and_classify_frame()

root.mainloop()

# Tombol untuk mereset GUI dan gambar yang ditangkap (berbentuk card hijau)
reset_button = Button(
   action_frame, text="Reset", command=reset_gui, bg="light steel blue", fg="black"
)
reset_button.pack(fill=tk.X)

# Tutup kamera setelah GUI ditutup
cap.release()
cv2.destroyAllWindows()
```

Pada root = tk.TK() digunakan untuk memanggil library tkinter yang akan digunakan untuk mendesain GUI, root.geometry(“1000x600) adalah ukuran GUI yang akan ditampilkan, root.configure(bg=”light steel blue”) merupakan warna

background pada GUI. Pada cap = cv2. Video Capture(0) digunakan untuk menginisialisasi kamera. Pada *command* (#frame) merupakan label, button dan canvas yang digunakan pada aplikasi GUI. Pada detect_and_classify_frame digunakan untuk memulai tampilan video dan deteksi dari kamera. root.mainloop digunakan untuk melakukan perulangan pada pengambilan gambar pada tkinter. cap.release() dan cv2.destoryAllWindows() digunakan untuk menutup kamera setelah GUI ditutup.

4.2.2.4. Fungsi Program untuk Mengambil Gambar Buah Nanas (Capture)

Setelah dapat menampilkan GUI dan mengaktifkan kamera, selanjutnya akan dilakukan penangkapan atau pengambil gambar nanas dengan mengklik tombol *capture* dan *save image*, untuk fungsi pemrogramannya dapat dilihat dibawah ini:

```
save_folder = "data_gambar"
if not os.path.exists(save_folder):
   os.makedirs(save_folder)

capture_button = Button(
   action_frame,
   text="Capture dan Save Image",
   command=capture_image,
   bg="light steel blue",
   fg="black",
)
capture_button.pack(pady=5, fill=tk.X)
captured_count = 0
max_captures = 2
results = []

def capture_image():
   global captured_count
   if captured_count < max_captures:
      ret, frame = cap.read()
      if ret:
         img_name = f"data_gambar/captured_nanas_{captured_count}.jpg"
         cv2.imwrite(img_name, frame)

         img = Image.open(img_name)
         img = img.resize((150, 150))
         img = ImageTk.PhotoImage(img)


captured_images[captured_count].config(image=img)
```

```
        captured_images[captured_count].image = img

        captured_count += 1
        if captured_count >= max_captures:
            process_button.config(state=tk.NORMAL)
```

Pada save_folder = “data_ gambar” merupakan tempat penyimpanan gambar yang akan ditangkap. capture_ count = 0 dan max_ captures = 2 maksudnya adalah pengambilan atau pada saat *capture* perhitungannya dilakukan dari nol dan maksimal pengambilan gambar adalah sebanyak dua kali. def capture_image(): merupakan fungsi untuk melakukan penangkapan gambar. cv2.imwrite(img_name, frame) berfungsi untuk menampilkan gambar yang telah ditangkap ke frame GUI. img = img. resize((150, 150)) merupakan ukuran gambar yang ditangkap dan capture_ count += 1 digunakan untuk melakukan capture Kembali sesuai max_ capture yang ditentukan yaitu dua kali.

4.2.2.5. Fungsi Program untuk Mendeteksi Kematangan Buah Nanas (Proses)

Setelah melakukan pengambilan gambar nanas, maka selanjutnya akan dilakukan tahap proses pada gambar nanas yang akan terdeteksi tingkat kematangan nanas, dapat dilihat fungsi pemrogramannya dibawah ini:

```
model = tf.keras.models.load_model("trained_vgg16_model.h5")

# Fungsi untuk mendeteksi dan mengklasifikasikan gambar
def detect_and_classify_frame():
    ret, frame = cap.read()
    if ret:
            frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        img = image.array_to_img(frame_rgb)
        img = img.resize((120, 120))
        img_array = image.img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0)

        prediction = model.predict(img_array)

        img = cv2.resize(frame, (400, 300))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = Image.fromarray(img)
        img = ImageTk.PhotoImage(image=img)
        video_canvas.create_image(0, 0, anchor=tk.NW, image=img)
```

```
        video_canvas.img = img

        root.after(10, detect_and_classify_frame)

# Fungsi untuk memproses gambar yang ditangkap
def process_images():
    global results
    results = []
    for i in range(max_captures):
        img_path = f"data_gambar/captured_nanas_{i}.jpg"
        img = image.load_img(img_path, target_size=(120, 120))
        img_array = image.img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0)

        prediction = model.predict(img_array)
        results.append(prediction[0][0] * 100)

    avg_result = sum(results) / len(results)
    result_text = f"Persentase Kematangan: {avg_result:.2f}%"
    processed_result_label.config(text=result_text, justify=tk.LEFT)

