APLIKASI KLASIFIKASI PENYAKIT PADA DAUN DAN BUAH JAMBU BIJI BERBASIS *MOBILE* DENGAN MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

**SKRIPSI**

**Diajukan Untuk Memenuhi Salah Satu Syarat**

**Memperoleh Gelar Strata Satu (S1) Teknik Informatika**



**Oleh:**

**FADLAN SAYYIDUL ANAM**

**NIM: 191351144**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI WASTUKANCANA**

**PURWAKARTA**

**2023**

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

APLIKASI KLASIFIKASI PENYAKIT PADA DAUN DAN BUAH JAMBU BIJI BERBASIS *MOBILE* DENGAN MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

**Oleh:**

**Fadlan Sayyidul Anam**

**191351144**

**Program Studi Teknik Informatika**

Sekolah Tinggi Teknologi Wastukancana Purwakarta

Menyetujui,

Tim Pembimbing

Tanggal ………………

|  |  |
| --- | --- |
| Pembimbing Utama  Muhammad Rafi Muttaqin, S.Kom., M.Kom.  NIDN: 0420088703 | Pembimbing Pendamping  Yudhi Raymond Ramadhan, S.Si., M.Kom.  NIDN: 0422078006 |

Mengetahui,

Ketua Program Studi Teknik Informatika

Teguh Imam Hermanto, M.Kom.

NIDN: 0425049102

**LEMBAR PENGESAHAN DOSEN PENGUJI**

APLIKASI KLASIFIKASI PENYAKIT PADA DAUN DAN BUAH JAMBU BIJI BERBASIS *MOBILE* DENGAN MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

**Oleh:**

**Fadlan Sayyidul Anam**

**191351144**

**Program Studi Teknik Informatika**

Sekolah Tinggi Teknologi Wastukancana Purwakarta

Menyetujui,

Tim Penguji

Tanggal ……………

|  |  |
| --- | --- |
| Ketua Penguji  Nama  NIDN: | Anggota Penguji  Nama  NIDN: |

|  |  |
| --- | --- |
| Pembimbing Utama  Muhammad Rafi Muttaqin, S.Kom., M.Kom.  NIDN: 0420088703 | Pembimbing Pendamping  Yudhi Raymond Ramadhan, S.Si., M.Kom.  NIDN: 0422078006 |

Mengetahui,

Ketua Program Studi Teknik Informatika

Teguh Imam Hermanto, M.Kom.

NIDN: 0425049102

**PERNYATAAN**

Saya Fadlan Sayyidul Anam menyatakan dengan sesungguhnya, bahwa karya tulis yang berjudul **“Aplikasi Klasifikasi Penyakit Pada Daun dan Buah Jambu Biji Berbasis *Mobile* Dengan Menggunakan Metode *Convolutional Neural Network* (CNN)”** adalah benar hasil karya sendiri. Serta tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu perguruan tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis/diterbitkan orang lain, kecuali yang tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Dinyatakan tanggal …..…….

Fadlan Sayyidul Anam

NIM. 191351144

# ABSTRAK

APLIKASI KLASIFIKASI PENYAKIT PADA DAUN DAN BUAH JAMBU BIJI BERBASIS *MOBILE* DENGAN MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

Oleh:

**Fadlan Sayyidul Anam**

**191351144**

**Program Studi Teknik Informatika**

Jambu biji merupakan komoditas tanaman *frutikultura* dengan jumlah produksi berperingkat ke-8 di Jawa Barat pada tahun 2021 dengan total 692 ribu kuintal. Namun produksi jambu biji tahun 2021 mengalami penurunan sebesar 12,82% dibandingkan produksi tahun 2020. Penelitian ini bertujuan untuk membantu klasifikasi penyakit pada daun dan buah jambu biji sehingga dapat dilakukan penanganan yang tepat terhadap daun dan buah jambu biji yang terkena penyakit dengan berdasarkan citra digital menggunakan aplikasi berbasis *mobile.*

Penelitian ini menggunakan teknologi *deep learning* yang akan melakukan pembelajaran terhadap *dataset* penyakit pada daun dan buah jambu yang telah diberikan label atau disebut *supervised learning*. Untuk melakukan klasifikasi citra digital dibutuhkan algoritma *Convolutional Neural Network* (CNN)dengan arsitektur *MobileNetV2*. Setelah model selesai dibuat maka akan diterapkan menjadi aplikasi *mobile* menggunakan bahasa pemrograman *Kotlin* Kemudian metode penelitian yang digunakan pada penelitian ini adalah *Cross Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment*.

Berdasarkan hasil penelitian ini *model* daun jambu biji memiliki hasil evaluasi model yang sangat baik dengan rincian sebagai berikut: *training accuracy* sebesar 99.6%, *validation accuracy* sebesar 100%, *training loss* sebesar 3.2%, dan *validation loss* sebesar 3.1%. Hasil *confusion matrix model* daun jambu memiliki akurasi 100% dari jumlah data validasi sebanyak 63 data. Sedangkan *model* buah jambu biji memerlukan *dropout* sebesar 0.2 dan *kernel regularizers* L2 sebesar 0.01 untuk mengurangi *overfitting* dengan rincian sebagai berikut: *training accuracy* 98,8*%* dan *validation accuracy* 91.6%, kemudian *training loss* sebesar 19.1%, dan *validation loss* sebesar 38.6%. Hasil confusion matrix model buah jambu memiliki akurasi 91.6% dari jumlah data validasi sebanyak 84 data. Kemudian kedua *model* yang telah dibuat berhasil diterapkan menjadi aplikasi berbasis *mobile.*

**Kata kunci :** *Deep learning, Supervised Learning, Convolutional Neural Network, MobileNetV2,* Jambu biji.

# *ABSTRACT*

***CLASSIFICATION APPLICATION OF DISEASES IN GUAVA LEAVES AND FRUITS BASED ON MOBILE USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD***

By:

**Fadlan Sayyidul Anam**

**191351144**

***Informatics Engineering***

*Guava is a fruticulture commodity with the 8th ranked production amount in West Java in 2021 with a total of 692 thousand quintals. However, guava production in 2021 has decreased by 12.82% compared to production in 2020. This research aims to help classify diseases on guava leaves and fruits so that appropriate treatment can be carried out on guava leaves and fruits affected by disease based on digital images using mobile-based applications.*

*This research uses deep learning technology that will perform learning on a dataset of diseases on guava leaves and fruits that have been labeled or called supervised learning. To classify digital images, a Convolutional Neural Network (CNN) algorithm with MobileNetV2 architecture is needed. After the model is completed, it will be applied into a mobile application using the Kotlin programming language. Then the research method used in this research is CRISP-DM.*

*Based on the results of this study, the guava leaf model has excellent model evaluation results with the following details: training accuracy of 99.6%, validation accuracy of 100%, training loss of 3.2%, and validation loss of 3.1%. The confusion matrix results of the guava leaf model have 100% accuracy from the total validation data of 63 data. While the guava fruit model requires a dropout of 0.2 and L2 kernel regularizers of 0.01 to reduce overfitting with the following details: training accuracy 98.8% and validation accuracy 91.6%, then training loss of 19.1%, and validation loss of 38.6%. The confusion matrix results of the guava fruit model have an accuracy of 91.6% of the total validation data of 84 data. Then the two models that have been made are successfully applied into mobile-based applications.*

***Keywords:*** *Deep learning, Supervised Learning, Convolutional Neural Network, MobileNetV2, Guava*

# PEDOMAN PENGGUNAAN SKRIPSI

Skripsi S1 yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Sekolah Tinggi Teknologi Wastukancana, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada pengarang dengan mengikuti aturan HaKI yang berlaku di Sekolah Tinggi Teknologi Wastukancana. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin pengarang dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya.

Memperbanyak atau menerbitkan sebagian atau seluruh Skripsi haruslah seizin Ketua Sekolah Tinggi Teknologi Wastukancana Purwakarta.

# KATA PENGANTAR

Puja dan puji syukur kehadirat Allah SWT yang Maha Pengasih lagi Maha Panyayang, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya, sehingga penulis dapat merampungkan skripsi yang berjudul “Aplikasi Klasifikasi Penyakit Pada Daun dan Buah Jambu Biji Berbasis *Mobile* Dengan Menggunakan Metode *Convolutional Neural Network* (CNN)”.

Melalui kesempatan yang berharga dan dengan menyelesaikan skripsi ini penulis banyak mendapatkan arahan, bimbingan, doa, serta dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh sebab itu, penulis ingin mengucapkan terimakasih kepada yang terhormat :

1. Bapak Apang Djafar Shieddieque, S.T., M.T. Selaku Ketua Sekolah Tinggi Teknologi Wastukancana Purwakarta.
2. Bapak Teguh Imam Hermanto, M.Kom. Selaku Kaprodi Teknik Informatika.
3. Bapak Muhammad Rafi Muttaqin, S.Kom., M.Kom. Selaku pembimbing utama.
4. Bapak Yudhi Raymond Ramadhan, S.Si., M.Kom. Selaku pembimbing pendamping.
5. Ibu, Bapak, Kakak, dan seluruh Keluarga Besar yang telah mendukung sepenuh hati dan memberikan dukungan baik secara moril maupun materi.
6. Serta semua pihak lainnya yang tidak bisa dituliskan penulis satu per satu yang telah membantu selama penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna dan masih memiliki kekurangan. Penulis mengharapkan kritik dan saran pembaca untuk menyempurnakan skripsi ini. Semoga skripsi ini bermanfaat bagi semua pihak yang membutuhkannya. Akhir kata, semoga Allah SWT melimpahkan rahmat dan hidayah-Nya kepada kita semua. Amin ya robbal ‘alamiin.

Purwakarta

Penulis

# DAFTAR ISI

[ABSTRAK i](#_Toc138684948)

[*ABSTRACT* ii](#_Toc138684949)

[PEDOMAN PENGGUNAAN SKRIPSI iii](#_Toc138684950)

[KATA PENGANTAR iv](#_Toc138684951)

[DAFTAR ISI v](#_Toc138684952)

[DAFTAR GAMBAR viii](#_Toc138684953)

[DAFTAR TABEL x](#_Toc138684954)

[DAFTAR LAMPIRAN xi](#_Toc138684955)

[BAB I PENDAHULUAN 1](#_Toc138684956)

[1.1 Latar Belakang 1](#_Toc138684957)

[1.2 Rumusan Masalah 3](#_Toc138684958)

[1.3 Batasan Masalah 3](#_Toc138684959)

[1.4 Tujuan Penelitian 3](#_Toc138684960)

[1.5 Manfaat Penelitian 4](#_Toc138684961)

[1.6 Sistematika Penulisan 4](#_Toc138684962)

[BAB II TINJAUAN PUSTAKA 6](#_Toc138684963)

[2.1 Implementasi 6](#_Toc138684964)

[2.2 *Cross Industry Standard Process for Data Mining* (CRISP-DM) 6](#_Toc138684965)

[2.3 Aplikasi 8](#_Toc138684966)

[2.4 Pertanian 8](#_Toc138684967)

[2.5 Hortikultura 8](#_Toc138684968)

[2.6 Penyakit Tanaman 8](#_Toc138684969)

[2.7 Jambu Biji 9](#_Toc138684970)

[2.7.1 Syarat Tumbuh Jambu Biji 9](#_Toc138684971)

[2.7.2 Penyakit Jambu Biji 11](#_Toc138684972)

[2.8 *Machine Learning* 15](#_Toc138684973)

[2.9 *Deep Learning* 16](#_Toc138684974)

[2.10 *Convolutional Neural Network* (CNN) 17](#_Toc138684975)

[2.10.1 *Feature Learning* 18](#_Toc138684976)

[2.10.2 *Classification* 22](#_Toc138684977)

[2.11 *MobileNet* 23](#_Toc138684978)

[2.12 *Tensorflow* 24](#_Toc138684979)

[2.13 *Tensorflow Lite* 25](#_Toc138684980)

[2.14 *Android* 25](#_Toc138684981)

[2.15 *Android Studio* 25](#_Toc138684982)

[2.16 *Integrated Development Environment* (IDE) 25](#_Toc138684983)

[2.17 *Kotlin* 26](#_Toc138684984)

[2.18 Penelitian Terdahulu 26](#_Toc138684985)

[BAB III METODOLOGI PENELITIAN 29](#_Toc138684986)

[3.1 Kerangka Berpikir 29](#_Toc138684987)

[3.2 *Business Understanding* 30](#_Toc138684988)

[3.3 *Data Understanding* 30](#_Toc138684989)

[3.4 *Data Preparation* 31](#_Toc138684990)

[3.5 *Modeling* 33](#_Toc138684991)

[3.6 *Evaluation* 33](#_Toc138684992)

[3.7 *Deployment* 34](#_Toc138684993)

[BAB IV PENGOLAHAN DATA DAN PEMBAHASAN 35](#_Toc138684994)

[4.1 *Data Preparation* 35](#_Toc138684995)

[4.3 *Modelling* 37](#_Toc138684996)

[4.3.1 Proses *modelling* 38](#_Toc138684997)

[4.3.2 Penentuan parameter *modelling* 40](#_Toc138684998)

[4.5 *Evaluation* 44](#_Toc138684999)

[4.6 *Deployment* 47](#_Toc138685000)

[BAB V KESIMPULAN DAN SARAN 51](#_Toc138685001)

[5.1 Kesimpulan 51](#_Toc138685002)

[5.2 Saran 51](#_Toc138685003)

[DAFTAR PUSTAKA 53](#_Toc138685004)

[LAMPIRAN 58](#_Toc138685005)

# DAFTAR GAMBAR

[Gambar 2.1 Diagram CRISP-DM 7](#_Toc138145859)

[Gambar 2.2 Penyakit karat merah 11](#_Toc138145860)

[Gambar 2. 3 Penyakit daun berlubang / *dot* 12](#_Toc138145861)

[Gambar 2.4 Penyakit *scab* 13](#_Toc138145862)

[Gambar 2.5 Penyakit *stylar end rot* 14](#_Toc138145863)

[Gambar 2.6 Penyakit *phytopthora palmivora* 14](#_Toc138145864)

[Gambar 2.7 Arsitektur *convolutional neural network* 17](#_Toc138145865)

[Gambar 2.8 *RGB* *convolution layer* 18](#_Toc138145866)

[Gambar 2.9 *Filter* 19](#_Toc138145867)

[Gambar 2.10 Proses konvolusi 19](#_Toc138145868)

[Gambar 2. 11 *Max pooling* dan *average pooling* 21](#_Toc138145869)

[Gambar 2.12 *Classification* 22](#_Toc138145870)

[Gambar 2.13 Arsitektur *MobileNet* 24](#_Toc138145871)

[Gambar 3.1 Kerangka berpikir…………………………………………………...29](#_Toc138102124)

[Gambar 3.2 Contoh gambar setelah proses rotasi 32](#_Toc138102125)

[Gambar 3.3 Contoh gambar setelah proses *flip* secara vertikal 32](#_Toc138102126)

[Gambar 3.4 Contoh gambar setelah proses *flip* secara horizontal 32](#_Toc138102127)

[Gambar 3.5 Contoh gambar setelah proses *shear* 33](#_Toc138102128)

[Gambar 4.1 *Source code* augmentasi data……………………………………….35](#_Toc138962585)

[Gambar 4.2 Jambu hasil *rotation* 35](#_Toc138962586)

