

**IMPLEMENTASI DEEP LEARNING PADA DETEKSI JENIS
TANAH BERDASARKAN WARNA MENGGUNAKAN
ALGORITMA YOLO (STUDI KASUS : KABUPATEN
SORONG)**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Komputer Pada Fakultas Teknik Universitas
Muhamadiyah Sorong**



Disusun oleh :

Titin Fadila Ayunadin

Nim : 202055202020

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS MUHAMADIYAH SORONG**

2024

LEMBAR PERSETUJUAN TUGAS AKHIR

**IMPLEMENTASI DEEP LEARNING PADA DETEKSI JENIS TANAH
BERDASARKAN WARNA MENGGUNAKAN ALGORITMA YOLO (STUDI
KASUS : KABUPATEN SORONG)**

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer
Pada Fakultas Teknik Universitas Muhammadiyah Sorong**

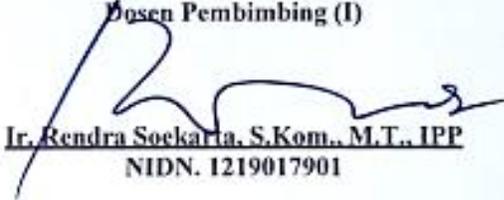
**Disusun Oleh :
TITIN FADILA AYUNADIN
NIM : 202055202020**



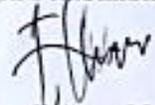
Sorong, Juli 2024

Menyetujui,

Dosen Pembimbing (I)

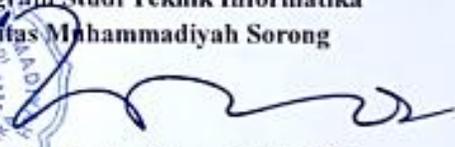

Ir. Rendra Soekarta, S.Kom., M.T., IPP
NIDN. 1219017901

Dosen Pembimbing (II)


Suhardi Aras, M. Kom
NIDN. 1405107901

Mengetahui,

**Ketua Program Studi Teknik Informatika
Universitas Muhammadiyah Sorong**


Ir. Rendra Soekarta, S.Kom., M.T., IPP
NIDN. 1219017901

LEMBAR PERSETUJUAN TUGAS AKHIR

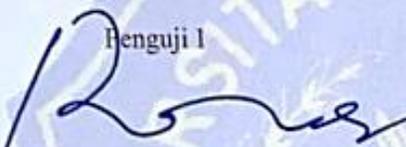
**IMPLEMENTASI DEEP LEARNING PADA DETEKSI JENIS TANAH
BERDASARKAN WARNA MENGGUNAKAN ALGORITMA YOLO
(STUDI KASUS : KABUPATEN SORONG)**

**Disusun Oleh :
TITIN FADILA AYUNADIN
NIM : 202055202020**

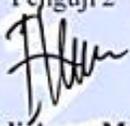
**Tugas Akhir ini telah dipertahankan di depan Tim Penguji pada tanggal 3
Juli 2024 dan yang bersangkutan dinyatakan LULUS serta berhak
memperoleh gelar sarjana Komputer.**

Tim Penguji

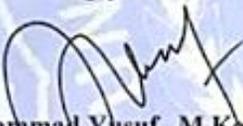
Penguji 1


Ir. Rendra Soekarta, S.Kom., M.T., IPP,
NIDN. 1219017901

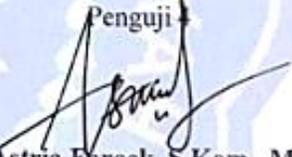
Penguji 2


Suhardi Aras, M.Kom.
NIDN. 1402098101

Penguji 3


Muhammad Yusuf, M.Kom.
NIDN. 1402098101

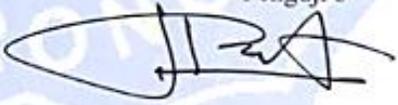
Penguji 4


Dewi Astria Faroek, S.Kom., M.Kom.
NIDN. 1402069601

Penguji 5


Ir. H. Irman Amri, S.T., M.T., IPM,
NIDN. 1212047601

Penguji 6


Muhammad Rizki Setyawan, S.Kom., M.Kom.
NIDN. 1425089601

**Mengetahui,
Ketua Program Studi Teknik Informatika
Universitas Muhammadiyah Sorong**



Ir. Rendra Soekarta, S.Kom., M.Kom., IPP,
NIDN. 1219017901

MOTO DAN PERSEMBAHAN

MOTO

“ Masa Depan Kita Gemilang, The Future Is Yours Do Your Best, Berbuat Yang Baik Jangan Sakiti Orang”