# Label untuk status kematangan
status_kematangan_label = Label(
    action_frame, text="TINGKAT KEMATANGAN NANAS :", bg="light steel
blue", fg="black"
)
status_kematangan_label.pack(fill=tk.X)

# Frame untuk hasil deteksi dari gambar yang diproses
processed_result_frame = Frame(action_frame, bg="light steel blue")
processed_result_frame.pack(fill=tk.X)
```

Pada model = tf.keras.models.load_model("trained_vgg16_model.h5") digunakan untuk membuat model yang telah dilatih sebelumnya. frame_ rgb = cv2. Cvt Color (frame, cv2.COLOR_BGR2RGB) adalah fungsi untuk melakukan konversi dari BGR menjadi RGB, prediction = model.predict(img_array) adalah dapat memprediksi gambar dari nol dan seterusnya. for i in range(max_captures) dapat mengambil atau penangkapan gambar sebanyak dua kali sesuai program yang dimasukkan. avg_result = sum(results) / len(results) merupakan rata-rata yang diperoleh dari sum/len yang dapat menghitung karakter pada gambar nanas. Setelah melakukan proses deteksi pada nanas maka hasilnya akan ditampilkan pada frame status

kematangan buah nanas dan terdapat persentase tingkat kematangan buah nanas pada frame.

4.2.2.5. Fungsi Program *Reset*

Setelah melakukan percobaan maka akan dilanjutkan dengan percobaan berikutnya, dengan mengklik tombol reset untuk mengulang kembali pengambilan data sampel nanas selanjutnya, dapat dilihat pada fungsi pemrograman dibawah ini:

```
def reset_gui():
    global captured_count, results
    captured_count = 0
    results = []
    for img_label in captured_images:
        img_label.config(image="")
    processed_result_label.config(text="")
    process_button.config(state=tk.DISABLED)
```

Pada global captured_ count, results, captured_ count = 0 berfungsi untuk mengulang Kembali yang dimana akan melakukan pengambilan dari nol Kembali dan process_ button. Config (state=tk.DISABLED) akan melakukan menonaktifkan lalu akan mengulang kembali dari awal.

4.3. Hasil Pengujian Sistem

Pada pengujian ini telah direalisasikan sistem deteksi tingkat kematangan buah nanas menggunakan metode *convolutional neural network* (CNN) dengan algoritma model arsitektur VGG16. Program dalam sistem ini telah dirancang dengan menggunakan *VsCode* dengan bahasa *python* dan divisualisasikan dengan menggunakan GUI yang didesain menggunakan tkinter. Berikut ini merupakan hasil pengujian sistem pada GUI.

4.3.1. Hasil Pengujian Akuisisi Data

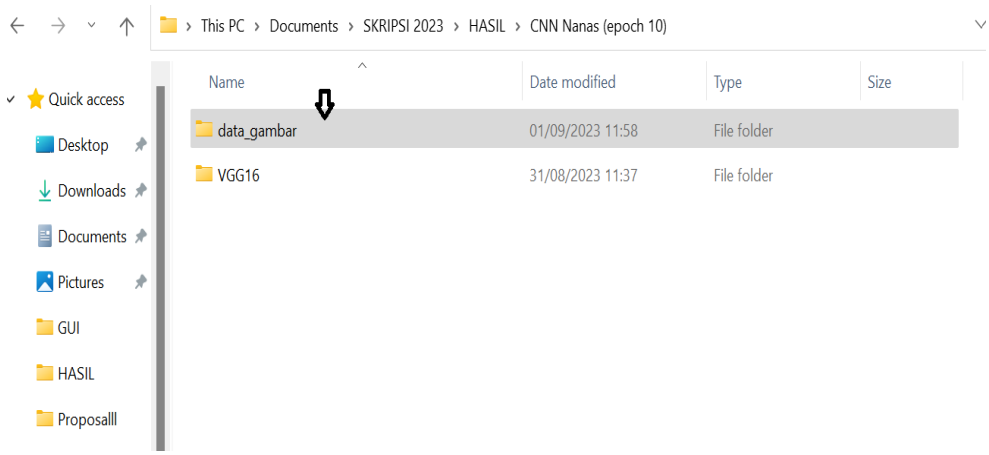
Pengujian yang dilakukan pada sistem deteksi tingkat kematangan buah nanas yaitu menjalankan program yang telah dibuat. Untuk menganalisa citra diawali dengan me *running* program terlebih dahulu maka tampilan GUI akan muncul dan otomatis *webcam* akan menyala. Pada **Gambar 4.5** menunjukkan tampilan GUI (*running*).

Gambar 4.5 Tampilan GUI setelah sistem *running*

Gambar 4.5 menunjukkan video *realtime* dari nanas yang berada di dalam kotak deteksi. Pengujian selanjutnya adalah menangkap gambar dan menyimpan yang telah ditampilkan oleh *webcam* dengan mengklik tombol "*capture* dan *save image*" pada GUI. Tombol ini akan menangkap video yang ditangkap melalui *webcam* sebanyak dua kali dan akan otomatis tersimpan pada dokumen yang telah disediakan. Pada **Gambar 4.6** menunjukkan tampilan GUI *button capture* dan *save image*.

Gambar 4.6 Tampilan GUI *button capture* dan *save image*

Gambar 4.6 merupakan tampilan dari hasil tangkapan gambar nanas dan akan secara langsung tersimpan pada dokumen. Pada **Gambar 4.7** menunjukkan tampilan tempat penyimpanan sampel nanas.

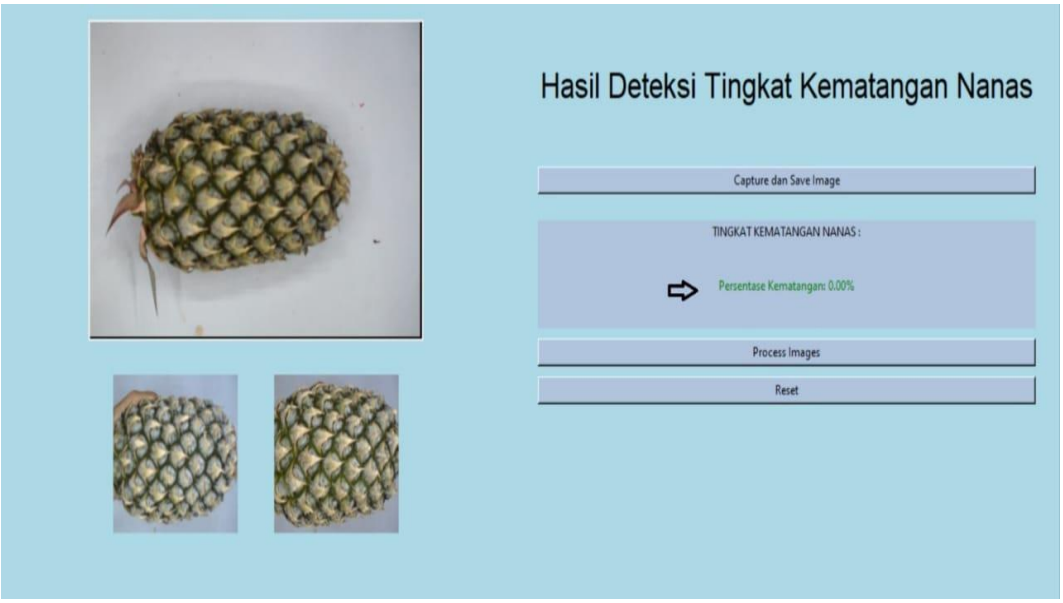


Gambar 4.7 Tempat penyimpanan sampel nanas pada dokumen

Selanjutnya dilakukan pengujian pada tombol "*process images*". Tampilan GUI dari proses dengan metode CNN akan menampilkan nilai kuantifikasi tingkat kematangan nanas yang akan ditampilkan GUI "persentase kematangan nanas". Pada **Gambar 4.8** menunjukkan tampilan GUI hasil *process images*.

Gambar 4.8 Tampilan GUI hasil dari *process images*

Setelah melakukan proses pada sampel nanas ke satu maka akan ditampilkan nilai tingkat kematangan nanas. Pada **Gambar 4.9** menunjukkan tampilan GUI hasil deteksi.



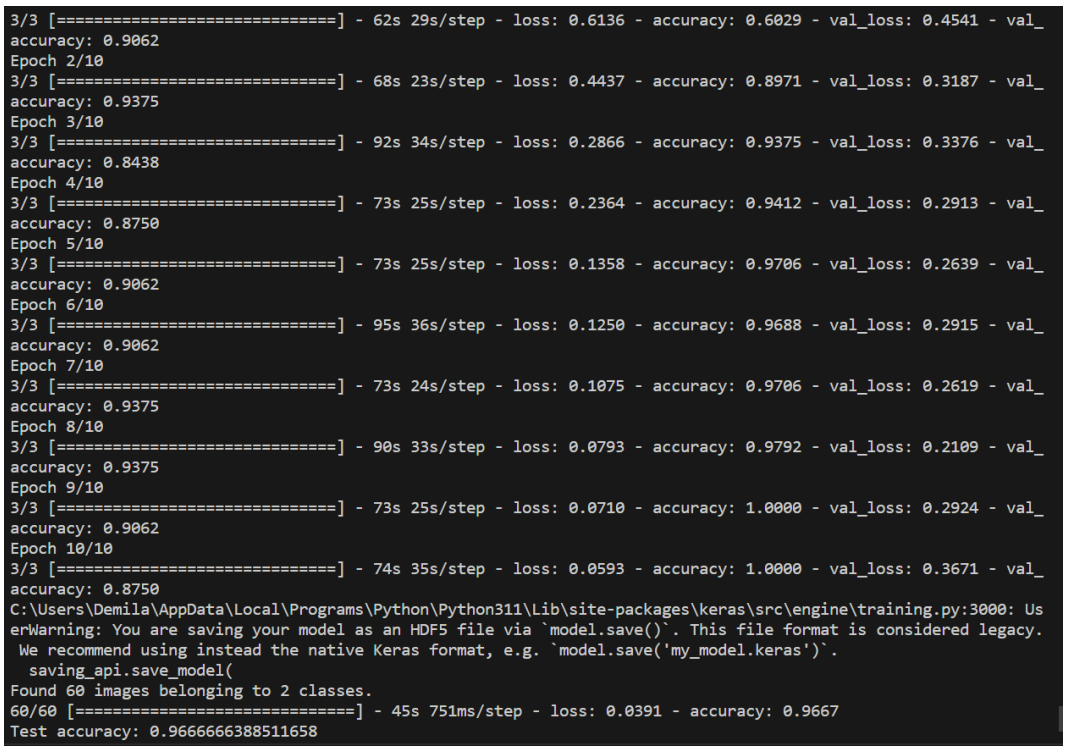
Gambar 4.9 Tampilan GUI hasil deteksi

Setelah melakukan pengujian maka akan dilanjutkan pengujian berikutnya dengan mengklik tombol "*reset*" dimana data dan gambar hasil *capture* buah nanas sebelumnya akan hilang. Pada **Gambar 4.10** menunjukkan tampilan GUI *button reset*.