[Gambar 4.3 Jambu hasil *horizontal flip* 36](#_Toc138962587)

[Gambar 4.4 Jambu hasil *vertical flip* 36](#_Toc138962588)

[Gambar 4.5 Jambu hasil *shear* 36](#_Toc138962589)

[Gambar 4.6 *Source code split dataset* buah jambu 37](#_Toc138962590)

[Gambar 4.7 Hasil *split dataset* daun jambu 37](#_Toc138962591)

[Gambar 4.8 *Source code base model* 38](#_Toc138962592)

[Gambar 4.9 *Source code sequential model* buah jambu 38](#_Toc138962593)

[Gambar 4.10 *Source code sequential model* daun jambu 39](#_Toc138962594)

[Gambar 4.11 *Source code compile model* 39](#_Toc138962595)

[Gambar 4.12 *Source code model fit* 39](#_Toc138962596)

[Gambar 4.13 *Graph loss* 20 *epoch* ………………………………………………40 Gambar 4.14 *Graph loss* 30 *epoch* 41](#_Toc138962597)

[Gambar 4.15 *Graph loss* 40 *epoch* 42](#_Toc138962598)

[Gambar 4.16 *Source code kernel regularizers* 43](#_Toc138962599)

[Gambar 4.17 *Graph loss kernel regularizers, dropout* 0.1dan 0.2 43](#_Toc138962600)

[Gambar 4.18 *Model evaluate kernel regularizers, 0.1 dropout* 43](#_Toc138962601)

[Gambar 4.19 *Model evaluate kernel regularizers, 0.2 dropout* 43](#_Toc138962602)

[Gambar 4.20 *Training graph model* buah jambu 44](#_Toc138962603)

[Gambar 4.21 Training *evaluation model* buah jambu 44](#_Toc138962604)

[Gambar 4.22 *Confusion matrix* *model* buah jambu 45](#_Toc138962605)

[Gambar 4.23 *Training graph model* daun jambu 45](#_Toc138962606)

[Gambar 4.24 *Training evaluation model* daun jambu 46](#_Toc138962607)

[Gambar 4.25 *Confusion matrix model* daun jambu 46](#_Toc138962608)

[Gambar 4.26 Halaman *splash* 47](#_Toc138962609)

[Gambar 4.27 Halaman *Onboarding* 1dan 2 47](file:///C:\Users\Fadlan\Documents\PROJECT-SKRIPSI\DOKUMEN\Implementasi%20Convolutional%20Neural%20Network%20(CNN)%20Untuk%20Klasifikasi%20Penyakit%20Pada%20Daun%20dan%20Buah%20Jambu%20Biji%20Berbasis%20Mobile.docx#_Toc138962610)

[Gambar 4.28 Halaman *Onboarding* 3 48](#_Toc138962611)

[Gambar 4.29 Halaman utama 48](file:///C:\Users\Fadlan\Documents\PROJECT-SKRIPSI\DOKUMEN\Implementasi%20Convolutional%20Neural%20Network%20(CNN)%20Untuk%20Klasifikasi%20Penyakit%20Pada%20Daun%20dan%20Buah%20Jambu%20Biji%20Berbasis%20Mobile.docx#_Toc138962612)

[Gambar 4.30 Halaman klasifikasi penyakit 49](file:///C:\Users\Fadlan\Documents\PROJECT-SKRIPSI\DOKUMEN\Implementasi%20Convolutional%20Neural%20Network%20(CNN)%20Untuk%20Klasifikasi%20Penyakit%20Pada%20Daun%20dan%20Buah%20Jambu%20Biji%20Berbasis%20Mobile.docx#_Toc138962613)

[Gambar 4.31 Halaman detail penyakit 50](file:///C:\Users\Fadlan\Documents\PROJECT-SKRIPSI\DOKUMEN\Implementasi%20Convolutional%20Neural%20Network%20(CNN)%20Untuk%20Klasifikasi%20Penyakit%20Pada%20Daun%20dan%20Buah%20Jambu%20Biji%20Berbasis%20Mobile.docx#_Toc138962614)

[Gambar 4.32 *QR Code* unduh aplikasi 50](#_Toc138962615)

# DAFTAR TABEL

[Tabel 2.1 Penelitian terdahulu 26](#_Toc138683996)

[Tabel 3.1 *Class dataset*…………………………………………………………..30](#_Toc138684047)

[Tabel 4.1 Perbandingan *dropout model* buah jambu…………………………….40](#_Toc138684062)

[Tabel 4.2 Perbandingan *dropout* model daun jambu 41](#_Toc138684063)

[Tabel 4.3 Perbandingan *epoch* model buah jambu 41](#_Toc138684064)

# DAFTAR LAMPIRAN

[Lampiran A Produksi Tanaman Buah-buahan Tahunan Jawa Barat Tahun 2020 dan 2021 58](#_Toc138684662)

[Lampiran B Curah Hujan di Stasiun Pengamatan Klimatologi Bogor Menurut Bulan (mm), 2020-2022 59](#_Toc138684663)

[Lampiran C Curah Hujan di Stasiun Pengamatan Meteorologi Citeko Menurut Bulan (mm), 2020-2022 59](#_Toc138684664)

[Lampiran D Jumlah Curah Hujan di Stasiun Pengamatan Meteorologi Jatiwangi Menurut Bulan (mm), 2020-2022 60](#_Toc138684665)

[Lampiran E Pengamatan Curah Hujan di Stasiun Pengamatan Geofisika Bandung Menurut Bulan (mm), 2019-2022 60](#_Toc138684666)

# 

**PENDAHULUAN**

## Latar Belakang

Pertanian adalah aktivitas yang dilakukan manusia untuk mengelola sumber daya hayati untuk menghasilkan bahan pangan, menurut Van Asrsten (1953) “pertanian adalah kegiatan manusia dalam memperoleh hasil yang asalnya dari tumbuh-tumbuhan, atau dari hewan yang pada awalnya dicapai dengan jalan yang disengaja dalam menyempurnakan kemungkinan, yang sudah diberikan oleh alam untuk mengembangbiakkan tumbuhan atau hewan tersebut.”

Salah satu subsektor dalam pertanian adalah *hortikultura*. Menurut Perrott, Jannick, dan Sygne (2023) *hortikultura* adalah cabang pertanian tanaman yang berhubungan dengan tanaman kebun, umumnya tanaman buah-buahan *(Frutikultura)*, tanaman sayuran *(Olerikultura)*, tanaman hias *(Florikultura)*, dan tanaman obat-obatan *(Bsiofarmaka)*. *Hortikultura* berasal dari bahasa Latin *hortus*, yaitu kebun dan *colere*, yaitu mengolah.

Salah satu jenis tanaman *Frutikultura* adalah jambu biji. “Jambu biji *(Psidium guajava L.)* telah ditanam dan dimanfaatkan sebagai buah yang penting di daerah tropis seperti India, Indonesia, Pakistan, Bangladesh, dan Amerika Selatan” (Kumar et al., 2021). Jambu biji dapat beradaptasi pada kisaran suhu antara 15°C dan 30°C. Di luar kisaran ini, efek suhu yang lebih rendah atau lebih tinggi mengurangi pembentukan buah, sementara suhu malam hari 5°C hingga 7°C menghentikan pertumbuhan. Selain itu, suhu rendah menghambat produksi, menyebabkan bunga rontok (Fischer & Melgarejo, 2021).

Jawa Barat merupakan salah satu Provinsi penghasil jambu biji dengan 692.488 kuintal pada tahun 2021, jumlah produksinya mengalami penurunan

sebesar 12.82% dari produksi tahun 2020, yaitu sebanyak 794.345 kuintal (Badan Pusat Statistik Provinsi Jawa Barat, 2022). Pada tahun 2020 hingga 2021 menurut

data dari 4 stasiun pengamatan cuaca di Jawa Barat, yaitu stasiun pengamatan Citeko (BPS Jawa Barat, 2022b), stasiun pengamatan Bogor (BPS Jawa Barat, 2022a), stasiun pengamatan Bandung (BPS Jawa Barat, 2022d), dan stasiun pengamatan Jatiwangi (BPS Jawa Barat, 2022c) tercatat curah hujan meningkat. Dengan tinggi nya curah hujan di tahun 2021 sehingga mengakibatkan tingginya penyakit yang menyerang tanaman, karena buah jambu optimal nya berada di suhu suhu sekitar 23 – 280C.

Perkembangan teknologi yang semakin maju dapat memudahkan dalam melakukan identifikasi penyakit yang terjangkit pada daun dan buah jambu, teknologi tersebut adalah *deep learning.* *Deep learning* adalah bagian dari machine learning yang berfokus pada area algoritma yang terinspirasi oleh pemahaman kita tentang bagaimana otak bekerja untuk mendapatkan pengetahuan (Amaratunga, 2021)*.* Pada penelitian ini teknologi *deep learning* akan melakukan pembelajaran *dataset* penyakit daun dan buah jambu yang sudah diberikan label atau disebut dengan metode *supervised learning.*

Mengenali citra daun dan buah jambu yang terkena penyakit diperlukan algoritma yang dapat melakukan klasifikasi gambar, algoritma untuk melakukan klasifikasi gambar adalah *Convolutional Neural Network* (CNN) merupakan “sebuah machine learning yang model pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada media 2 dimensi seperti gambar, video, teks atau suara.” (Mubarok, 2019, hal. 25). Dan akan menggunakan arsitektur *MobileNetV2.*

*Model deep learning* yang telah dibuat akan diimplementasikan menjadi sebuah aplikasi yang dapat diakses melalui *smartphone* .Berdasarkan (*Newzoo*, 2021) pada tahun 2021 pengguna *smartphone* di Indonesia mencapai 178 juta pengguna. Kemudian yang menggunakan *smartphone* dengan sistem operasi *android* pada bulan oktober 2022 menurut (*Statista*, 2022) mencapai 90%. Sehingga pada penelitian ini dibuatlah aplikasi yang berbasis sistem operasi *android.* Menurut Yosef Murya (2014) *android* adalah sistem operasi berbasis *linux* yang di gunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).

Solusi yang ditawarkan penulis adalah mengembangkan aplikasi *mobile* yang di jalankan pada sistem operasi *android*. Aplikasi tersebut dapat mengenali jenis penyakit yang menyerang daun dan buah jambu, serta akan memberikan informasi untuk menangani penyakit yang menyerang, dengan demikian akan memudahkan pengguna untuk mengambil tindakan pencegahan atau pengobatan yang tepat, sehingga diharapkan akan meningkatkan produktifitas hasil panen dan mengurangi kerugian yang diakibatkan oleh penyakit yang menyerang.

## Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah yang diambil adalah sebagai berikut:

1. Bagaimana menerapkan metode *supervised learning* menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur *MobileNetV2* untuk mengklasifikasikan jenis penyakit pada daun dan buah jambu?
2. Bagaimana menerapkan model yang telah dirancang ke aplikasi berbasis *mobile*?

## Batasan Masalah

Batasan masalah yang dibahas dan dapat diuraikan adalah sebagai berikut:

1. Menggunakan arsitektur *MobileNetV2.*
2. Hanya menggunakan *dataset* penyakit pada daun dan buah jambu.
3. Menggunakan metode *Cross Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari *Business Understanding, Data Undserstanding, Data Preparation, Modeling, Evaluation,* dan *Deployment*.
4. Evaluasi *model* menggunakan *training graph* dan *confusion matrix.*
5. Menerapkan model ke aplikasi berbasis *Mobile* dengan bahasa pemrograman *kotlin.*

## Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan metode *supervised learning* dengan menggunakan algoritma *Convolutional neural network* (CNN)dan arsitektur *MobileNetV2.*
2. Menerapkan model yang telah dirancang ke aplikasi berbasis *mobile.*

## Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

1. Memberikan pengetahuan mengenai implementasi metode *supervised learning* dengan menggunakan algoritma *Convolutional Neural Network* dan arsitektur *MobileNetV2*.
2. Mengetahui tingkat akurasi model yang dirancang menggunakan arsitektur *MobileNetV2.*
3. Memberikan pengetahuan dalam menerapkan model yang telah dirancang ke aplikasi berbasis *mobile.*
4. Membantu penanganan penyakit pada daun dan buah jambu sehingga diharapkan dapat meningkatkan produktifitas buah jambu yang sempat menurun.

## Sistematika Penulisan

Sistematika penulisan skripsi yang disusun terbagi dalam beberapa bab sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini menguraikan berbagai hal mengenai latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan. Uraian bab ini dimaksudkan untuk menjelaskan latar belakang penelitian sehingga dapat memberi masukan sesuai dengan tujuan penelitian dengan batasan-batasan masalah yang ada.

**BAB II TINJAUAN PUSTAKA**

Bab ini berisi tentang referensi pustaka yang mendukung dan terkait langsung dengan penelitian, yang diperoleh melalui buku, jurnal penelitian maupun sumber literatur lain. Pada bab ini juga dielaborasi kan hasil-hasil penelitian terdahulu yang berkaitan dengan masalah yang dikaji oleh penulis dalam penelitian.

**BAB III METODOLOGI PENELITIAN**

Bab ini berisi tentang langkah-langkah terstruktur dan sistematis yang dilakukan dalam penelitian. Langkah-langkah (tahapan tersebut disajikan dalam bentuk diagram alir yang disertai dengan penjelasan singkat.

**BAB IV PENGOLAHAN DATA DAN PEMBAHASAN**

Bab ini berisi tahapan-tahapan perancangan dengan menggunakan metode rasional, uraian lengkap mengenai pembuatan alat bantu pedal kaki melalui pengumpulan dan pengolahan data *anthropometri*, pengumpulan dan pengolahan data waktu siklus (sebelum dan setelah penggunaan alat bantu pedal kaki), serta analisis dan interpretasi hasil penelitian.

**BAB V KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran perbaikan yang perlu dilakukan pada objek penelitian.

# 

**TINJAUAN PUSTAKA**

## Implementasi

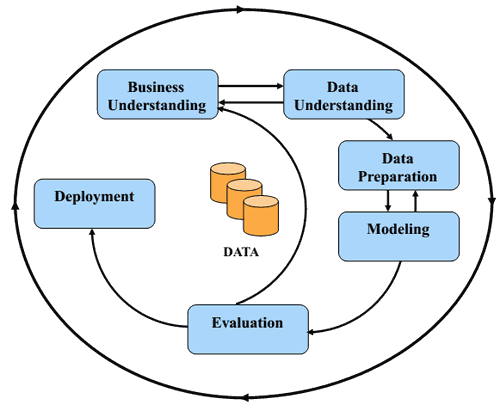
Menurut Kamus Besar Bahasa Indonesia (KBBI), implementasi adalah proses, cara, perbuatan menerapkan. Menurut Ben Lutkevich (2022) implementasi adalah pelaksanaan atau praktik rencana, metode atau desain, ide, model, spesifikasi, standar atau kebijakan apa pun untuk melakukan sesuatu. Dengan demikian, implementasi adalah tindakan yang harus mengikuti pemikiran awal agar sesuatu benar-benar terjadi.

Berdasarkan definisi-definisi tersebut dapat disimpulkan bahwa implementasi adalah suatu kegiatan yang terencana yang dilakukan untuk mencapai suatu gagasan, konsep, dan ide yang telah ditentukan.