(Prabowo Subianto”

“Allah tidak membebani seseorang itu melainkan sesuai dengan kesanggupannya”

(Q.S Al-Baqarah: 286)

PERSEMBAHAN

Skripsi ini penulisan persembahkan :

Untuk kedua orang tua saya bapak ayunadin dan ibu amina yang selalu mendoakan yang baik-baik dan menjadikan monitivasi saya dalam menyelesaikan skripsi ini dengan baik, saya bersembahkan skripsi ini untuk bapak ibu.

Bapak dan Ibu Dosen Teknik Informatika yang telah membimbing dan mengarahkan saya untuk menyelesaikan skripsi ini.

Kepada diri saya sendiri, Titin fadila ayunadin karena terus berusaha dan sudah berjuang sampai sejauh ini.

ABSTRAK

Pada umumnya Identifikasi jenis tanah sangat penting untuk pertanian, manajemen lingkungan, dan perencanaan pembangunan. Namun, identifikasi manual memerlukan waktu yang lama dan sering kali kurang akurat. Dengan kemajuan teknologi deteksi warna dan analisis citra, penelitian ini mengembangkan sistem deteksi otomatis yang lebih efisien dan akurat. Pada penelitian yang akan dilakukan menggunakan Algoritma YOLO dalam mendeteksi objek secara cepat dan tepat. Penelitian ini menggunakan dataset berupa gambar berformat JPG yang terdiri dari tiga kelas yaitu terdiri dari, tanah Alluvian, tanah histosol dan tanah latosol. Total dataset yang digunakan berjumlah 900 gambar. Evaluasi model menggunakan confusion matrix menunjukkan tingkat akurasi sebesar 80%. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu mengidentifikasi jenis tanah dengan akurasi yang cukup tinggi.

Kata Kunci – *Deteksi objek, Algoritma Yolov8, Jenis Tanah.*

KATA PENGANTAR

Assalaamualaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah SWT, yang senantiasa memberikan rahmat dan petunjuk sehingga penulis dapat menyelesaikan proposal ini dengan baik. Proposal ini disusun sebagai tugas akhir belajar dan syarat guna memperoleh derajat S-1 pada Program Studi Teknik Informatika, Fakultas Teknik Universitas Muhammadiyah Sorong dengan judul **“IMPLEMENTASI DEEP LEARNING PADA DETEKSI JENIS TANAH BERDASARKAN WARNA MENGGUNAKAN ALGORITMA YOLO (STUDI KASUS : KABUPATEN SORONG)”**

Berkenan dengan penulisan penelitian ini, penulis ingin menyampaikan ucapan terimakasih dan penghargaan setinggi-tingginya atas bantuan dan dukungannya kepada:

1. Ayah, Ibu, Kakak, Adek dan Sepupu-Sepupu atas ucapan Terimakasih yang sebesar-besarnya karena telah mendoakan saya, Memberikan semangat dan kasih sayang yang tidak pernah ada habisnya, selalu ada dalam berbagai kondisi serta dukungan untuk menyelesaikan skripsi ini.
2. Bapak Dr. H. Muhammad Ali, M.M., M.H sebagai Rektor Universitas Muhammadiyah Sorong yang telah memberikan kesempatan kepada penulis untuk melanjutkan pendidikan di Universitas Muhammadiyah Sorong.
3. Bapak Dr. Ir. H. Hendrik Pristitianto, S.T., M.T., IPM sebagai dekan fakultas Teknik Universitas Muhammadiyah Sorong, Yang telah Menyediakan fasilitas dan kemudahan kepada penulis selama studi di fakultas Teknik Universitas Muhammadiyah Sorong.