Gambar 4.10 Tampilan GUI *button reset*

4.3.2. Model hasil pelatihan

Setelah melakukan beberapa proses dalam algoritma *convolutional neural network* (CNN) didapatkan hasil latih, validasi dan *test*. Proses ini menggunakan jumlah *epoch* 10. **Gambar 4.11** menunjukkan hasil pelatihan dataset pada terminal python.



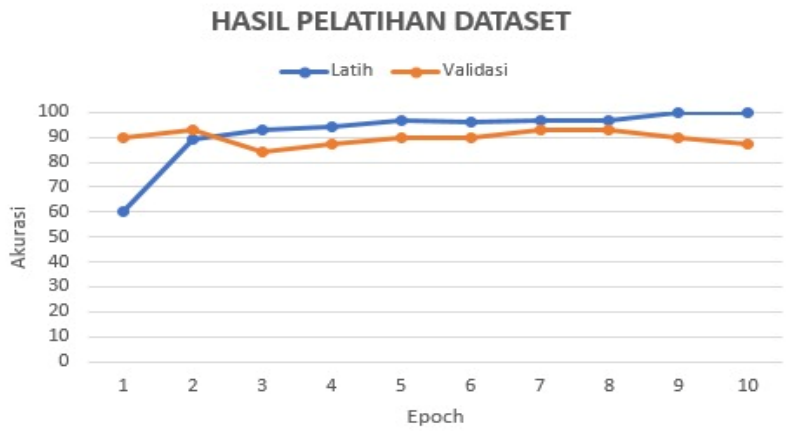
Gambar 4.11 Hasil pelatihan dataset pada tampilan terminal python

Pada **Tabel 4.2** menunjukkan hasil pelatihan dataset buah nanas (latih, validasi dan uji).

Tabel 4.2 Hasil Pelatihan dataset nanas dengan arsitektur VGG16

Epoch	Waktu (*s*)	Latih		Validasi	
		Akurasi (%)	Eror (%)	Akurasi (%)	Eror (%)
1	62	60	61	90	45
2	68	89	44	93	31
3	92	93	28	84	33
4	73	94	23	87	29
5	73	97	13	90	26
6	95	96	12	90	29
7	73	97	10	93	26
8	90	97	7	93	21
9	73	100	7	90	29
10	74	100	5	87	36
Uji	45	**Akurasi:**	96%	**Eror:**	3%

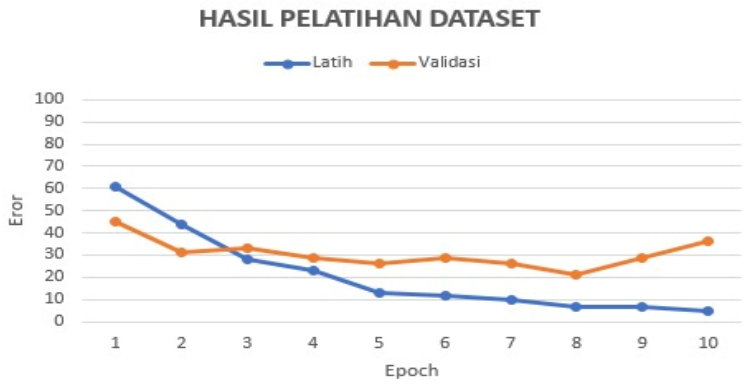
Pada pelatihan dataset dengan menggunakan algoritma VGG16, menghasilkan nilai akhir akurasi pada data uji sebesar 96%. Hasil tersebut cukup tinggi untuk tahap pelatihan ini yang akan dilanjutkan ke tahap deteksi tingkat kematangan nanas. Adapun bentuk grafik dari hasil pelatihan dataset nanas pada **Gambar 4.12** dan **Gambar 4.13**.



Gambar 4.12 Grafik nilai akurasi hasil pelatihan dataset

Pada **Gambar 4.12** terdapat sebuah grafik akurasi pada dataset latih dan validasi. Dapat dikatakan semakin tinggi akurasi semakin bagus hasil yang didapatkan dan semakin kecil nilai eror. Dataset latih pada epoch ke 1 hingga ke 5 mengalami

peningkatan akurasi namun pada epoch ke 6 mengalami penurunan akurasi sebesar 96%, pada epoch ke 7 hingga ke 10 megalami peningkatan hingga 100%. Dataset validasi pada epoch ke 1 dan 2 mengalami peningkatan, epoch ke 3 mengalami penurunan hingga 84% dan epoch ke 4 sampai ke 8 mengalami peningkatan kembali, epoch ke 9 dan 10 mengalami penurunan Kembali hingga mendapatkan nilai 87%. Namun pada **Tabel 4.2** mendapatkan hasil akhir pelatihan data uji sebesar 96% akurasi.



Gambar 4.13 Grafik nilai eror hasil pelatihan dataset

Pada **Gambar 4.13** terdapat sebuah grafik eror pada dataset latih dan validasi. Dapat dikatakan semakin kecil hasil nilai eror semakin bagus akurasi yang di diperoleh. Pada dataset latih selama melewati pelatihan mengalami penurunan eror sebesar 5%. Beda hal nya dengan dataset validasi yang mengalami naik turun pada nilai eror. Epoch 1 dan 2 mengalami penurunan, epoch ke 3 mengalami kenaikan eror sebesar 33%, epoch ke 4 dan 5 mengalami penurunan, epoch 6 mengalami kenaikan eror kembali sebesar 29%, epoch ke 7 dan 8 mengalami penurunan Kembali, epoch ke 9 dan 10 mengalami kenaikan eror sebesar 36%. Dapat dikatakan nilai eror tersebut bisa dikatakan masih cukup besar erornya. Namun pada **Tabel 4.2** mendapatkan hasil akhir pelatihan data uji sebesar 3% eror.