## *Cross Industry Standard Process for Data Mining* (CRISP-DM)

*Cross Industry Standard Process for Data Mining* (CRISP-DM) merupakan standarisasi proses data mining yang dikembangkan pada tahun 1996 bertujuan untuk melakukan proses analisis strategi pemecahan masalah dari penelitian ataupun bisnis. (Fitriani, 2019, hal.11).

Menurut F. Krose (dalam Kurniawan & Yasir, 2022) Dalam siklus pengembangannya CRISP-DM dianggap sebagai metodologi *data mining* terlengkap dalam hal pemenuhan kebutuhan proyek industri, dan telah menjadi yang paling luas penggunaannya dalam proyek analitik, *data mining*, serta *data science*. Metode CRISP-DM memiliki enam tahapan, tahapan pertama *Business Understanding,* kemudian *Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment.*

**

Gambar 2.1 Diagram CRISP-DM (ML Wiki, 2014)

Berikut 6 tahapan CRISP-DM (Hasanah, Soim, dan Handayani, 2021) :

*Business Understanding*

Beberapa hal yang dilakukan pada tahap ini seperti memahami kebutuhan serta tujuan dari sudut pandang bisnis selanjutnya mengartikan pengetahuan ke dalam bentuk pendefinisian masalah pada *data mining* dan kemudian menentukan rencana serta strategi untuk mencapai tujuan *data mining.*

*Data Understanding*

*Data understanding* adalah tahap persiapan, melakukan pengecekan terhadap data yang digunakan, mengumpulkan data awal serta melakukan identifikasi pada kualitas data. Dalam *data understanding*, data yang digunakan akan melalui proses deskripsi dari setiap fiturnya.

*Data Preparation*

*Data preparation* merupakan proses yang dilakukan setelah data telah dikumpulkan. Pada tahap ini, data akan melalui proses identifikasi, pemilihan data, pembersihan data dan transformasi data.

*Modeling*

*Modeling* merupakan tahap implementasi algoritma yang akan digunakan untuk melakukan pencarian, identifikasi, serta menghasilkan pola yang akan digunakan pada data penelitian.

*Evaluation*

*Evaluation* adalah suatu proses untuk melakukan pengukuran hasil evaluasi dari *model* yang telah diimplementasikan sebelumnya di tahap *modeling*. Hasil evaluasi tersebut menggambarkan proses dari *data mining* yang telah dilakukan dan mengukur *model* yang paling baik untuk digunakan.

*Deployment*

Menggunakan *model* yang dihasilkan seperti pembuatan laporan atau penerapan proses *data mining.*

## Aplikasi

Aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju (Andi, 2015).

## Pertanian

Menurut *Van asresten* yang dikutip dari forbes.id (2019) pertanian adalah kegiatan manusia dalam memperoleh hasil yang asalnya dari tumbuh-tumbuhan, atau dari hewan yang pada awalnya dicapai dengan jalan yang disengaja dalam menyempurnakan kemungkinan, yang sudah diberikan oleh alam untuk mengembangbiakkan tumbuhan atau hewan tersebut.

## Hortikultura

Hortikultura, yaitu cabang pertanian tanaman yang berhubungan dengan tanaman kebun, umumnya buah-buahan *(Frutikultura)*, tanaman sayuran *(Olerikultura)*, tanaman hias *(Florikultura)*, dan tanaman obat-obatan *(Biofarmaka)*.Kata ini berasal dari bahasa Latin *hortus*, yaitu kebun dan *colere*, yaitu mengolah. (Perrott, Jannick, dan Sygne, 2023)

## Penyakit Tanaman

Penyakit tanaman yaitu gangguan pada tanaman yang mengakibatkan tanaman tidak dapat bereproduksi secara baik dan akhirnya mati secara perlahan, tanaman yang diserang memiliki perubahan dan gangguan pada organ – organ tanaman. Masalah penyakit tanaman ini selalu muncul setiap saat dari berbagai musim tanam, serta sering dijumpai – penyakit baru yang belum pernah ada. Penyebab tanaman terserang penyakit bisa dari berbagai hal yaitu disebabkan oleh mikroorganisme misalnya jamur, virus, dan bakteri serta bisa kekurangan salah satu unsur hara.

Tanaman yang berpenyakit memiliki ciri – ciri yaitu sulit dilihat oleh mata telanjang, disebabkan oleh mikroorganisme (virus, jamur, dan bakteri), penyakit tanaman tidak langsung terlihat tetapi tanaman mati secara perlahan – lahan. Selain tanda – tanda yang sulit terlihat ada juga tanda – tanda yang bisa dilihat secara langsung yaitu tanaman menjadi layu karena kekurangan air, rontok juga salah satunya bisa rontok pada daun, ranting, buah, dan bunga. Perubahan warna yang terjadi pada tumbuhan misal daun menjadi kuning, redup, dan pucat tetapi perubahan warna juga bisa terjadi karena kurangnya cahaya matahari yang mengakibatkan rusaknya klorofil, daun yang berlubang juga salah satu tanda awalnya bisa berupa bercak lingkaran, kerdil, daun mengeriting, busuk pada batang, daun, dan buah. (Agribisnis Universitas Medan Area, 2022)

## Jambu Biji

*Psidium guajava L.* atau sering biasa kita sebut jambu biji ini merupakan tanaman yang berasal dari Amerika Serikat Tengah, lalu penyebaran tanaman ini meluas ke kawasan Asia Tenggara dan ke wilayah Indonesia melalui Thailand (Cahyono, 2010). Di Indonesia, Pulau Jawa merupakan sentra penanaman buah jambu terbesar antara lain di DKI Jakarta, Jawa Barat, Jawa Tengah, Daerah DI Yogyakarta, dan Jawa Timur. Sentra produksi yang lain adalah Sumatera dan Kalimantan. (Prihatman, 2000)

Jawa Barat merupakan salah satu Provinsi penghasil jambu biji, berdasarkan katalog yang diterbitkan Badan Pusat Statistik Jawa Barat yang berjudul “Produksi Hortikultura Buah dan Sayur Provinsi Jawa Barat“ produksi jambu biji di tahun 2021 sebanyak 692.488 kuintal. (Badan Pusat Statistik Provinsi Jawa Barat, 2022)

### Syarat Tumbuh Jambu Biji

Sebuah tanaman memiliki syarat tumbuh agar tumbuhan tersebut dapat tumbuh dengan baik dan memiliki produktifitas yang bagus dalam memproduksi buahnya. Syarat tumbuh yang pertama, yaitu iklim. “Iklim adalah jalannya suatu keadaan cuaca atau keseluruhan dari gejala-gejala cuaca di daerah tertentu sepanjang tahun dan dari tahun ke tahun.” (Daljoeni, 1983). Perubahan iklim dapat mempengaruhi produktivitas suatu tanaman. Berikut ini iklim yang cocok untuk budidaya buah jambu biji:

* 1. Dalam budidaya tanaman jambu biji angin berperan dalam penyerbukan, namun angin yang kencang dapat menyebabkan kerontokan pada bunga.
  2. Tanaman jambu biji merupakan tanaman daerah tropis dan dapat tumbuh di daerah sub-tropis dengan intensitas curah hujan yang diperlukan berkisar antara 1000-2000 mm/tahun dan merata sepanjang tahun.
  3. Tanaman jambu biji dapat tumbuh berkembang serta berbuah dengan optimal pada suhu sekitar 23-28oC di siang hari. Kekurangan sinar matahari dapat menyebabkan penurunan hasil atau kurang sempurna (kerdil), yang ideal musim berbunga dan berbuah pada waktu musim kemarau yaitu sekitar bulan Juli - September sedang musim buahnya terjadi bulan November-Februari bersamaan musim penghujan.
  4. Kelembaban udara sekeliling cenderung rendah karena kebanyakan tumbuh di dataran rendah dan sedang. Apabila udara mempunyai kelembaban yang rendah, berarti udara kering karena miskin uap air. Kondisi demikian cocok untuk pertumbuhan tanaman jambu biji. (Prihatman, 2000)

Syarat tumbuh yang kedua, yaitu media tanam. “Media tanam diartikan sebagai media yang digunakan untuk menumbuhkan tanaman/bahan tanaman, tempat akar atau bakal akar tumbuh dan berkembang.” (Hariyono, 2021). Berikut ini media tanam yang cocok untuk tanaman jambu biji:

1. Tanaman jambu biji sebenarnya dapat tumbuh pada semua jenis tanah.
2. Jambu biji dapat tumbuh baik pada lahan yang subur dan gembur serta banyak mengandung unsur nitrogen, bahan organik atau pada tanah yang keadaan liat dan sedikit pasir.
3. Derajat keasaman tanah (pH) tidak terlalu jauh berbeda dengan tanaman lainnya, yaitu antara 4,5-8,2 dan bila kurang dari pH tersebut maka perlu dilakukan pengapuran terlebih dahulu. (Prihatman, 2000)

Syarat tumbuh yang ketiga, yaitu ketinggian ideal untuk menanam jambu biji. Ketinggian ideal ini dihitung dengan satuan meter diatas permukaan laut (mdpl), ketinggian ideal untuk menanam tanaman jambu biji dengan ketinggian antara 5-1200 mdpl yang dapat tumbuh subur pada daerah tropis.

### 2.7.2 Penyakit Jambu Biji

Berikut ini beberapa penyakit yang dapat menyerang daun jambu biji :

1. Karat merah / *Red rust (Cephaleuros sp)*



Gambar 2.2 Penyakit karat merah (Azeem, 2022)

Penyebab

Menurut Misra (dalam Bakara, dan Kurniawati, 2020) Penyakit ini disebabkan oleh alga *Cephaleuros virescens* pada daun, bunga, buah, ranting dan batang. Pada permukaan atas daun ditumbuhi talus yang tegak, dengan filamen berwarna kuning hingga merah. Daun di infeksi pada bagian tepi, pinggir atau seringkali pada area dekat tulang daun. Menurut (Rajbongshi, dkk., 2022) penyebabnya juga disebabkan oleh curah hujan yang tinggi, kelembapan yang berlebihan dan suhu hangat.

Pengendalian

Menurut Misra (dalam Bakara, dan Kurniawati, 2020) Pengendalian karat merah dapat dilakukan dengan penyemprotan tembaga *oksiklorida* (0.3%) sebanyak 3 sampai 4 kali dengan interval 15 hari hingga gejala membaik. Selain itu jarak tanam yang tidak rapat dapat mengurangi penyebaran *red rust /* karat merah.

*Dot* / Daun berlubang



Gambar 2. 3 Penyakit daun berlubang / *dot*

Penyebab

*Dot* / Daun berlubang disebabkan oleh ulat kantung (*Lepidoptera: Psychidae*), ulat kantung memakan daun muda terutama pada bagian bawah daun, sehingga mengakibatkan daun berlubang dan kering. Gejala kerusakan pada daun disebabkan aktivitas makan larva pada lapisan epidermis bawah dan jaringan *mesofil* dengan menyisakan epidermis atasnya mengakibatkan window panning. Sisa epidermis atas tersebut lalu mengering dan menyisakan tulang daun (Emmanuel et al. 2012 dalam Bakara, dan Kurniawati, 2020)

Pengendalian

Pengendalian penyakit daun berlubang dapat dikendalikan secara biologis, mekanik, dan kimiawi. Berikut adalah penjelasannya :

Biologis: pemanfaatan musuh alami predator *Sycanus sp. (Hemiptera: Reduviidae)* dan *Eocanthecona furcellata (Hemiptera: Pentatomidae)* serta *parasitoid* ulat *Brachimeria lasus (Hymenoptera: Chalcididae)*

Mekanik: pemangkasan daun yang terserang oleh ulat kantung, pemanfaatan perangkap cahaya untuk menangkap imago jantan dari ulat kantung.

Kimiawi: dilakukan jika populasi sudah melewati ambang batas. Aplikasi dapat dilakukan dengan penyemprotan secara langsung atau injeksi/infus akar. Bahan aktif yang direkomendasikan yaitu *abakmektin, abakmektin+sipermetrin, deltametrin, diazinon, kloramtiniliprol,* dan lain-lain. (Mubin, Khairini, Triwidodo, dan Bandi, 2021)

1. *Scab* / Kudis



Gambar 2.4 Penyakit *scab* (Azeem, 2022)

Penyebab

*Scab* disebabkan oleh jamur yang memiliki gejala bulat telur, gabus, dan kerusakan berbentuk bulat pada permukaan buah jambu biji. Kerusakan ini mempengaruhi daging di bawahnya, mengurangi kualitas buah dan nilai produksi Jamur terutama mempengaruhi buah-buahan mentah, yang menciptakan gelap dan kerusakan kudis coklat berdiameter 2-4mm. Kulit buah di bawah proses pematangan berubah menjadi lunak dan menghiasi coklat muda hingga coklat tua. Kelembaban relatif tinggi merusak buah-buahan di permukaan tanah yang ditutupi dengan dedaunan tebal. Buah-buahan yang jatuh telah sangat dirugikan. (Rajbongshi et al., 2022)

Pengendalian

Berikut ini beberapa pengendalian yang bisa dilakukan untuk menangani penyakit kudis pada buah jambu :

1. Pertahankan jarak tanam yang direkomendasikan (5-6m × 5-6m).
2. Buang cabang yang berlebih dan rusak setelah panen.
3. Tutupi buah-buahan dengan kertas atau kantong plastik dengan satu sisi terbuka di panggung kecil.
4. Kendalikan serangga kepik *Helopeltis Spp* untuk mengurangi inisiasi penyakit.
5. *Stylar end Rot* / busuk pangkal buah



Gambar 2.5 Penyakit *stylar end rot* (Azeem, 2022)

Penyebab

*Stylar end rot* atau disebut busuk pangkal buah, menurut Ellis (2021) buah jambu mengalami gejala kerusakan pada ujung buah yang kemudian membusuk yang disebabkan oleh patogen *alternaria citri*. Buah yang tidak sehat atau rusak rentan terhadap infeksi. Infeksi dapat terjadi saat buah masih berada di pohon. Dan menurut Rajbongshi, dkk., (2022) buah jambu yang terkena kelembapan yang lebih tinggi menjadi rusak yang paling parah.

Pengendalian

Penyakit dapat dicegah dengan tanah berdrainase baik, banyak sinar matahari, pupuk sesekali, air yang cukup, dan pengendalian hama.( Ellis, 2021)

1. *Phytophthora palmivora* / Penyakit busuk buah



Gambar 2.6 Penyakit *phytopthora palmivora* (Azeem, 2022)

Penyebab

Buah yang terinfeksi busuk *phytophthora* tampak berwarna coklat keabu-abuan dan basah kuyup dengan bagian tengah berwarna hitam keabu-abuan. Penyakit ini muncul dalam bentuk pertumbuhan miselium putih bertepung di atas buah. (Misra, 2006)

Jamur yang menginfeksi buah dapat bersumber dari tanah, batang yang sakit kanker batang, buah yang sakit, dan tumbuhan inang lain. *Phytophthora palmivora* akanbertahan dalam tanah, dari sini jamur dapat terbawa oleh percikan air hujan ke buah-buah yang dekat tanah. Setelah mengadakan infeksi, dalam waktu beberapa hari jamur pada buah sudah dapat menghasilkan banyak sporangium (bagian tubuh jamur yang berfungsi sebagai tempat pembentukan spora). *Sporangium* ini dapat terbawa oleh percikan air, atau oleh angin, dan mencapai buah yang lebih tinggi. (Kurniawati, 2021)

Pengendalian

Berikut ini beberapa pengendalian penyakit yang dapat dilakukan :

1. Memetik semua buah busuk yang dilakukan bersamaan dengan pemangkasan ataupun saat panen, kemudian dibenamkan ke dalam tanah sedalam 30 cm.
2. Pengaturan kelembaban kebun bisa dilakukan dengan memperbaiki drainase, memangkas tanaman dan pohon pelindung dengan teratur, dan dengan mengendalikan gulma. (Kurniawati, 2021)
3. Pemberian *dithane* efektif dalam mengendalikan penyakit (Misra, 2006)

## *Machine Learning*

Menurut Tom M. Mitchell (DqLab*, 2020*) *machine learning* adalah studi mengenai algoritma komputer yang dapat meningkatkan kinerja program komputer secara otomatis melalui data sebelumnya. Cara kerja machine learning adalah dengan mengumpulkan, memeriksa, dan membandingkan data berukuran kecil hingga besar untuk menemukan pola dan mengeksplorasi perbedaannya.