4. Bapak Ir. H. Rendra Soekarta, S.Kom., M.T., IPP sebagai Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Sorong yang membimbing penulis selama mengikuti pendidikan di Fakultas Teknik Program Studi Informatika Universitas Muhammadiyah Sorong.
5. Bapak Ir. H. Rendra Soekarta, S.Kom., M.T., IPP. Sebagai Dosen Pembimbing I dan Juga Bapak Suhardi Aras M.Kom. Sebagai Dosen Pembimbing II yang telah meluangkan waktunya di sela-sela kesibukan mau membimbing penulis agar dapat menyelesaikan penelitian ini.
6. Bapak dan Ibu Dosen yang berada di lingkungan Fakultas Teknik khususnya Program studi Teknik Informatika yang membimbing dan memberikan ilmunya kepada penulis dengan sabar.
7. Untuk Bayu Sasena sebagai suport sistem saya ucapkan Terimakasih yang sebesar-besarnya karena telah mendoakan saya, dan Memberikan semangat, selalu ada dalam berbagai kondisi serta dukungan untuk menyelesaikan skripsi ini.
8. Untuk teman – teman angkatan 2020 sebagai partner terbaik dalam mengerjakan skripsi ini saya ucapkan terima kasih yang sebesar – besarnya.
9. Untuk teman – teman dekat saya, Putri Ussalama, Fridilla N.A.L Tampilang, Gunawan Selayar, Andi Muhammad Fitra Passaribu, Eka Nirmla Nursabar, Ubaidillah Fadhil, Muhammad Arif, Putriana Tantra, dan Rahma Na'afiu La apa atas dukungannya dan doanya untuk menyelesaikan skripsi ini.

Atas bantuan semangat dan bimbingan tersebut sekali lagi penulis ingin mengucapkan terimakasih. Semoga Allah SWT memberikan ganjaran pahala atas amal yang diberikan kepada penulis. Aamiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Sorong, Juni 2024

Titin Fadila Ayunadin

DAFTAR ISI

LEMBAR.....	II
PERSETUJUAN.....	II
MOTO DAN PERSEMBAHAN.....	III
ABSTRAK.....	IV
KATA PENGANTAR.....	V
DAFTAR ISI.....	VIII
DAFTAR TABEL.....	XI
DAFTAR GAMBAR.....	XI
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.1 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5
BAB II.....	7
LANDASAN TEORI.....	7
2.1 VOSviewer.....	7
2.2 State of The Art.....	8
2.3 Literatur Terkait.....	16
2.3.1 Tanah.....	16
2.3.2 Histosol.....	16
2.3.3 Alluvial.....	17
2.3.4 Latosol.....	18
2.3.5 Deep Learning.....	19
2.3.6 python.....	20
2.3.7 Objek Detection.....	20

2.3.8	You Only Look Once.....	21
2.3.9	Google Colaboratory.....	23
2.3.10	Confusion Matrix.....	23
2.3.11	ONNX (Open Neural Network Exchange.....	25
2.3.12	Extreme programming.....	26
2.3.13	android studio.....	27
2.3.14	android.....	27
2.3.15	PyTorch.....	28
2.3.16	Use case diagram.....	28
2.3.17	Flowchart.....	30
2.3.19	Black Box Testing.....	32
2.3.18	Usability Testing.....	33
BAB III.....		36
METODE PENELITIAN.....		36
3.1	Lokasi dan Waktu Penelitian.....	36
3.1.1	Lokasi Penelitian.....	36
3.1.1	Waktu Penelitian.....	36
3.2	Tahapan Penelitian.....	37
3.2.1	Alur Penelitian.....	37
3.3	Metode Pengumpulan Data.....	39
3.4	Pengembangan Model Yolo.....	39
3.5	Pengembangan Sistem Extreme Programming.....	42
BAB IV.....		52
HASIL DAN PEMBAHASAN.....		52
4.1	Hasil Pengumpulan Data.....	52
4.2	Implementasi Algoritma Yolo V8.....	52
4.2.1	Pengumpulan Dataset.....	52
4.2.2	Labeling Dataset.....	53
4.2.3	Split Dataset.....	54
4.2.4	Preprocessing.....	54