4.3.3. Analisis Persentase Kematangan Nanas

Pada penelitian ini dilakukan analisis untuk mengetahui tingkat kematangan buah nanas dengan cara melihat peningkatan kematangan dari buah nanas yang telah di uji. Pada **Gambar 4.14, Gambar 4.15** dan **Gambar 4.16** merupakan nilai tingkat kematangan nanas pada tampilan GUI.

Gambar 4. 14 Hasil deteksi tingkat kematangan nanas (0%)

Gambar 4. 15 Hasil deteksi tingkat kematangan nanas (50%)

Gambar 4.16 Hasil deteksi tingkat kematangan nanas (100%)

Pada **Tabel 4.3** menunjukkan hasil keseluruhan deteksi kematangan nanas yang dilakukan.

Tabel 4.3 Persentase kematangan nanas hasil manual dan hasil deteksi

Gambar	Deteksi	Manual	Persentase Akurasi Kematangan Nanas (%)
	Persentase Tingkat Kematangan Nanas (%)	Persentase Tingkat Kematangan Nanas (%)	
1	0	6	0
2	0	4	0
3	0	4	0
4	0,01	2	1
5	0	3	0
6	0,01	5	1
7	10,99	7	43
8	2,49	6	41
9	50	73	68
10	22,74	5	94
11	49,96	62	81
12	56,31	76	74
13	46,59	52	89
14	50,44	68	74
15	48,08	71	68
16	50	69	69
17	50	73	69
18	61,79	77	81
19	50	72	70
20	66,03	77	86
21	100	82	78
22	100	100	100
23	100	96	96
24	100	83	80

Berdasarkan **Tabel 4.3** diperoleh hasil persentase tingkat kematangan nanas pada sistem dan perhitungan manual dengan menggunakan persamaan 3.6, serta nilai akurasi tingkat kematangan buah nanas dengan menggunakan persamaan 3.8. Alat deteksi kematangan ini memiliki persentase akurasi kematangan yang bervariasi sehingga akurasi yang paling tinggi terdapat pada data ke 22 sebesar 100%. Beberapa data yang tingkat kematangannya bervariasi dikarenakan adanya eror yang disebabkan karena beberapa permasalahan yang muncul dalam pengambilan gambar seperti hasil tangkap *webcam* yang kurang fokus, terdapat beberapa bagian yang tidak dibutuhkan ditangkap oleh kamera atau derau, posisi nanas yang miring dan lampu led kurang stabil dan kualitas *webcam* yang rendah mengakibatkan hasil

gambar nanas kurang optimal. Dari sisi metode yang digunakan yaitu CNN telah terdapat proses *noise* untuk menghilangkan data yang tidak dibutuhkan.

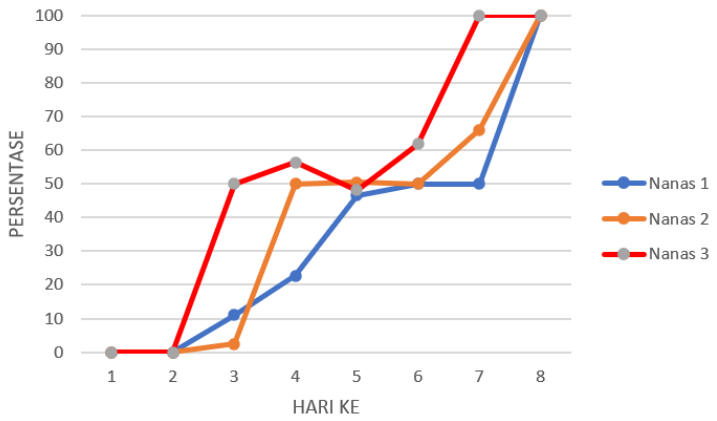
Untuk tingkat akurasi dari sistem ini jika dibandingkan dengan penelitian sebelumnya dengan akurasi 93% menggunakan teknik *thresholding*. ia menggunakan 3 sampel nanas untuk mengklasifikasi nanas dalam keadaan mentah, setengah matang dan matang. Dari perbandingan akurasi yang dilakukan Yanti *et al.,* (2022), dapat dilihat bahwa penelitian ini masih memiliki banyak kelemahan dan beberapa ketidakstabilan yang dapat dilihat pada **Tabel 4.4**.

Tabel 4.4 Data tingkat persentase hasil sistem deteksi

Sampel	Pengambilan ke-	Persentase Tingkat Kematangan Nanas (%)
1	1	0
	2	0.01
	3	10.99
	4	22.74
	5	46.59
	6	50
	7	50
	8	100
2	1	0
	2	0
	3	2.49
	4	49.96
	5	50.44
	6	50
	7	66.03
	8	100
3	1	0
	2	0.01
	3	50
	4	56.31
	5	48.08
	6	61.79
	7	100
	8	100

Ketidakstabilan pada alat deteksi kematangan nanas dalam menguji alat deteksi digunakan 3 sampel nanas dengan tingkat kematangan nanas yang sama.