Mengutip dari (Nurhikmat, 2018) dalam pembelajaran *machine learning*, terdapat beberapa skenario-skenario. Seperti:

*Supervised Learning*

Penggunaan skenario *supervised learning*, pembelajaran menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label.

*Unsupervised Learning*

Penggunaan skenario *Unsupervised Learning*, pembelajaran menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokkan data berdasarkan karakteristik-karakteristik yang ditemui.

*Reinforcement Learning*

Pada skenario *reinforcement learning* fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar.

## *Deep Learning*

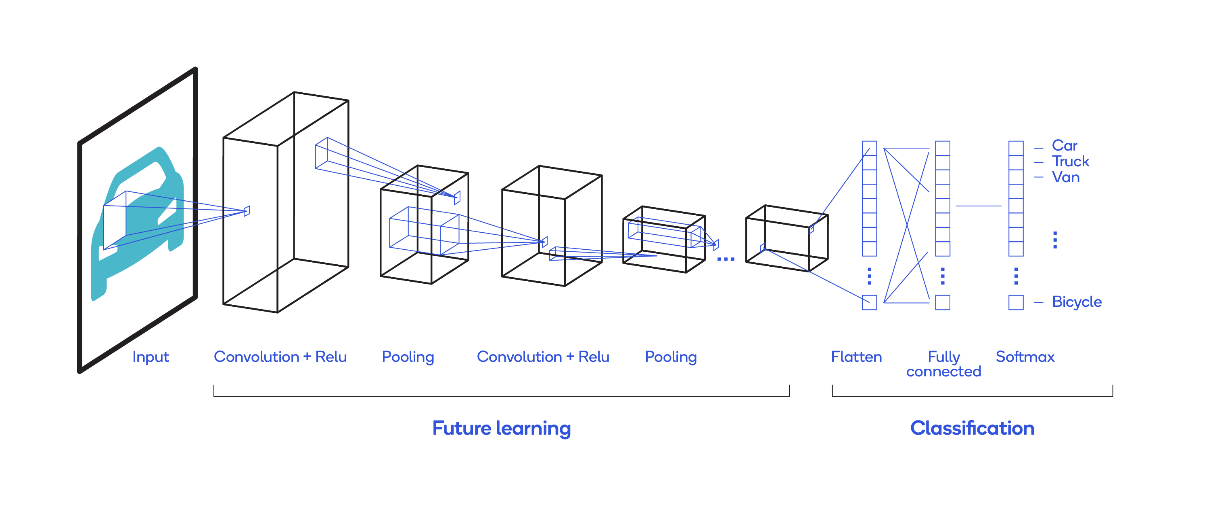
*Deep Learning* merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik.

Pada *Machine Learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi. Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditangguhkan pada algoritma *Machine Learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer vision, Speech recognition,* dan *Natural Language Processing.*

Dalam *Deep Learning,* metode CNN atau *Convolutional Neural Network* sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis *non-linier* yang dapat meningkatkan kekompleksitasan sebuah model. Model yang kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia *Deep Learning* pengunaan GPU sudah sangatlah umum. (Danukusumo, 2017)

## *Convolutional Neural Network* (CNN)

*Convolutional Neural Network* (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk ke dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data cita dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Penelitian awal yang mendasari penemuan CNN ini pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, 1968) mengenai visual *cortex* pada indera penglihatan kucing. Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan *(input)* dan keluaran *(output)* dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan *pooling* layer. (Nurhikmat, 2018)



Gambar 2.7 Arsitektur *convolutional neural network* (Mathworks, tanpa tahun)

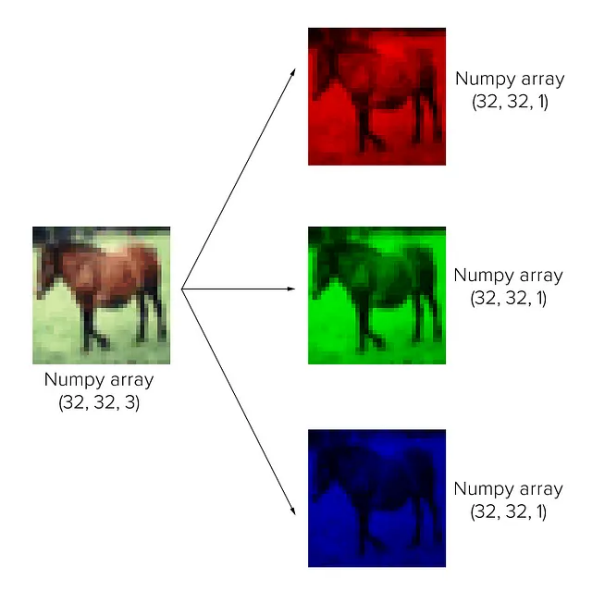
Arsitektur CNN dibagi menjadi 2 bagian besar, *yaitu Feature Learning* dan *Classification*

***Input* *Layer*,** lapisan ini berguna untuk menampung pixel value dari gambar yang diinputkan. Semisal gambar yang diinputkan adalah gambar dengan ukuran piksel 128×128 dengan 3 *channel* warna RGB *(Red, Green, Blue),* maka akan dimasukkan menjadi piksel *array* berukuran 128×128×3. (Syihabirawan, 2020)

### *Feature Learning*

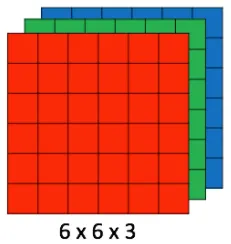
*Feature learning* adalah proses *encoding* dari sebuah gambar menjadi feature yang berupa nilai-nilai yang merepresentasikan gambar tersebut. Proses ini terdiri dari beberapa layer yang saling bekerjasama untuk mengambil ciri dari sebuah gambar.(Sena, 2017) *Feature learning* memiliki dua bagian, yaitu *Convolution Layer* dan *Pooling Layer.*

*Convolution Layer*



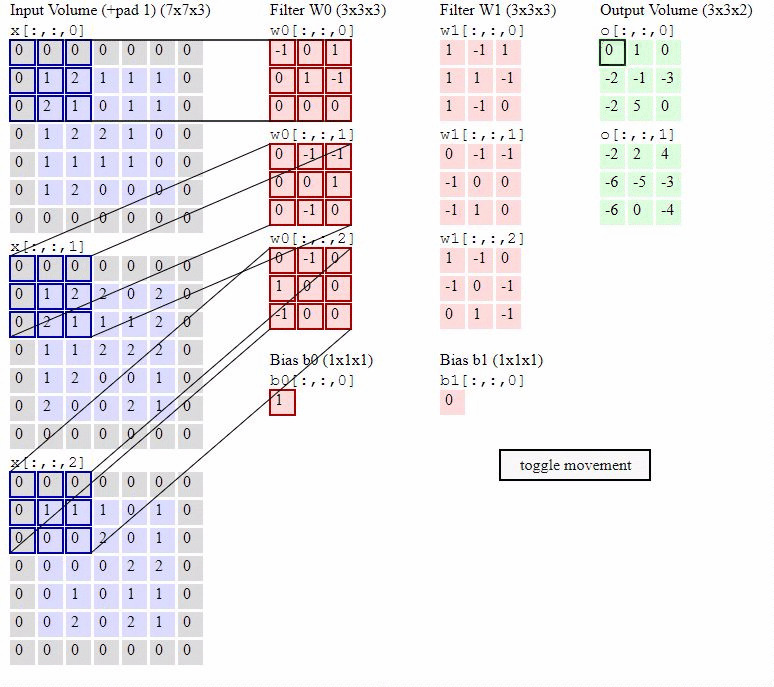
Gambar 2.8 *RGB* *convolution layer* (Sena, 2017)

Gambar diatas adalah RGB *(Red, Green, Blue)* gambar berukuran 32×32 *pixels* yang sebenarnya adalah *multidimensional array* dengan ukuran 32×32×3 (3 adalah jumlah *channel*). *Convolutional layer* terdiri dari *neuron* yang tersusun sedemikian rupa sehingga membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*). Sebagai contoh, *layer* pertama pada *feature extraction layer* dengan ukuran 5×5×3. Panjang 5 *pixels*, tinggi 5 *pixels* dan tebal/jumlah 3 buah sesuai dengan *channel* dari gambar tersebut.



Gambar 2.9 *Filter* (Syihabirawan, 2020)

Ketiga *filter* ini akan digeser ke seluruh bagian dari gambar. Setiap pergeseran akan dilakukan operasi “*dot*” antara *input* dan nilai dari *filter* tersebut sehingga menghasilkan sebuah *output* atau biasa disebut sebagai *activation map* atau *feature map*.(Sena, 2017)



Gambar 2.10 Proses konvolusi (Sena, 2017)

1. *Stride*

*Stride* adalah parameter yang menentukan berapa jumlah pergeseran *filter*. Jika nilai *stride* adalah 1, maka *convolution filter* akan bergeser sebanyak 1 *pixels* secara horizontal lalu vertikal. Pada ilustrasi diatas, *stride* yang digunakan adalah 2.

Semakin kecil *stride* maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar. Namun perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil kita tidak selalu akan mendapatkan performa yang bagus. (Sena, 2017)

1. *Padding*

Sedangkan *Padding* atau *Zero Padding* adalah parameter yang menentukan jumlah pixels (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolution layer (Feature Map)*.

Tujuan dari padding adalah :

1. Dimensi *output* dari *convolution layer* selalu lebih kecil dari *input*-nya (kecuali penggunaan 1×1 *filter* dengan *stride* 1). *Output* ini akan digunakan kembali sebagai *input* dari *convolution layer* selanjutnya, sehingga makin banyak informasi yang terbuang. Dengan menggunakan *padding*, kita dapat mengatur dimensi *output* agar tetap sama seperti dimensi input atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan *convolution layer* yang lebih dalam/*deep* sehingga lebih banyak *features* yang berhasil di-*extract*.
2. Meningkatkan performa dari model karena *convolution filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut.

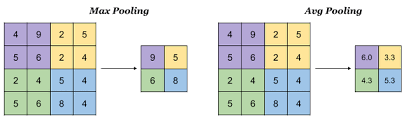
Pada ilustrasi gambar 2.8, dimensi dari *input* sebenarnya adalah 5×5, jika dilakukan *convolution* dengan *filter* 3×3 dan *stride* sebesar 2, maka akan didapatkan *feature map* dengan ukuran 2×2. Namun jika kita tambahkan *zero padding* sebanyak 1, maka *feature map* yang dihasilkan berukuran 3×3. ( Sena, 2017)

1. *Rectified Linear Unit (ReLU) activation*

*ReLU* *layer* berfungsi untuk menerapkan fungsi aktivasi terhadap nilai *output* hasil konvolusi pada *convolution layer.* Keluaran dari fungsi aktivasi dinyatakan sebagai 0 (nol) apabila *input*-nya negatif. Namun jika input-nya positif, maka *output*-nya akan sama dengan nilai *input* fungsi aktivasi itu sendiri (Ramba, 2020)

*Pooling Layer*

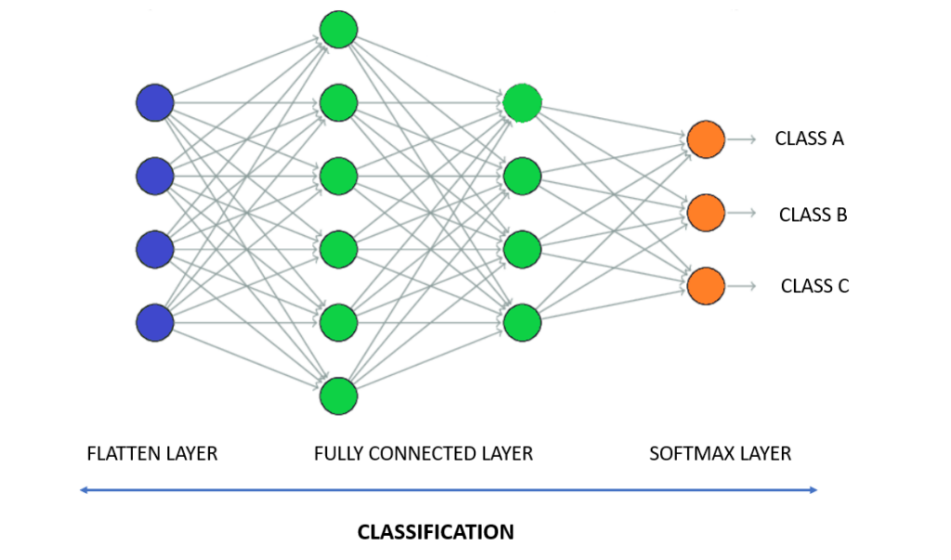
*Pooling layer* biasanya berada setelah *convolution layer*. Pada prinsipnya pooling layer terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. *Pooling* yang biasa digunakan adalah *Max Pooling* dan *Average Poolin*g. Sebagai contoh jika kita menggunakan *Max Pooling* 2×2 dengan *stride* 2, maka pada setiap pergeseran *filter*, nilai maksimum pada area 2×2 *pixel* tersebut yang akan dipilih, sedangkan *Average Pooling* akan memilih nilai rata-ratanya.(Sena, 2017)



Gambar 2. 11 *Max pooling* dan *average pooling* (Ramba, 2020)

Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus diupdate semakin sedikit dan mengatasi *overfitting*.(Sena, 2017)

### *Classification*



Gambar 2.12 *Classification* (Ramba, 2020)

Proses *classification* berfungsi untuk melakukan klasifikasi terhadap setiap *neuron* yang telah di ekstraksi pada proses *feature learning.* Bagian ini terdiri dari beberapa layer yang saling berkaitan satu sama lain. (Ramba, 2020) Berikut ini adalah penjelasan setiap fungsi pada bagian *classification*.

*Flatten*

*Feature map* yang dihasilkan pada proses *feature learning* adalah dalam bentuk *multidimensional array*. Sedangkan *input* untuk *fully connected layer* haruslah data dalam bentuk vektor. Oleh karena itu, dibutuhkan sebuah fungsi yang dapat mengubah data *multidimensional array* ke dalam bentuk vektor.