4.2.5	Konfigurasi Parameter.....	55
4.2.6	Training.....	56
4.2.7	Evaluasi.....	57
4.2.8	Hasil Deteksi.....	66
4.2.9	Saving Model.....	67
4.3	Implementasi Pada Android.....	68
4.3.1	Planning.....	68
4.3.2	Desain.....	68
4.3.3	Coding.....	71
4.3.4	Testing.....	71
4.4	Usability Testing.....	73
BAB V.....		81
PENUTUP.....		81
4.1	Kesimpulan.....	81
4.2	Saran.....	81
DAFTAR PUSTAKA.....		83

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian.....	15
Tabel 2.2 Confusion Matrix	24
Tabel 2.3 Simbol – Simbol Use Case	29
Tabel 2.4 Simbol – Simbol Flowchart.....	30
Tabel 3.1 Waktu Pelaksanaan Penelitian.....	37
Tabel 3.2 Langkah – Langkah Pengujian Sistem.....	50
Tabel 4.1 Jumlah Dataset	52
Tabel 4.2 <i>Confusion Matrix</i>	52
Tabel 4.3 Hasil Black Box Testing	74
Tabel 4.4 Skor Kuisisioner.....	78
Tabel 4.5 Skor Pertanyaan 1.....	80
Tabel 4.6 Skor Pertanyaan 2	80
Tabel 4.7 Skor Pertanyaan 3	81
Tabel 4.8 Skor Pertanyaan 4	81
Tabel 4.9 Skor Pertanyaan 5.....	82

DAFTAR GAMBAR

Gambar 2.1 State Of The Art	7
Gambar 2.2 Tanah Histosol	17
Gambar 2.3 Tanah Alluvial.....	18
Gambar 2.4 Tanah Latosol.....	19
Gambar 2.5 Arsitektur Yolo.....	22
Gambar 2.6 Extreme Programming	27
Gambar 2.7 Black Box Testing.....	32
Gambar 3.1 Lokasi Penelitian	36
Gambar 3.2 Alur Penelitian.....	38
Gambar 3.3 Flowchart You Only Look Once.....	40
Gambar 3.4 Use Case.....	44
Gambar 3.5 Flowchart Sistem Identifikasi.....	45
Gambar 3.6 Flowchart Training.....	46
Gambar 3.7 Tampilan Splash Screen	47
Gambar 3.8 Tentang Aplikasi	48
Gambar 3.9 Camera	49
Gambar 4.1 Jenis Tanah.....	53
Gambar 4.2 Proses Pelebelan/Anotasi Gambar	53
Gambar 4.3 Split Data	54
Gambar 4.4 Resize Gambar	54
Gambar 4.5 Split Data.....	55
Gambar 4.6 Pemilihan Versi Model Yolo	56
Gambar 4.7 Konfigurasi Parameter.....	56
Gambar 4.8 Training	57
Gambar 4.9 Confusion Matrix	59
Gambar 4.10 Pecision	61
Gambar 4.11 Pecision-Recall.....	62
Gambar 4.12 Recall.....	63

Gambar 4.13 F1-Score	64
Gambar 4.14 Hasil Training mAP	65
Gambar 4.15 Deteksi Model	66
Gambar 4.16 Hasil Deteksi Menggunakan Handphone	71
Gambar 4.17 Hasil Deteksi	66
Gambar 4.18 Save Model	67
Gambar 4.19 Tampilan Splash Screen	68
Gambar 4.20 Tentang Aplikasi	69
Gambar 4.21 Camera	70
Gambar 4.22 Diagram Pertanyaan 1	73
Gambar 4.23 Diagram Pertanyaan 2	74
Gambar 4.24 Diagram Pertanyaan 3	74
Gambar 4.25 Diagram Pertanyaan 4	75
Gambar 4.26 Diagram Pertanyaan 5	76