Pengambilan gambar untuk satu sampel uji dilakukan sebanyak delapan kali pengambilan gambar. Hasil yang dapat dilihat terdapat adanya pengurangan dan penambahan persentase pada tingkat kematangan nanas, walaupun tidak terlalu signifikan. Dari hasil deteksi pada **Tabel 4.4** disajikan juga dalam bentuk grafik pada **Gambar 4.17**.



Gambar 4.17 Grafik hasil deteksi tingkat kematangan nanas

Pada **Gambar 4.17** terdapat sebuah grafik yang menunjukkan tingkat kematangan 3 sampel nanas selama 8 hari, dari nanas mentah hingga ke matang. Pada nanas 1 mengalami tingkat kematangan nanas secara berturut hingga hari ke 8 sebesar 100%. Beda halnya dengan sampel nanas ke 2 mengalami kenaikan dan penurunan kematangan nanas, hari ke 1 dan ke 2 mendapatkan nilai sistem deteksi sebesar 0%, namun pada hari ke 3 hingga hari ke 5 mengalami peningkatan kematangan sebesar 50.44%, namun hari ke 6 mengalami penurunan kematangan sebesar 50%. Tetapi hari selanjutnya mengalami peningkatan kematangan hingga 100%. Sama hal nya dengan nanas ke 2 dimana nanas ke 3 mengalami kenaikan dan penurunan tingkat kematangan nanas, hari ke 1 hingga hari ke 4 mengalami peningkatan kematangan sebesar 56.31%, namun pada hari ke 5 mengalami penurunan sebesar 48.08%, tetapi pada hari selanjutnya mengalami kenaikan tingkat kematangan sebesar 100%.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Berdasarkan hasil penelitian dan analisis data yang dilakukan selama 8 hari pengambilan sampel, dengan 3 sampel buah nanas mendeteksi tingkat kematangan buah nanas masih mentah hingga ke matang, dapat disimpulkan bahwa telah terealisasi suatu sistem deteksi tingkat kematangan buah nanas menggunakan metode *Convolutional Neural Network* (CNN) dengan algoritma model *Visual Geometry Group* (VGG) 16. Akurasi yang dihasilkan pada pelatihan dataset nanas sebesar 96%. Sedangkan akurasi pada sistem deteksi tingkat kematangan buah nanas sebesar 100%.

5.2. Saran

Saran untuk pengembangan penelitian selanjutnya yaitu dapat menggunakan semua arsitektur yang terdapat pada model CNN seperti AlexNet, ResNet, GoogleNet, Inception dan Squeezenet, supaya dapat mengetahui perbandingan arsitektur mana yang sangat akurat untuk digunakan pada penelitian yang berhubungan dengan *imageNet.*

DAFTAR PUSTAKA

Andono, P. N., & Sutojo, T. 2017. *Pengolahan Citra Digital*. CV. Andi Offset. Yogyakarta.

Asnor, J. I., Rosnah, S., Wan, Z. W. H., & Badrul, H. A. B. 2013. Pineapple Maturity Recognition Using RGB Extraction. *International Journal of Electrical and Computer Engineering*, 7(6), 3-4.

Astrid, M. 2019. *Pooling Layer*. Diakses pada 21 Juli 2019 dari https: //youtu.be/ ryMcIrkiYvQ

Azman, Aizuddin, A., & Ismail, F. S. 2017. Convolutional Neural Network for Optimal Pineapple Harvesting. *Journal of Electrical Engineering*, 16(2), 4-5.

Candra, P. N., & Aditya, P. 2020. Klasifikasi Gambar Asli dan Manipulasi Menggunakan Error Level Analysis (ELA) sebagai Proses Komputasi Metode Convolutional Neural Network (CNN). *Journal of Informatics and Computer Science* , 2(1), 9-10.