*Flatten* berfungsi untuk membentuk ulang (*reshape*) *feature map* dari *multidimensional array* menjadi vektor. Hal ini diperlukan agar nilai-nilai tersebut dapat digunakan sebagai *input* pada *fully connected layer.* (Ramba, 2020)

*Fully Connected Layer*

Lapisan *Fully-connected* adalah lapisan dimana semua neuron aktivitas dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti hal nya jaringan syaraf tiruan bisa. Setiap aktivitas dari lapisan sebelumnya perlu diubah menjadi data satu dimensi sebelum dapat dihubungkan ke semua *neuron* di lapisan *Fully-Connected*.

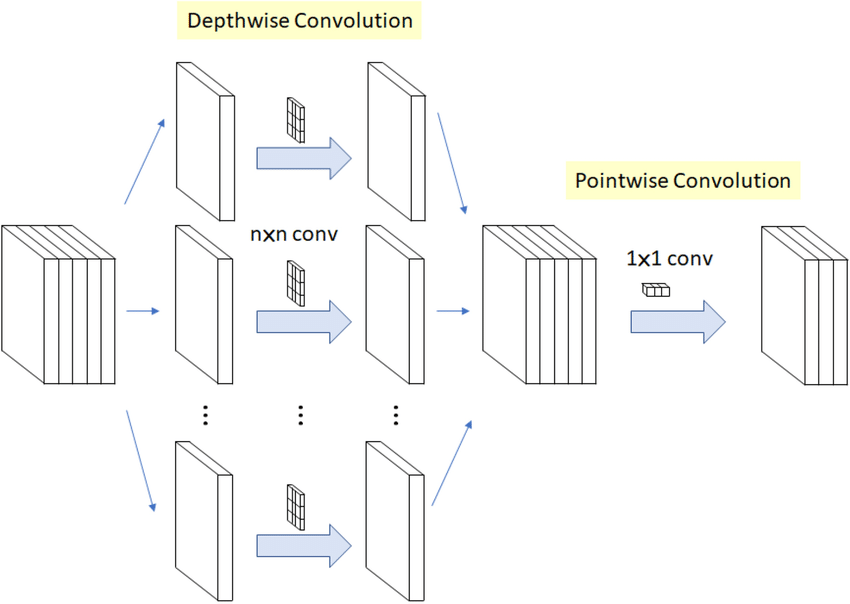
Lapisan *Fully-Connected* biasanya digunakan pada metode Multi lapisan *Perceptron* dan bertujuan untuk mengolah data sehingga bisa diklasifikasikan. Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah *neuron* di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*. Sementara lapisan *Fully-Connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk *dot*, sehingga fungsinya tidak begitu berbeda.(Nurhikmat, 2018)

*Softmax*

Peran utama dari *softmax* dalam proses klasifikasi adalah untuk menghitung probabilitas dari setiap kelas target terhadap semua kelas target yang ada. Rentang probabilitas output pada *softmax* adalah nilai dari 0 hingga 1, dan apabila semua nilai probabilitas dari kelas target tersebut dijumlahkan, maka nilainya akan sama dengan satu. *Softmax* menggunakan eksponensial dari nilai *input* yang diberikan dan jumlah nilai eksponensial dari semua nilai dalam *output*. Rasio *eksponensial* dari nilai *input* dan jumlah nilai *eksponensial* adalah *output* dari fungsi *softmax*.(Ramba, 2020)

## *MobileNet*

*MobileNet*, merupakan salah satu arsitektur *convolutional neural network* (CNN) yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih. Seperti namanya, *Mobile*, para peneliti dari *Google* membuat arsitektur CNN yang dapat digunakan untuk ponsel. Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur CNN pada umumnya adalah penggunaan lapisan atau layer konvolusi dengan ketebalan *filter* yang sesuai dengan ketebalan dari input image. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.(Ekoputris, 2018)



Gambar 2.13 Arsitektur *MobileNet* (Tsang, 2018)

Menurut Howard, dkk., 2017 (dalam Jauhari, 2022) Model pada arsitektur *MobileNet* yang didasarkan pada konvolusi yang mendalam dapat dipisah-pisah (*depthwise separable convolution*) menjadi bentuk konvolusi yang menguraikan konvolusi standar (*standard convolution*) menjadi konvolusi mendalam (*depthwise convolution*) dan konvolusi 1×1 atau konvolusi 20 searah (*pointwise convolution*). *Depthwise convolution* memiliki penerapan satu filter pada setiap saluran masukan, kemudian *pointwise convolution* menggunakan konvolusi 1×1 untuk menggabungkan hasil keluaran dari *depthwise convolution.* Sedangkan *standard* *convolution* mem-filter dan melakukan penggabungan masukan pada sebuah set keluaran yang baru. *Deptwise separable convolution* membagi lapisan menjadi dua bagian berdasarkan fungsinya, yaitu lapisan untuk *filter* dan lapisan untuk penggabung. Pembagian lapisan ini dapat mengurangi komputasi dan ukuran model.

## *Tensorflow*

*Tensorflow* adalah kerangka kerja *opensource* yang dikembangkan oleh peneliti Google untuk menjalankan *machine learning*, *deep learning*, dan beban kerja analitik statistik dan prediktif lainnya. Seperti platform serupa, ini dirancang untuk merampingkan proses pengembangan dan pelaksanaan aplikasi analitik canggih untuk pengguna seperti *data scientists*, ahli statistik, dan pemodel prediktif. (Vaughan, 2018)

## *Tensorflow Lite*

Dikutip dari situs resmi *Tensorflow* (Tensorflow, 2022) *TensorFlow Lite* adalah seperangkat alat yang memungkinkan pembelajaran mesin di perangkat dengan membantu pengembang menjalankan model di perangkat seluler, dan IOT. Fitur *Tensorflow Lite:*

1. Dioptimalkan untuk pembelajaran mesin di perangkat, dengan mengatasi 5 kendala utama: latensi (tidak ada perjalanan pulang pergi ke server), privasi (tidak ada data pribadi yang keluar dari perangkat), konektivitas (konektivitas internet tidak diperlukan), ukuran (model yang diperkecil dan ukuran biner), dan konsumsi daya (kesimpulan yang efisien dan kurangnya koneksi jaringan).
2. Dukungan berbagai platform, mencakup perangkat Android dan iOS, *embedded Linux*, dan *microcontroller*.
3. Dukungan bahasa yang beragam, yang mencakup *Java, Swift, Objective-C, C++,* dan *Python.*
4. Performa tinggi, dengan akselerasi perangkat keras dan optimalisasi model.
5. Contoh, untuk tugas-tugas *machine learning* yang umum seperti klasifikasi gambar, deteksi objek, estimasi pose, menjawab pertanyaan, klasifikasi teks, dan lain-lain di berbagai platform.

## *Android*

Menurut Yosef Murya (Murya, 2014) android adalah “sistem operasi berbasis linux yang di gunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).”

## *Android Studio*

*Android Studio* adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi *Android*. *Android Studio* menawarkan lebih banyak fitur yang meningkatkan produktivitas saat membuat aplikasi *Android*. (*Android Developers*, tanpa tahun)

## *Integrated Development Environment* (IDE)

*Integrated Development Environment* (IDE) adalah aplikasi perangkat lunak yang membantu para pemrogram mengembangkan kode perangkat lunak secara efisien. Aplikasi ini meningkatkan produktivitas developer dengan menggabungkan kemampuan seperti pengeditan, pembangunan, pengujian, dan pengemasan perangkat lunak dalam aplikasi yang mudah digunakan. Layaknya para penulis yang menggunakan editor teks dan para akuntan yang menggunakan spreadsheet, developer perangkat lunak menggunakan IDE untuk memudahkan pekerjaan mereka. (AWS Amazon, tanpa tahun)

## *Kotlin*

*Kotlin* adalah bahasa pemrograman *general-purpose* yang dikembangkan oleh *JetBrains*, yang telah membangun *Integrated Development Environment* (IDE) kelas dunia seperti *IntelliJ IDEA, PhpStorm, Appcode*, dan lain-lain. Ini pertama kali diperkenalkan oleh *JetBrains* pada tahun 2011 dan merupakan bahasa baru untuk JVM. *Kotlin* adalah bahasa berorientasi objek, dan "bahasa yang lebih baik" daripada *Java*, tetapi masih sepenuhnya dapat dioperasikan dengan kode *Java*. *Kotlin* disponsori oleh *Google*, diumumkan sebagai salah satu bahasa resmi untuk Pengembangan *Android* pada tahun 2017. (*GeeksforGeeks*, 2023)

## Penelitian Terdahulu

Berikut ini adalah acuan penelitian terdahulu.

Tabel 2.1 Penelitian terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| **Penulis** | **Judul** | **Metode** | **Hasil Penelitian** |
| Ari Kurniawan Sudiarto, Khoirida Aelani, Fresa Dwi Juniar (2020) | Identifikasi Penyakit pada Daun Jambu Kristal Berbasis *Android* dengan Metode *Enterprise*  *Unified Process* | Metode penelitian yang digunakan adalah metode pengembangan perangkat lunak *(Enterprise Unified Process)* yang tahapannya meliputi *Contruction, Transition, dan Production.* | Sample daun yang berpenyakit dan sehat akan di proses menggunakan aplikasi MATLAB untuk mengetahui nilai matriks RGB *(Red, Green, Blue)* dari sampel daun yang di proses. Kemudian nilai *matriks* tersebut menjadi acuan untuk aplikasi *android* yang akan menggunakan bahasa pemrograman *JAVA*. Aplikasi *android* akan merubah foto yang diambil menjadi nilai *matriks* kemudian nilai *matriks* tersebut akan disandingkan dengan nilai *matriks* daun berpenyakit dan daun sehat dan akan memberikan output berupa jenis penyakit dan cara penanganannya. |
| Umar Ghoni, Emmie Fatkhunnajah, Helmi Saputra, Anisatun Cahyani (2021) | Deteksi Penyakit Daun Pada Citra Daun Jambu Biji Menggunakan Segmentasi Warna | Metode yang digunakan pada penelitian ini adalah algoritma *kmeans*. *Kmeans* digunakan untuk mengelompokkan citra daun jambu biji menjadi dua daerah yaitu daerah yang sehat, dan daerah yang sakit. | Sample daun berpenyakit dilakukan proses *preprocessing* untuk mengambil nilai a dan nilai b menggunakan *cielab*, kemudian hasil dari nilai tersebut dimasukan ke algoritma *k-means* untuk mengelompokkan citra daun yang sehat dan berpenyakit. |
| Putri Teresia Ompusunggu (2022) | Klasifikasi Penyakit Tanaman Pada Daun Kentang Dengan Metode Convolutional Neural Network Arsitektur *Mobilenet* | Metode yang digunakan adalah Convolutional Neural Network dengan arsitektuk *MobileNet* | Dataset dibagi menjadi train, validasi, dan test. Kemudian akan dilakukan train model menggunakan algoritma Convolutional neural network dan arsitektur *MobileNet*. Dan akan diukur performanya dengan confusion matrix dengan melihat nilai akurasi, *loss, recall*, dan *f1-score.* |
| Mirza Faturrachman, Indra Yustiana, Somantri (2022) | Sistem Pendeteksi Penyakit Pada Daun Tanaman Singkong Menggunakan *Deep Learning* Dan *Tensorflow* Berbasis *Android* | metode *Convolutional Neural Network* (CNN) dan dengan bantuan *Framework Tensorflow* yang berbasis *Android*. Metode pengembangan sistem yang digunakan adalah *Prototyping.* | *Dataset* akan dilakukan proses augmentasi terlebih dahulu dan akan dibagi sebanyak 90% untuk data *train* dan 10% untuk *testing*. Kemudian akan dilakukan proses *train* model menggunakan metode *k-fold cross validation*. Selanjutnya model akan di konversi menadi file *tensorflow lite*. file *tensorflow lite* akan di *deploy* menjadi aplikasi *android* dengan bahasa pemrograman kotlin, proses klasifikasinya dengan mengambil gambar dari galeri ataupun kamera dan akan memberikan *output* jenis penyakitnya. |

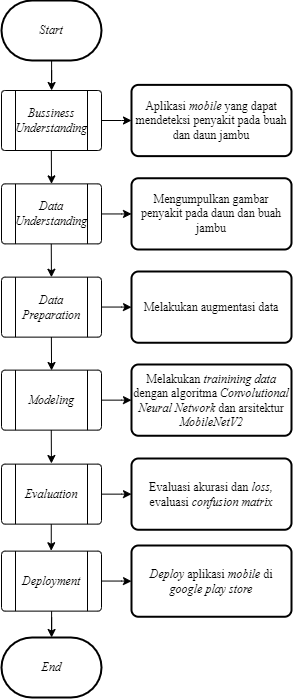
# 

# BAB III

**METODOLOGI PENELITIAN**

## Kerangka Berpikir

Berikut gambar 3.1 kerangka berpikir dari penelitian ini.



Gambar 3.1 Kerangka berpikir

## 

Perancangan *model deep learning* pada penelitian ini menggunakan metode *Cross Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari enam tahapan, tahapan pertama *Business Understanding,* kemudian *Data Understanding, Data Preparation, Modeling, Evaluation,* dan *Deployment.*

## *Business Understanding*

Penelitian ini bertujuan untuk membuat sebuah aplikasi *mobile* yang dapat melakukan klasifikasi penyakit pada daun dan buah jambu dengan metode *supervised learning* yang menggunakan algoritma *Convolutional Neural Network* dan arsitektur *MobileNetV2.* Aplikasi akan melakukan pengenalan penyakit pada daun dan buah jambu dengan cara mengambil gambar dengan menggunakan kamera *handphone* kemudian akan menghasilkan *output* jenis penyakit dari gambar yang telah diambil.

## *Data Understanding*

Data yang digunakan berupa gambar penyakit pada daun dan buah jambu yang didapatkan dari *website Kaggle.* Gambar di dataset ini memiliki ukuran piksel yang beragam, *dataset* ini memiliki 6 buah *class*/label. Berikut ini *class* tersebut:

Tabel 3.1 *Class dataset*

|  |  |  |
| --- | --- | --- |
| ***Class*** | **Definisi** | **Sumber** |
| Jambu (sehat) | Gambar buah jambu yang sehat | https://www.kaggle.com/datasets/aelchimminut/fruits262  (Minut, 2021) |
| Daun jambu (sehat) | Gambar daun jambu yang sehat | https://www.kaggle.com/datasets/mhantor/guava-disease  (Azeem, 2022) |
| *Phytopthora* | Gambar penyakit *Phytopthora* / busuk pada buah jambu |
| *Red rust* | Gambar penyakit *red rust* / karat merah pada daun jambu |
| *Scab* | Gambar penyakit *scab* /kudis pada buah jambu |
| *Stylar end rot* | Gambar penyakit *stylar end rot* /busuk pangkal buah pada buah jambu |
| *Dot* | Gambar penyakit *dot* / daun jambu berlubang | <https://www.kaggle.com/datasets/omkarmanohardalvi/guava-disease-dataset-4-types>  (Dalvi, 2022) |

## *Data Preparation*

Sebelum dilakukan *training* model *dataset* perlu untuk dilakukan *data augmentation*, menurut D. Seita (dalam Sanjaya & Ayub, 2020) Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Teknik augmentasi data seperti *cropping, padding, dan flipping horizontal.* Berikut ini *data augmentation* yang diterapkan pada penelitian ini:

*Rescale*

Dilakukan untuk merubah skala dari citra *dataset*, agar semua gambar memiliki skala yang sama dan akan meringankan komputasi ketika proses *train model* dilakukan. Pada penelitian ini dilakukan *rescale* sebesar 224 piksel.