BAB I PENDAHULUAN

1.1 Latar Belakang

Tanah merupakan material yang terdiri dari *agregat* (butiran) mineral padat yang tidak terikat secara kimia satu sama lain. Partikel padat ini merupakan bagian dari komponen organik yang terlapuk, dengan cairan dan gas menempati ruang yang ditinggalkan oleh partikel padat (Ferdeanty, Sufardi, and Arabia 2019). Selain itu Dengan kata lain, tanah adalah kumpulan partikel-partikel mineral atau hubungan yang muncul di antara partikel-partikel tersebut akibat pelapukan batuan. Tidak hanya itu Tanah juga memiliki peranan yang sangat penting bagi seluruh kehidupan di bumi, karena tanah mendukung kehidupan tumbuhan. Dengan adanya hara serta air yang menjadi penopang bagi akar untuk tetap tumbuh dan berkembang (UNY 2018).

Karakteristik Tanah di Kabupaten Sorong terdiri dari Tekstur tanah yang halus, sedang, kasar, dan bergambut menjadi salah satu ciri tanah di Kabupaten Sorong. Tekstur yang ada di wilayah Kabupaten Sorong sebagian besar mulus. Podsolik Kelabu, Podsolik Merah Kuning, Organosol, Aluvial, Tanah Kompleks, Renzina, dan Latosol merupakan jenis tanah yang banyak terdapat di Kabupaten Sorong (Su, Sitorus, and Musfira 2022).

Pada umumnya identifikasi jenis tanah memiliki peranan *krusial* dalam berbagai aspek, seperti pertanian, manajemen lingkungan, dan perencanaan pembangunan. Tanah yang berbeda memiliki karakteristik warna yang

beragam. Mengidentifikasi jenis tanah secara manual dapat menjadi tugas yang kompleks dan membutuhkan waktu, terutama jika area yang harus diidentifikasi luas. Tidak hanya itu Untuk mengetahui jenis tanah yang ada, cara yang paling mudah untuk dilakukan adalah dengan mendeteksi warna dari tanah tersebut (Puspaningrum, Nugroho, and Manggala 2020).

Kemajuan dalam teknologi deteksi warna dan analisis citra memungkinkan pengembangan sistem deteksi yang lebih canggih. Penggunaan sensor warna tinggi resolusi dan algoritma deteksi yang efisien dapat memberikan hasil yang lebih akurat dan terinci. Tidak hanya itu adanya sistem deteksi jenis tanah berdasarkan warna juga dapat menjadi media pembelajaran bagi mahasiswa atau pelajar tentang jenis-jenis tanah yang ada pada kabupaten sorong.

Pada penelitian terkait yang telah dilakukan sebelumnya menggunakan metode yang berbeda tetapi memiliki objek yang sama yaitu jenis tanah dengan judul “Penerapan Radial Basis Function Untuk Klasifikasi Jenis Tanah” kekurangan dalam penelitian ini yaitu hanya mengklasifikasi 3 jenis tanah saja dari berbagai jenis tanah yang ada di Indonesia, tidak hanya itu pada penelitian ini juga menggunakan data yang diambil dari website NRCS (*Natural Resources Conservation Services*) yang terdiri dari 36 data gambar yang dimana di bagi menjadi 2 yaitu data training dan data uji. Pada penelitian ini peneliti menggunakan klasifikasi dengan JST, dan RBF akan dilakukan ekstraksi ciri dengan menggunakan GLCM. Penelitian ini melakukan 5 kali percobaan dengan rentang nilai spread 0,1 – 2 dimana nilai spread 1 adalah

nilai terbaik untuk melakukan proses klasifikasi. Hasil pelatihan untuk iterasi mencapai maksimum 1000 dan waktu tercatat 10 detik dengan nilai spread 1. Dari performance nilai error (MSE) neuron ke 0- 10 semakin kecil artinya nilainya semakin mendekati nilai error goal yaitu 0,0000001. Hasil akhir performance nilai error pada neuron hidden ke 10. Hasil uji coba dan evaluasi dari klasifikasi ke 3 jenis tanah yang telah dilakukan dengan menggunakan hidden layer 10 mendapatkan akurasi 90,10% (Puspaningrum et al. 2020).