Clinton, R. M., & Sengkey, R. 2019. Purwarupa Sistem Daftar Pelanggaran Lalu lintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 8(3), 181-192.

Delfi, F. D. 2021. Pengaruh Penambahan Sari Buah Nanas (Ananas Comosus (L) Merr) Terhadap Kadar Protein Dan Aktivitas Antioksidan Dahid Susu Kerbau Sebagai Alternatif Peningkatan Sistem Imun. *Skripsi*. Biologi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Perintis Indonesia, Bukit Tinggi, Padang.

Desy, F. T., Surtono, A., Supriyanto, A., & Junaidi, J. 2020. Rancang Bangun Purwarupa Pemilah Nanas Berdasarkan Tingkat Kematangan Menggunakan Mikrokontroler Blue Pill STM32F103C8T6. *Journal of Energy, Material, and Instrumentation Technology*, 1(3), 81-90.

Indian, T. 2021. Lapisan Yang Terhubung Sepenuhnya dalam Jaringan Syaraf Konvolusional. Diakses pada 08 Maret 2021 dari https: //indiantechwarrior. com/ fully-connected-layers-in-convolutional-neural-networks/

Kaewapichai, W., Kaewtrakulpong, P., Prateepasen, A., & Khongkraphan, K. 2007. Fitting A Pineapple Model For Automatic Maturity Grading. *Journal of Energy*, 1(1), 25-27.

Kusumanto, R.D., & Tompunu, A. N. 2011. Pengolahan Citra Digital untuk Mendeteksi Objek Menggunakan Pengolahan Warna Model Normalisasi RGB. *Jurnal Semantik*, 1(1), 4-5.

Lubis, E. R. 2020. *Budidaya Nanas.* Bhuana Ilmu Populer. Jakarta.

Luthfieta, N. Z. 2016. Perhitungan Matematika Algoritma CNN. Diakses pada 16 Oktober 2016 dari https://youtu.be/vWZCMmfa980?si=g1ZlxfLsAzJEVxO4

Malhotra, A. 2018. *Tutorial on Feedforward Neural Network.* Diakses pada tanggal 1 Februari 2018 dari https://medium.com /@akankshamalhotra24/tutorial-on-feedforward-neural-network-part-1-659eeff574c3

Mostafa, S., & Wu, F. 2021. Diagnosis Gangguan Spektrum Autisme dengan Auto Encoder Konvolusional dan Gambar MRI Struktural. *Jurnal Pencitraan dan Analisis Sinyal*, 1(1), 9-21.

Mohri. 2012. *Foundation of* Machine *Learning.* Cambridge: MIT Press.

Nawawi, M. A. A., & Ismail, F. S. 2017. Simulation and Segmentation Techniques for Crop Maturity Identification of Pineapple Fruit. 17^{th} *Asia Simulation Coference.* Faculity of Electical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia.

Nurviyanto, I. 2012. Deteksi Wajah Dengan Metode Viola Jones Pada OpenCV Menggunakan Pengolahan Python. *Skripsi*. Ilmu Komputer, Fakultas Matematika dan Ilmu Prngetahuan Alam Universitas Muhammadiyah Surakarta, Surakarta, Jawa Tengah.

Pannu, A., & Tech. 2015. Artificial Intelligence and Application in Different Areas. *International Journal of Engineering and Innovative Technology (IJEIT),* 4.

Patel, K. 2019. *Convolutional Neural Networks-A Beginner's Guide*. Diakses pada tanggal 08 September 2019 dari https: //towardsdatasciense. com/ convolutional- neural- networks-a- neginners- guide- implementing-a-mnist-hand-written-digit-8aa60330d022