*Split dataset*

Dilakukan untuk membagi dataset untuk keperluan *train* dan validasi. Pada penelitian ini dataset dibagi 80% untuk kebutuhan *train* dan 20% untuk kebutuhan validasi.

*Rotation*

Berfungsi untuk melakukan rotasi pada sebuah citra. Penelitian ini dilakukan rotasi sebesar 20 derajat.

Gambar 3.2 Contoh gambar setelah proses rotasi (Setiawan, 2021)

*Horizontal & vertical flip*

*Horizontal* dan *vertical flip* berfungsi untuk membalik sebuah citra menjadi *horizontal* ataupun *vertical.*

**

Gambar 3.3 Contoh gambar setelah proses *flip* secara vertikal (Setiawan, 2021)

**Gambar 3.4 Contoh gambar setelah proses *flip* secara horizontal (Setiawan, 2021)

*Zoom*

Berfungsi untuk melakukan perbesar pada gambar. Di penelitian ini akan memberikan nilai *zoom* sebesar 20%.

*Shear*

*Shear* berfungsi untuk melakukan pergeseran gambar sehingga dapat menghasilkan gambar yang dilihat dari sudut yang berbeda. Di penelitian ini diberikan *shear* sebesar 20 derajat.

**

Gambar 3.5 Contoh gambar setelah proses *shear* (Setiawan, 2021)

## *Modeling*

Model yang digunakan pada penelitian ini adalah hasil *training* dengan menggunakan algoritma *Convolutional Neural Network* dengan arsitektur *MobileNetV2* yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih (Ekoputris, 2018).

Gambar *dataset* yang di *input* dengan ukuran 224×224 piksel dengan 3 channel warna RGB (Red, Green, Blue) dan akan melewati proses pada arsitektur *MobileNetV2* kemudian *model* akan melakukan proses *training* secara berulang hingga mendapatkan akurasi yang tinggi.

## *Evaluation*

Pada tahap ini akan dilakukan evaluasi dari model yang telah di *training,* evaluasi pertama adalah melihat nilai dari grafik tingkat akurasi yang terdiri dari *training accuracy* dan *validation accuracy,* kemudian grafik tingkat *error* yang terdiri dari *training loss* dan *validation loss*. Jika nilainya rendah dan terjadi *overfitting* atau *underfitting* maka akan dilakukan perancangan dan *training* ulang *model*.

Tahap evaluasi yang kedua adalah dengan membuat *confusion matrix* dari *model* yang telah dibuat. Menurut D. Putra (dalam Normawati & Prayogi, 2021) *Confusion matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah.

## *Deployment*

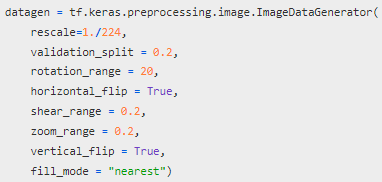
Melakukan *deployment* aplikasi *mobile* sehingga aplikasi dapat di unduh dan dapat digunakan oleh siapapun yang menggunakan *smartphone* dengan sistem operasi *Android.*

# BAB IV

**PENGOLAHAN DATA DAN PEMBAHASAN**

## *Data Preparation*

*Dataset* yang telah didapatkan dari situs *Kaggle* akan dilakukan augmentasi data menggunakan *ImageDataGenerator* sebelum melakukan proses *modeling*, berikut augmentasi data yang digunakan pada penelitian ini:



Gambar 4.1 *Source code* augmentasi data

*Split dataset*

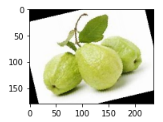
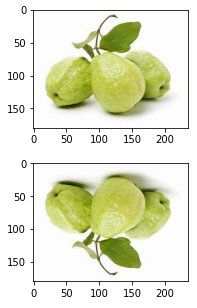
Pada penelitian ini sebesar 80% dari *dataset* untuk kebutuhan *training* dan sebesar 20% dari *dataset* untuk kebutuhan validasi.

*Rescale*

Pada penelitian ini gambar akan dilakukan *rescale* sehingga gambar memiliki ukuran 224 piksel.

*Rotation*

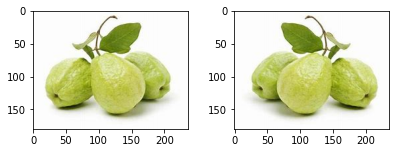
Pada penelitian ini dilakukan rotasi sebesar 20 derajat.



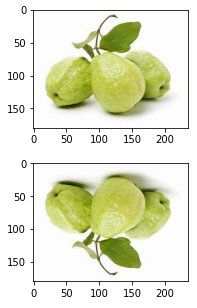
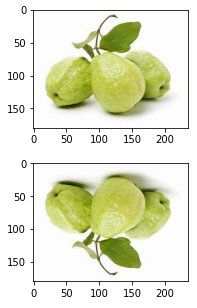
Gambar 4.2 Jambu hasil *rotation*

*Horizontal* dan *vertical flip*

Berikut ini hasil gambar yang telah dilakukan *vertical flip* dan *horizontal flip*



Gambar 4.3 Jambu hasil *horizontal flip*



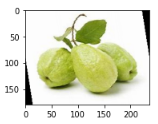
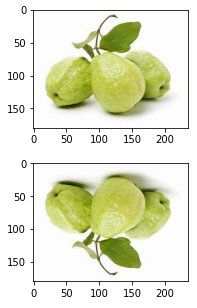
Gambar 4.4 Jambu hasil *vertical flip*

*Zoom*

Pada penelitian ini dilakukan *zoom* sebesar 20%.

*Shear*

Pada penelitian ini dilakukan *shear* sebesar 20 derajat yang berlawanan dengan arah jarum jam.



Gambar 4.5 Jambu hasil *shear*

*Fill\_mode*

Pada penelitian ini menggunakan *fill mode nearest*, di mana nilai piksel yang paling dekat dipilih dan diulangi untuk semua nilai yang kosong. (Sarin, 2019)



Gambar 4.6 *Source code split dataset* buah jambu



Gambar 4.7 Hasil *split dataset* daun jambu

Setelah membuat augmentasi data yang disimpan pada variabel *datagen*, selanjutnya mengambil *dataset* dan memisahkannya antara *train* dan *validation* seperti yang terlihat pada gambar 4.6. Kemudian gambar 4.7 menyatakan hasil split data dari *dataset* daun jambu.

## *Modelling*

*Modelling* terbagi menjadi dua bagian, yaitu proses *modelling* yang merupakan tahapan dalam pembuatan *model*, dan penentuan parameter *modelling* yang merupakan tahap menentukan parameter *model* dan akan membandingkan parameter mana yang lebih baik performanya. Berikut ini dua bagian yang diambil dalam *Modelling* :

### Proses *modelling*

Berikut adalah tahapan yang diambil dalam *modelling:*

1. Membuat *base model*



Gambar 4.8 *Source code base model*

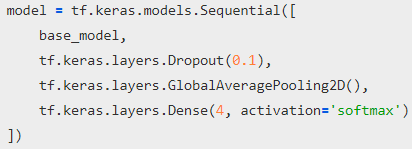
Yang pertama dilakukan dalam *modelling* adalah membuat *variable* untuk menyimpan *base model* arsitektur *MobilNetV2,* berikut ini uraian penjelasan parameter yang diisi:

Nilai *input\_shape* = 224,224,3 yang berarti ukuran piksel 224×224 dengan 3 channel warna RGB (*Red, Green, Blue*).

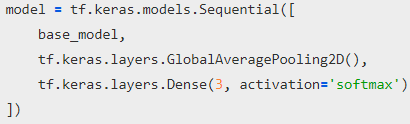
*Include\_top = false,* berfungsi untuk tidak perlu membawa *prediction node* yang merupakan bagian dari *MobileNetV2* yang memiliki nilai 1000 kelas (Afif, 2020), sementara di penelitian ini kelasnya tidak mencapai 1000.

*Weights = ‘imagenet’, Weights* berfungsi untuk bobot dari masing-masing layer yang sudah di training berdasarkan bobot yang ada pada *library* *imagenet*, sehingga proses *training* akan lebih ringan. (Afif, 2020).

1. Membuat *sequential model*



Gambar 4.9 *Source code sequential model* buah jambu



Gambar 4.10 *Source code sequential model* daun jambu

Selanjutnya membuat *sequential* *model* yang akan dilakukan *training,* berikut ini penjelasan yang ada pada *sequential model*:

Menambahkan *base model MobileNetv2.*

*Dropout* sebesar 0.1, *dropout* merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* menghilangkan *neuron* secara acak. (Lina, 2019)

*GlobalAveragePooling2D,* yang berfungsi untuk mengambil nilai rata-rata dari dimensi data.

*Dense* berfungsi untuk menambahkan *fully-connected layer,* yang memiliki kelas 4 buah sesuai kelas dari penyakit pada buah jambu biji, dan 3 kelas sesuai dengan jumlah kelas dari penyakit pada dun jambu biji. Selanjutnya aktivasi *softmax* yang berfungsi untuk mengambil hasil *output* dengan probabilitas paling tinggi yang akan dijadikan sebagai prediksi.

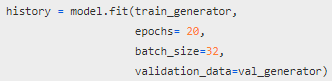
1. *Compile Model*



Gambar 4.11 *Source code compile model*

Melakukan *compile model* menggunakan *adam optimizer,* untuk *loss function* menggunakan *categorical\_crossentropy* karena memiliki lebih dari dua kelas, dan *metrics* evaluasi yang dipilih adalah *accuracy.*

1. *Model fit*



Gambar 4.12 *Source code model fit*

Selanjutnya melakukan proses *training* dengan menggunakan *Model fit.* Disini menambahkan dataset *train* dan *validation* yang telah disimpan pada variabel *train\_generator* dan *val\_generator* yang telah dibuat pada tahap *split dataset,* kemudian menentukan jumlah iterasi *training* atau *epochs,* dan berapa banyak *dataset* yang akan diambil setiap sekali *epochs* atau disebut *batch size.*

### Penentuan parameter *modelling*

Pada saat melakukan *training model* harus menentukan parameter dalam *model* yang akan menentukan seberapa baiknya *model* yang telah selesai *training.* Parameter yang dimaksud adalah penentuan nilai *dropout,* jumlah *epoch,* penentuan nilai *kernel regularization.* Penentuan parameter berfungsi untuk membandingkan parameter mana yang memiliki performa terbaik yang akan digunakan pada *model.*

Pengaruh nilai *dropout*

*Dropout* merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* menghilangkan *neuron* secara acak. (Lina, 2019). Berikut ini perbandingan nilai *dropout* pada *model* buah jambu dan *model* daun jambu.

*Model* buah jambu

Berikut ini adalah hasil perbandingan *dropout* pada *model* penyakit buah jambu dengan menggunakan 20 *epoch.*

Tabel 4.1 Perbandingan *dropout model* buah jambu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Dropout* | *Training Accuracy* | *Validation Accuracy* | *Training Loss* | *Validation Loss* |
| 0 | 0.977 | 0.880 | 0.115 | 0.313 |
| 0.1 | 0.971 | 0.892 | 0.121 | 0.313 |
| 0.2 | 0.974 | 0.904 | 0.114 | 0.246 |

Model daun jambu

Berikut ini adalah hasil perbandingan *dropout* pada model penyakit daun jambu dengan menggunakan 20 *epoch*.

Tabel 4.2 Perbandingan *dropout model* daun jambu

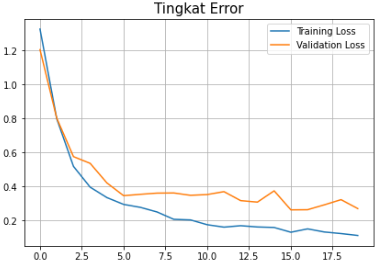
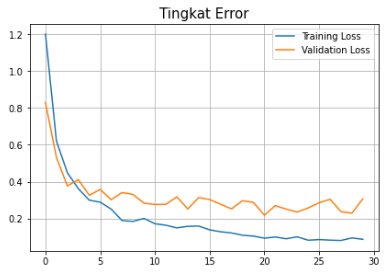
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Dropout* | *Training Accuracy* | *Validation Accuracy* | *Training Loss* | *Validation Loss* |
| 0 | 0.996 | 1.0 | 0.032 | 0.031 |
| 0.1 | 0.996 | 1.0 | 0.037 | 0.029 |
| 0.2 | 0.996 | 1.0 | 0.027 | 0.027 |

Pengaruh jumlah *epoch*

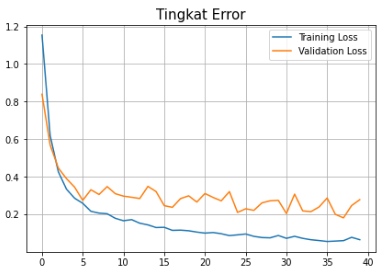
*Epoch* adalah ketika seluruh dataset sudah melalui proses training pada *Neural Network* sampai dikembalikan ke awal dalam satu putaran. (Nurhikmat, 2018). Pada penelitian ini model daun jambu sudah memiliki nilai *training loss* dan *validation loss* yang sangat baik meskipun hanya menggunakan 20 *epoch,* sehingga pada perbandingan jumlah *epoch* hanya akan membandingkan jumlah *epoch* dari model buah jambu, berikut ini tabel 4.3 hasil perbandingan *epoch* model buah jambu dengan menggunakan *dropout* sebesar 0.2.

Tabel 4.3 Perbandingan *epoch* model buah jambu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Epoch* | *Training Accuracy* | *Validation Accuracy* | *Training Loss* | *Validation Loss* |
| 20 | 0.974 | 0.904 | 0.114 | 0.246 |
| 30 | 0.988 | 0.916 | 0.073 | 0.257 |
| 40 | 0.982 | 0.904 | 0.063 | 0.232 |



Gambar 4.13 *Graph loss* 20 *epoch* Gambar 4.14 *Graph loss* 30 *epoch*

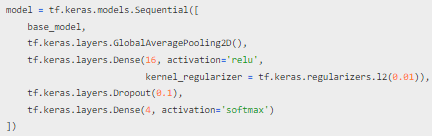


Gambar 4.15 *Graph loss* 40 *epoch*

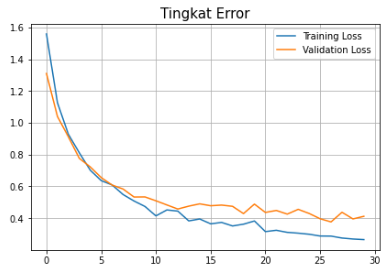
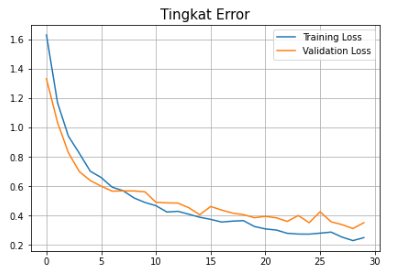
Dapat dilihat pada tabel 4.3 hasil penambahan *epoch* pada *model* penyakit buah jambu masih memiliki jarak yang jauh antara nilai *training loss* dan *validation loss* sehingga model memiliki masalah *overfitting* seperti yang terlihat pada *graph loss* gambar 4.13, gambar 4.14, dan gambar 4.15. Menurut (Novindasari, 2020) “*Model* yang memiliki *error* rendah pada *training* *data* akan tetapi memiliki *error* yang tinggi pada *test data*. Kondisi seperti ini disebut kondisi *overfitting*, dimana model kita memiliki *generalization error* yang tinggi.”