Identifikasi jenis tanah yang cepat dan akurat juga memiliki implikasi dalam pemetaan dan pengelolaan sumber daya alam secara umum. Ini dapat membantu dalam perencanaan penggunaan lahan yang berkelanjutan. Oleh karena itu, pengembangan sistem deteksi jenis tanah berdasarkan warna menjadi suatu kebutuhan untuk memberikan informasi yang akurat dan cepat tentang jenis tanah. Berdasarkan permasalahan dan pembahasan diatas, penulis mengambil judul “**Implementasi Deep Learning Pada Deteksi Jenis Tanah Berdasarkan Warna Menggunakan Algoritma Yolo (Studi Kasus : Kabupaten Sorong)**” yang diharapkan dapat mengatasi permasalahan tersebut.

1.1 Rumusan Masalah

Dari latar belakang permasalahan, adapun rumusan masalah yang dapat diambil dalam penelitian ini yaitu :

1. Bagaimana implementasi sistem Deteksi jenis tanah berdasarkan warna berdasarkan Metode Yolov8 ?

2. Berapa besar tingkat akurasi dari perancangan sistem deteksi jenis tanah yang dilakukan ?

1.3 Tujuan Penelitian

Adapun tujuan dalam pembuatan proposal tugas akhir ini yaitu :

1. Merancang dan membuat sistem yang dapat mendeteksi jenis tanah berdasarkan warna menggunakan metode yolo.
2. Mengetahui tingkat akurasi dari implementasi sistem deteksi jenis tanah berdasarkan warna.

1.4 Batasan Masalah

1. Penelitian ini menggunakan 3 jenis tanah yaitu Tanah Histosol, Tanah Latosol dan Tanah Alluvial.
2. Sistem mendeteksi tanah dalam wadah.
3. Jumlah dataset yang digunakan yaitu 300 gambar pada setiap class.
4. Aplikasi dirancang berbasis android.
5. Sistem tidak memiliki database.
6. Sistem dirancang dengan multi-objek
7. Proses deteksi harus di tempat yang terang.
8. Penelitian ini menggunakan algoritma YOLO v8.
9. Dataset tidak menghilangkan background gambar.

1.5 Manfaat Penelitian

1. Sistem deteksi ini sangat bermanfaat bagi para mahasiswa

khususnya jurusan pertanian agar dapat lebih mudah membedakan jenis tanah hanya dengan menggunakan 1 tools saja.

1.6 Sistematika Penulisan

Secara keseluruhan penelitian ini terdiri dari 3 (Tiga) bab. Sistematika penulisan ini bertujuan untuk memudahkan pembaca dalam memahami penelitian ini. Isi penelitian secara garis besar sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini menjelaskan uraian tentang latar belakang penulisan yang mendasari dilaksanakannya penelitian, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, dan Sistematika Penulisan.

BAB II LANDASAN TEORI

Bab ini berisi kumpulan Landasan Teori atau Studi Pustaka yang berhubungan dengan topik tugas akhir.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam penelitian, baik dalam pengumpulan data, maupun metode yang digunakan dalam membuat proyek tugas akhir, yaitu Lokasi dan Waktu Penelitian, Alat dan Bahan, Tahapan Penelitian, dan Langkah-Langkah Penelitian.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini membahas hasil pengujian berdasarkan metode dan skenario pengujian yang telah dituliskan pada bab

sebelumnya serta pada bab ini juga menjelaskan hasil analisis terhadap hasil pengujian yang telah dilakukan.