Pathaveerat, S., Terdwong, W. A., & Phaungsombut, A. 2008. Multivariate Data Analysis for Classification of Pineapple Maturity. *Journal of Food Engineering.* Diakses pada 12 April 2008 dari https://doi.Org/10.1016/j.j foodeng

Prasetyo, N. A. 2021. Rancang Bangun Sistem Identifikasi Tingkat Kematangan Buah Nanas Secara Non-Destruktif Berbasis *Computer Vision. Skripsi.* Fisika, Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Lampung. Bandar Lampung, Lampung.

Purba, Y. B. E., Saragih, N. F., Silalahi, A. P., Sitepu, S., & Gea, A. 2022. Perancangan Alat Pendeteksi Kematangan Buah Nanas Dengan Menggunakan Mikrokontroler Dengan Metode Convolutional Neural Network (CNN). *Jurnal Ilmiah Teknik Informatika Matematika,* 2(1), 13-21.

Putra, A. R. 2016. Klasifikasi Citra Menggunakan *Convolutional Neural Network* (CNN) pada *Caltech 101*. *Jurnal Teknik ITS*, 5(1), 1-6.

Rahman, M. F. 2016. Deteksi dan Pemutuan Nanas Berbasis Computer Vision. *Skripsi*. Departemen Teknik Mesin Dan Biosistem Fakultas Teknologi Pertanian, Institut Pertanian Bogor. Bogor.

Rismiyati & Ardytha, L. 2021. VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Jurnal Informatika dan Teknologi Informasi*, 18(1), 42-43.

Sadya, S. 2022. *Indonesia Produksi Nanas Hingga 2,89 Juta Ton Pada 2021*. Diakses pada 5 Oktober 2022 dari https: //dataindonesia.id/ sektor-rill/ detail/indonesia-produksi-nanas-hingga-289-juta-ton-pada-2021

Saputra, M., Kusrini, K., & Kurniawan, M. P. 2020. Identifikasi Mutu Biji Kopi Arabika Berdasarkan Cacat Dengan Teknik *Convolutional Neural Network*. *Jurnal Teknologi Informasi Dan Komunikasi*, 10(1), 27-35.

Senna, S. 2018. *Pengenalan Deep Learning Part 1: Neural Network.* Diambil Kembali dari Medium: https://medium.com/@samuelsena/pengenalan- deep-learning-8fbb7d8028ac

Silaban, I., & Rahmanisa, S. 2016. Pengaruh Enzim Bromelin Buah Nanas (ananas comosus) Terhadap Awal Kehamilan. *Jurnal Majority*, 5(4), 80-85.

Trivusi. 2022. Pengertian dan Cara Kerja Algoritma Convolutional Neural Network. Diakses pada 28 Juli 2022 dari https: //www. trivusi. web.id/ 2022/04/algoritma-cnn.html?m=1

Sulandari, W., Hartatik, & Sudibyo, N. A. 2014. Metode Statistika untuk Kimia: Analisis Data dengan Excel. BIPTEK INDONESIA. Sukoharjo.

Wicaksono, A. D. P., & Henri, T. 2023. Hybrid Algoritma Vgg16-Net Dengan Support Vector Machine untuk Klasifikasi Jenis Buah dan Sayuran. *Jurnal Teknologi Informasi dan Multimedia*, 5(2), 57.

Winastia, B. 2011. Analisa Asam Amino Pada Enzim Bromelin Dalam Buah Nanas (Ananas Comosus) Menggunakan Spektrofotometer. *Skripsi*. Teknik Informatika, Fakultas Teknik, Universitas Diponegoro Semarang, Semarang, Jawa Tengah.

Yanti, W. P. 2022. Deteksi Kematangan Buah Nanas Berdasarkan Tingkat Kekuningan Mata Nanas Dengan Menggunakan Metode *Thresholding. Skripsi.* Fisika, Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Lampung, Bandar Lampung, Lampung.

Yusuf, A., Wihandika, R. C., & Dewi, C. 2019. Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(11), 3-4.