1. *Kernel regularizers*

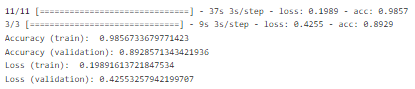
Regularisasi adalah teknik yang digunakan untuk melakukan modifikasi pada *model neural network* yang bertujuan untuk mengurangi *generalization error*, bukan mengurangi *training error* seperti peran *loss function.* (Novindasari, 2020)*.* Ada dua jenis *kernel regularizers* yaitu, regularisasi L1 yang mempertimbangkan bobot(*weight*) *layer* apa adanya sedangkan regularisasi L2 mempertimbangkan kuadrat bobot (*weight)*. (Darshan, 2022). Pada penelitian ini menggunakan *kernel regularizers* L2 sebesar 0.01, *dropout* sebesar 0.1 – 0.2, dan *epoch* sebesar 30.



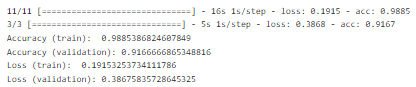
Gambar 4.16 *Source code kernel regularizers*



Gambar 4.17 *Graph loss kernel regularizers, dropout* 0.1dan 0.2



Gambar 4.18 *Model evaluate kernel regularizers, 0.1 dropout*



Gambar 4.19 *Model evaluate kernel regularizers, 0.2 dropout*

Dapat dilihat pada gambar 4.17 hasil *training loss* dan *validation loss* setelah diberi *kernel regularizers* yang menggunakan *dropout* 0.1 dan 0.2berhasil mengurangi jarak nilai *loss* antara *training loss* dan *validation loss* (*overfitting)*. Kemudian hasil *model evaluate* menyatakan bahwa *model* yang menggunakan *dropout* sebesar 0.2 memiliki hasil yang lebih baik, dengan nilai *training accuracy* sebesar 98.8%, *validation accuracy* sebesar 91.6%, *training loss* sebesar 19.1%, dan *validation loss* sebesar 38.6%.

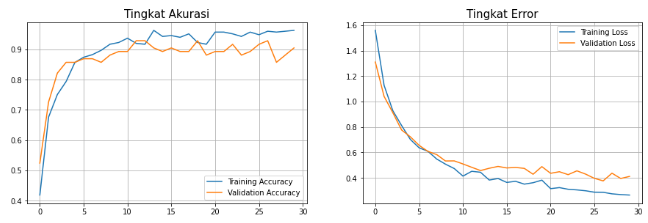
## 4.5 *Evaluation*

Tahap evaluasi yang dilakukan pada penelitian ini adalah dengan melihat grafik dari *model* yang telah dilatih dan melihat hasil *confusion matrix.*

Hasil *training* buah jambu

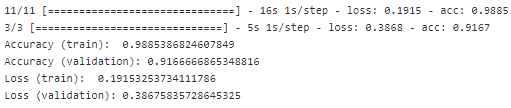
Setelah dilakukan proses *training* berikut ini adalah hasil dari *training graph* dan *confusion matrix* dari model buah jambu yang menggunakan 30 *epochs,* 0.2 *dropout,* dan *kernel regularizers* L2 sebesar 0.01.

1. *Training Graph*



Gambar 4.20 *Training graph model* buah jambu

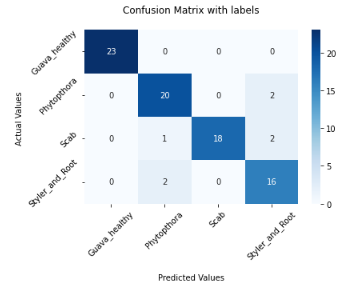
Dari hasil *training graph* tingkat *loss* dapat dilihat bahwa *model* berhasil dikurangi *overfitting* setelah menambahkan *kernel regularizers.*



Gambar 4.21 Training *evaluation model* buah jambu

Dari gambar 4.19 *training evaluation* diatas menyatakan bahwa model memiliki nilai *training accuracy* 98,8*%* dan *validation accuracy* 91.6%, kemudian nilai *training loss* sebesar 19.1%, dan nilai *validation loss* sebesar 38.6%.

1. *Confusion Matrix*



Gambar 4.22 *Confusion matrix* *model* buah jambu

Berikut ini perhitungan akurasi dari *confusion matrix* model buah jambu pada gambar 4.22:

Akurasi =

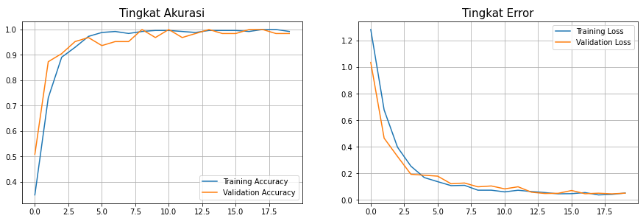
=

Jadi, hasil akurasi model penyakit pada buah jambu biji dari jumlah data validasi sebanyak 84 data adalah 91,6%

Hasil *training* daun jambu

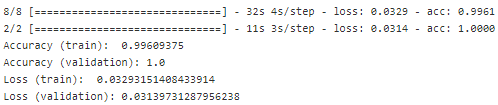
Berikut adalah hasil *training graph* dan *confusion matrix* dari model penyakit daun jambu biji. Proses *training* menggunakan 20 *epochs,* dan tanpa menggunakan *dropout.*

*Training Graph*



Gambar 4.23 *Training graph model* daun jambu

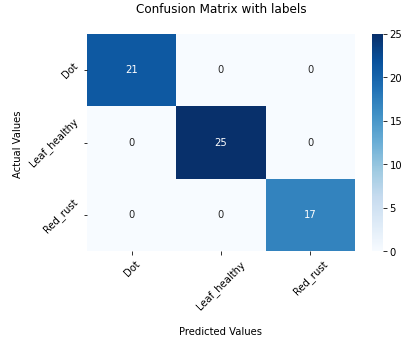
Dapat dilihat dari hasil *training graph* mengindikasikan bahwa model memiliki performa yang baik dan tidak ada indikasi model mengalami *underfitting* ataupun *overfitting*.



Gambar 4.24 *Training evaluation model* daun jambu

Dari gambar *training evaluation* diatas menyatakan bahwa model memiliki nilai *training accuracy* 99.66%dan *validation accuracy* yang mencapai 100%, nilai *training loss* yang hanya 3,29%, dan nilai *validation loss* yang hanya 3.13%.

*Confusion Matrix*



Gambar 4.25 *Confusion matrix model* daun jambu

Berikut ini perhitungan akurasi dari *confusion matrix* model daun jambu pada gambar 4.25:

Akurasi =

=

Jadi, hasil akurasi model penyakit pada daun jambu biji dari jumlah data validasi sebanyak 63 data adalah 100%.

## *Deployment*

1. Tampilan aplikasi

Halaman *Splash*

Halaman *splash* adalah tampilan awal yang muncul ketika aplikasi dibuka. Berikut ini gambar 4.26 tampilan halaman splash.



Gambar 4.26 Halaman *splash*

Halaman *onboarding*

Halaman *onboarding* adalah halaman yang menjelaskan fitur-fitur utama pada aplikasi, berikut ini gambar 4.27 dan gambar 4.28 yang merupakan halaman *onboarding.*



Gambar 4. Halaman *Onboarding* 1dan 2



Gambar 4.28 Halaman *Onboarding* 3

Halaman utama

Halaman utama aplikasi menampilkan dua buah tombol untuk ke halaman klasifikasi, kemudian terdapat daftar penyakit pada daun dan buah jambu. Berikut ini gambar 4.29 halaman utama.

Gambar 4. Halaman utama

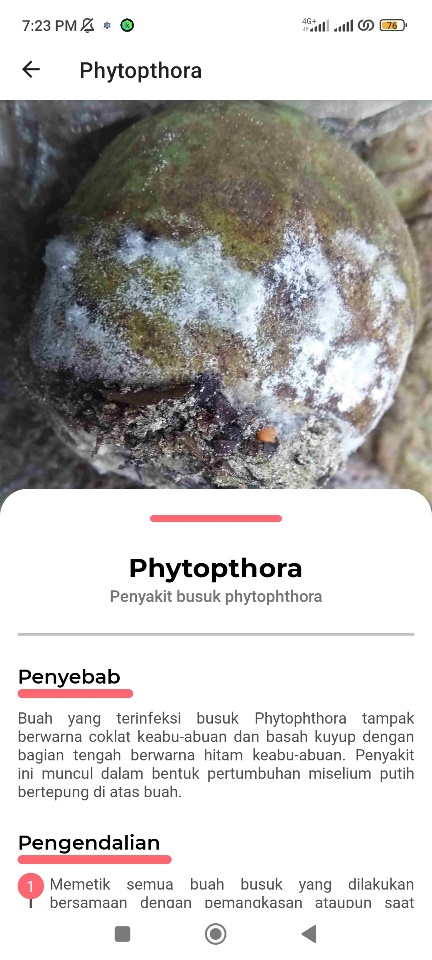
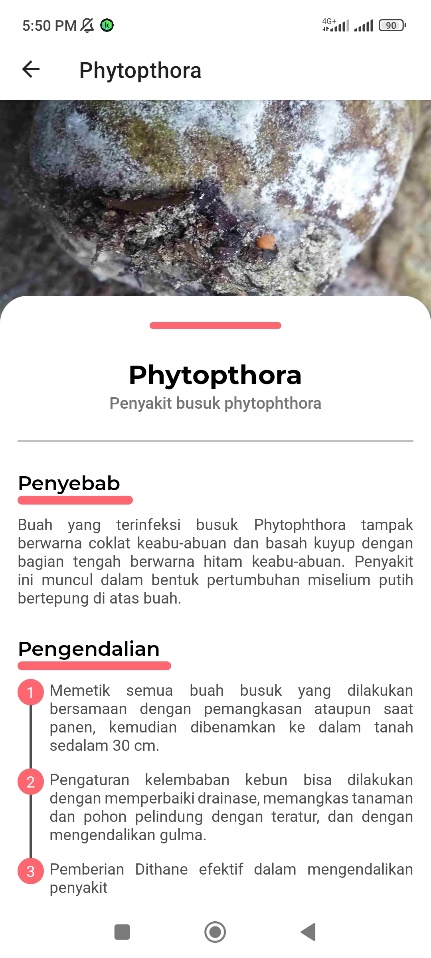
Halaman klasifikasi penyakit daun dan jambu

Halaman ini merupakan fitur utama dari aplikasi yang dapat melakukan klasifikasi jenis penyakit pada daun dan buah jambu. Terdapat fitur *flash light* secara otomatis, manual, ataupun tidak menggunakan *flash light*. Untuk melihat hasil klasifikasi bisa secara manual menekan tombol *shutter* atau bisa dengan deteksi otomatis, kemudian hasil klasifikasi dapat ditekan agar berpindah ke halaman detail penyakit. Berikut ini gambar 4.30 tampilan dari halaman klasifikasi.



Gambar 4.30 Halaman klasifikasi penyakit

Halaman detail penyakit

Pada halaman detail penyakit akan menampilkan gambar penyakit kemudian penyebab penyakit tersebut menyerang daun atau buah jambu, dan akan mejelaskan pengendalian yang bisa dilakukan dalam membasmi penyakit tersebut. Berikut ini gambar 4.31 halaman detail.

*Link* unduh aplikasi

Gambar 4. Halaman detail penyakit

Berikut ini tautan untuk unduh aplikasi **bit.ly/guavacare** atau melalui *Qr Code* berikut:



Gambar 4.32 *QR Code* unduh aplikasi

# BAB V

**KESIMPULAN DAN SARAN**

## Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan berikut ini beberapa kesimpulan yang didapat:

* 1. Peneliti berhasil menerapkan metode *supervised learning* dengan menggunakan algoritma *Convolutional Neural Network* dan arsitektur *MobileNetV2.* Hasil Model daun jambu memiliki performa yang sangat baik dan tidak *overfitting* ataupun u*nderfitting* dengan hasil evaluasi model *training accuracy* sebesar 99.6%, *validation accuracy* sebesar 100%, *training loss* sebesar 3.2%, dan *validation loss* sebesar 3.1%. Hasil *confusion matrix* memiliki akurasi 100% dari jumlah data validasi sebanyak 63 data.

Sedangkan model buah jambu berhasil dikurangi *overfitting* dengan menggunakan *dropout* 0.2 dan *kernel regularizers* L2 sebesar 0.01. Hasil *evaluasi model* memiliki nilai *training accuracy* 98,8*%* dan *validation accuracy* 91.6%, kemudian nilai *training loss* sebesar 19.1%, dan nilai *validation loss* sebesar 38.6%. Hasil *confusion matrix* *model* buah jambu memiliki akurasi 91.6% dari jumlah data validasi sebanyak 84 data.

1. Pada penelitian ini, peneliti berhasil menerapkan *model* pada aplikasi berbasis *mobile*.

## Saran

Berdasarkan hasil penelitian yang telah dilakukan berikut ini beberapa saran yang diberikan pada penelitian ini:

1. Diharapkan pada penelitian selanjutnya dapat menambah kelas jenis penyakit pada daun dan buah jambu, ataupun mungkin menambahkan kelas untuk melakukan klasifikasi hama yang menyerang pada tanaman jambu biji.
2. Diharapkan pada penelitian selanjutnya dapat melakukan variasi lagi di augmentasi data dan *parameter* ketika proses *modeling,* sehingga dapat meningkatkan akurasi *model* dan menurunkan nilai *training loss* dan *validation loss* lebih baik lagi terutama pada *model* buah jambu.
3. *Model* yang telah dikonversi menjadi *file* yang berekstensi *.tflite* dapat disimpan secara *cloud*, sehingga dapat lebih memperkecil ukuran aplikasi *mobile.*

# DAFTAR PUSTAKA

Amaratunga, T. (2021). Deep Learning on Windows. In *Deep Learning on Windows*. https://doi.org/10.1007/978-1-4842-6431-7

Andi, J. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, *1*(1), 1–8.

*Apa itu Penyakit Tanaman*. (2022, December 6). https://agribisnis.uma.ac.id/2022/12/06/apa-itu-penyakit-tanaman/

AWS Amazon. (n.d.). *Apa itu IDE? Penjelasan IDE - AWS*. Retrieved December 7, 2022, from https://aws.amazon.com/id/what-is/ide/

Badan Pusat Statistik Provinsi Jawa Barat. (2022). Produksi Hortikultura Buah dan Sayur Tahunan Provinsi Jawa Barat 2021. *BPS Provinsi Jawa Barat*, 1–128. https://jabar.bps.go.id/publication/2022/12/23/1bb94ee2b41974c0e1cb3ab8/produksi-hortikultura-buah-dan-sayur-tahunan-provinsi-jawa-barat-2021.html

Ben Lutkevich. (2022). *What is Implementation?* https://www.techtarget.com/searchcustomerexperience/definition/implementation

BPS Jawa Barat. (2022a, July 25). *Curah Hujan di Stasiun Pengamatan Klimatologi Bogor Menurut Bulan (mm), 2019-2022*. https://jabar.bps.go.id/indicator/151/430/1/-curah-hujan-di-stasiun-pengamatan-klimatologi-bogor-menurut-bulan.html

BPS Jawa Barat. (2022b, July 25). *Curah Hujan di Stasiun Pengamatan Meteorologi Citeko Menurut Bulan (mm), 2019-2021*. https://jabar.bps.go.id/indicator/151/433/1/curah-hujan-di-stasiun-pengamatan-meteorologi-citeko-menurut-bulan.html

BPS Jawa Barat. (2022c, July 25). *Jumlah Curah Hujan di Stasiun Pengamatan Meteorologi Jatiwangi Menurut Bulan (mm), 2019-2021*. https://jabar.bps.go.id/indicator/151/438/1/jumlah-curah-hujan-di-stasiun-pengamatan-meteorologi-jatiwangi-menurut-bulan.html

BPS Jawa Barat. (2022d, July 25). *Pengamatan Curah Hujan di Stasiun Pengamatan Geofisika Bandung Menurut Bulan (mm), 2019-2021*. https://jabar.bps.go.id/indicator/151/425/1/pengamatan-curah-hujan-di-stasiun-pengamatan-geofisika-bandung-menurut-bulan.html

Cahyono, B. (2010). *Sukses Budi Daya Jambu Biji di Pekarangan dan Perkebunan* (1st ed.). Andi.

Daljoeni, N. (1983). *Pokok-pokok klimatologi*. Alumni.

Danukusumo, K. P. (2017). *IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI CITRA CANDI BERBASIS GPU*.

Darshan M. (2022, June 25). *How do Kernel Regularizers work with neural networks?* https://analyticsindiamag.com/kernel-regularizers-with-neural-networks/

Fiki Syihabirawan. (2020, November 25). *Convolutional Neural Network (CNN) - Kotakode.com | Komunitas Developer Indonesia*. https://kotakode.com/blogs/2707/Convolutional-Neural-Network-(CNN)

Fischer, G., & Melgarejo, L. M. (2021). Ecophysiological aspects of guava (Psidium guajava L.). A review. *Revista Colombiana de Ciencias Hortícolas*, *15*(2), 0–3. https://doi.org/10.17584/rcch.2021v15i2.12355

Fitriani, M. (2019). *IMPLEMENTASI ASSOCIATION RULE DENGAN ALGORITMA APRIORI PADA DATA PEMINJAMAN BUKU UPT PERPUSTAKAAN UNIVERSITAS LAMPUNG MENGGUNAKAN METODOLOGI CRISP-DM*. *8*(5), 55.

Geeksforgeeks. (2023). *Kotlin Programming Language - GeeksforGeeks*. https://www.geeksforgeeks.org/kotlin-programming-language/

Hafizhan Aliady Afif. (2020, April 28). *Membuat klasifikasi gambar(images), Menggunakan Keras-Tensorflow (tf.keras) di Python | by Hafizhan Aliady Afif | Medium*. https://medium.com/@hafizhan.aliady/membuat-klasifikasi-gambar-images-menggunakan-keras-tensorflow-tf-keras-dan-python-53f7ae953cea

Hariyono. (2021). *Panduan Praktikum : Teknologi Budidaya Tanaman*. 41.

Hasanah, M. A., Soim, S., & Handayani, A. S. (2021). Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir. *Journal of Applied Informatics and Computing*, *5*(2), 103–108. https://doi.org/10.30871/jaic.v5i2.3200

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. http://arxiv.org/abs/1704.04861

Ida Novindasari. (2020, October 12). *Mengapa Diperlukan Regularisasi pada Model Neural Network? | by Ida Novindasari | Medium*. https://idanovinda.medium.com/mengapa-diperlukan-regularisasi-pada-model-neural-network-d622ed98f9a8

*Indonesia: mobile OS share 2022 | Statista*. (n.d.). Retrieved November 30, 2022, from https://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/

Jauhari, A. F. (2022). *KLASIFIKASI JENIS BERAS MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK PADA ARSITEKTUR MOBILENET* (Vol. 2, Issue 8.5.2017).

Kumar, M., Tomar, M., Amarowicz, R., Saurabh, V., Nair, M. S., Maheshwari, C., Sasi, M., Prajapati, U., Hasan, M., Singh, S., Changan, S., Prajapat, R. K., Berwal, M. K., & Satankar, V. (2021). Guava ( Psidium guajava L .) Leaves : Nutritional Composition. *Foods*, *10*(752), 1–20. https://doi.org/https://doi.org/ 10.3390/foods10040752

Kurniawan, D., & Yasir, M. (2022). Optimization Sentimen Analysis using CRISP-DM and Naive Bayes Methods Implemented on Social Media. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, *6*(2), 74. https://doi.org/10.22373/cj.v6i2.12793

Kurniawati, D. (2021). *Mengenal Penyakit Busuk Buah (Phytophthora palmivora) Pada Kakao - Website Resmi DPKP DIY*. https://dpkp.jogjaprov.go.id/baca/Mengenal+Penyakit+Busuk+Buah+%28Phytophthora+palmivora%29+Pada+Kakao/180521/ce8738c6d0c64e87b2a7ca431858ce824c0b04f417d21435726d70242c657de8316

*Machine Learning adalah Bagian dari Artificial Intelligence,...* (2020). https://www.dqlab.id/perbedaan-machine-learning-dan-ai

Mary Ellen Ellis. (2021, July 30). *What Is Stylar End Rot: Common Stylar End Breakdown Causes*. https://www.gardeningknowhow.com/edible/fruits/citrus/managing-fruit-with-stylar-end-rot.htm

*Meet Android Studio  |  Android Developers*. (n.d.). Retrieved December 7, 2022, from https://developer.android.com/studio/intro

Minut, M. (2021). *Fruits-262 | Kaggle*. https://www.kaggle.com/datasets/aelchimminut/fruits262

Misra, A. K. (2006). Guava Diseases — their Symptoms, Causes and Management. *Diseases of Fruits and Vegetables: Volume II*, *December*, 81–119. https://doi.org/10.1007/1-4020-2607-2\_4

ML Wiki. (2014). *CRISP-DM - ML Wiki*. http://mlwiki.org/index.php/CRISP-DM

Mubarok, H. (2019). Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN). *Universitas Islam Negeri Maulana Malik Ibrahim Malang*, *3*(1), 10–12.

Mubin, N., Khairani, H. S., Triwidodo, H., & Bandi, D. (2021). *Pengendalian Hama dan Penyakit pada Komoditas Jambu Mete, Kelapa,Sirih, dan Pinang*.

Murya, Y. (2014). *Pemrograman Android Black Box*. Jasakom.

Mutsani, H. (n.d.). *10 Definisi Pengertian Pertanian Menurut Para Ahli Dan Contohnya Lengkap | Forbes.Id*. Retrieved November 30, 2022, from https://forbes.id/10-definisi-pengertian-pertanian-menurut-para-ahli-dan-contohnya-lengkap/

Noamaan Abdul Azeem. (2022). *Guava Dataset | Kaggle*. https://www.kaggle.com/datasets/noamaanabdulazeem/guava-dataset

Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(2), 697–711. http://ejurnal.tunasbangsa.ac.id/index.php/jsakti/article/view/369

Nurhikmat, T. (2018). IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK. *Journal of Controlled Release*, *11*(2), 430–439.

Omkar Manohar Dalvi. (n.d.). *Guava Disease Dataset (4 types) | Kaggle*. Retrieved May 25, 2023, from https://www.kaggle.com/datasets/omkarmanohardalvi/guava-disease-dataset-4-types

Prihatman, K. (2000). Jambu Biji / Jambu Batu. *Tentang Budidaya Pertanian*, 1–17.

Qolbiyatul Lina. (2019, January 2). *Apa itu Convolutional Neural Network? | by QOLBIYATUL LINA | Medium*. https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4

Rajbongshi, A., Sazzad, S., Shakil, R., Akter, B., & Sara, U. (2022). A comprehensive guava leaves and fruits dataset for guava disease recognition. *Data in Brief*, *42*, 108174. https://doi.org/10.1016/J.DIB.2022.108174

Ramba, L. S. (2020). *Perancangan Sistem Home Automation Dengan Kendali Perintah Suara Menggunakan Deep Learning Convolutional Neural Network (Dl-Cnn)*.

Rizqi Okta Ekoputris. (2018, May 9). *MobileNet: Deteksi Objek pada Platform Mobile | by Rizqi Okta Ekoputris | Nodeflux | Medium*. https://medium.com/nodeflux/mobilenet-deteksi-objek-pada-platform-mobile-bbbf3806e4b3

Roy Perrott, Jules Jannick, P. M. S. (2023). *Horticulture | Definition, Types, Techniques, & Uses | Britannica*. https://www.britannica.com/science/horticulture

Ruomenson Dedi Jefri Bakara, F. K. (2020). Pendampingan Petani dalam Pengendalian Hama dan Penyakit Jambu Biji ( Psidium guajava L . ) di Desa Cibening , ( Farmer Association in Pest and Disease Control on Guava ( Psidium guajava L . ) in Cibening Vilage , Pamijahan. *Jurnal Pusat Inovasi Masyarakat*, *2*(1), 131–143.

Samuel Sena. (2017, November 13). *Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN) | by Samuel Sena | Medium*. https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94

Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika Dan Sistem Informasi*, *6*(2), 311–323. https://doi.org/10.28932/jutisi.v6i2.2688

Sarin, S. (2019). *Exploring Data Augmentation with Keras and TensorFlow | by Sumit Sarin | Towards Data Science*. https://towardsdatascience.com/exploring-image-data-augmentation-with-keras-and-tensorflow-a8162d89b844

Setiawan, A. (2021). *Augmentasi Data Pada Computer Vision | by Adhi Setiawan | Data Folks Indonesia | Medium*. https://medium.com/data-folks-indonesia/augmentasi-data-pada-computer-vision-45c5ebe10e8f

Tensorflow. (2022). *TensorFlow Lite*. https://www.tensorflow.org/lite/guide

*Top Countries/Markets by Smartphone Penetration & Users | Newzoo*. (n.d.). Retrieved November 30, 2022, from https://newzoo.com/insights/rankings/top-countries-by-smartphone-penetration-and-users

Tsang, S.-H. (2018). *Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification) | by Sik-Ho Tsang | Towards Data Science*. https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

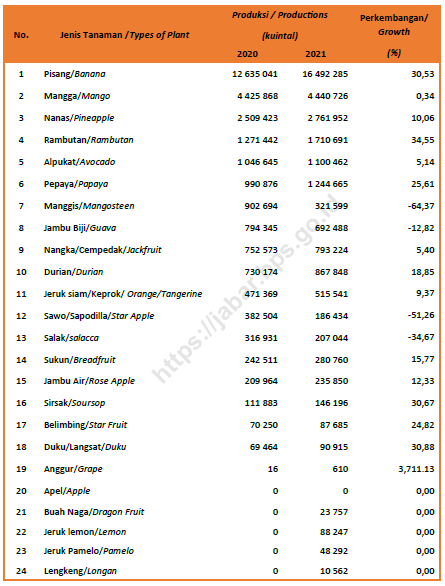
Vaughan, J. (2018). *What is TensorFlow? | Definition from TechTarget*. https://www.techtarget.com/searchdatamanagement/definition/TensorFlow

*What Is a Convolutional Neural Network? | 3 things you need to know - MATLAB & Simulink*. (n.d.). Retrieved March 28, 2023, from https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html

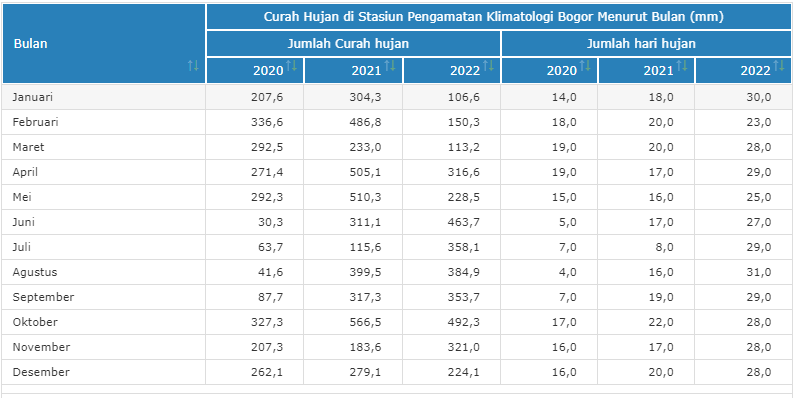
Wiesel, H. &. (1968). RECEPTIVE FIELDS AND FUNCTIONAL ARCHITECTURE OF MONKEY STRIATE CORTEX. *Journal of Physiology*, *195*(1), 215–243.

# LAMPIRAN

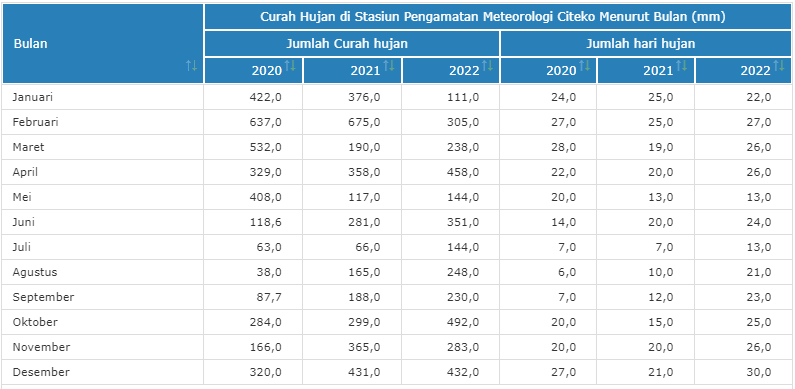
Lampiran A Produksi Tanaman Buah-buahan Tahunan Jawa Barat Tahun 2020 dan 2021



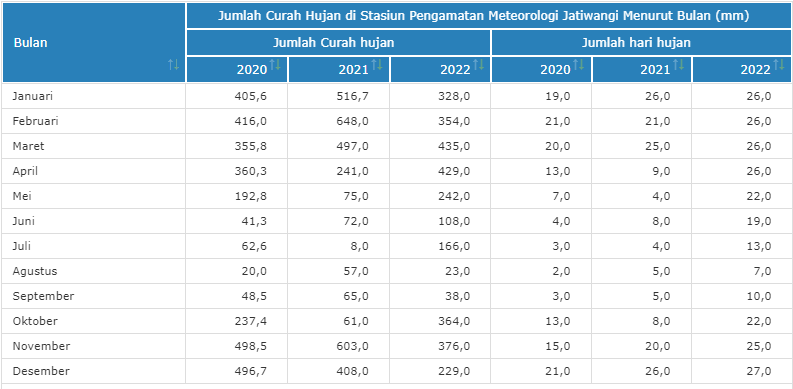
Lampiran B Curah Hujan di Stasiun Pengamatan Klimatologi Bogor Menurut Bulan (mm), 2020-2022



Lampiran C Curah Hujan di Stasiun Pengamatan Meteorologi Citeko Menurut Bulan (mm), 2020-2022



Lampiran D Jumlah Curah Hujan di Stasiun Pengamatan Meteorologi Jatiwangi Menurut Bulan (mm), 2020-2022



Lampiran E Pengamatan Curah Hujan di Stasiun Pengamatan Geofisika Bandung Menurut Bulan (mm), 2019-2022