BAB V : KESIMPULAN

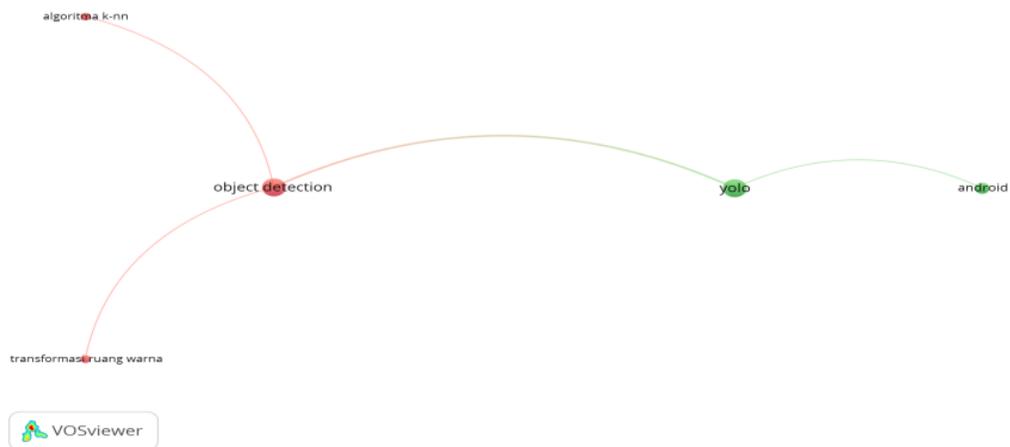
Bab ini berisi kesimpulan dari hasil penelitian yang telah dilaksanakan serta saran terkait aplikasi yang telah dibuat untuk dikembangkan menjadi sistem yang lebih baik.

DAFTAR PUSTAKA

BAB II LANDASAN TEORI

2.1 State Of The Art

State Of The Art merupakan kumpulan jurnal yang digunakan sebagai referensi dalam penelitian ini. *State Of The Art* turut memberikan penjabaran mengenai perbedaan antara penelitian terdahulu dan penelitian yang akan dilakukan. *State of The Art* ini diambil dari beberapa penelitian guna untuk panduan serta menjadi acuan perbandingan penelitian yang nantinya akan dilakukan. Seperti pada Gambar 2.2 di bawah ini, ditemukan 10 jurnal nasional



Gambar 2.2 State Of The Art

Pada gambar 2.2 merupakan penelitian – penelitian serupa dengan penelitian yang akan dilakukan, berikut merupakan hasil dari jurnal yang digunakan.

1. Penelitian dengan judul “Rancang Bangun Sistem Deteksi Kematangan Buah Kelapa Sawit Berdasarkan Deteksi Warna

Menggunakan Algoritma K-NN” pada penelitian ini peneliti menggunakan Algoritma K-NN dan juga menggunakan ekstraksi ciri berupa warna. dalam penelitian ini pengolahan data citra dilakukan menggunakan perangkat lunak MATLAB dan implementasi aplikasi menggunakan bahasa pemrograman *Python* dengan menggunakan *library OpenCV* dan *scikit-learn*. Proses pelatihan dan validasi algoritma K-NN ini juga dilakukan dengan menggunakan perangkat lunak *Python*. Pada penelitian ini didapatkan hasil akurasi paling tinggi sebesar 86% (Saputra and Oktaviyani 2023). kesamaan penelitian ini dengan penelitian yang akan datang yaitu sama-sama deteksi menggunakan deteksi warna pada objek sedangkan perbedaan penelitian ini dengan penelitian yang akan dilakukan yaitu berbeda dari jenis objeknya pada penelitian yang akan dilakukan peneliti menggunakan objek jenis tanah berdasarkan warna tidak hanya itu penelitian yang akan dilakukan juga menggunakan Algoritma *YOLOv8*.

2. Penelitian dengan judul “Sistem Deteksi Kualitas Buah Jambu Air Berdasarkan Warna Kulit Menggunakan Algoritma *Principal Component Analysis (Pca)* dan *K-Nearest Neighbor (K-NN)*” penelitian ini dilakukan deteksi kualitas buah jambu air berdasarkan warna pada kulitnya menggunakan dua algoritma yaitu PCA dan juga K-NN. Pada proses pengujian sistem pada penelitian ini peneliti menggunakan data

latih dan juga data uji dengan nilai akurasi paling tinggi yaitu mencapai 70% (Novianto and Sugihartono 2020). Pada penelitian ini juga aplikasi belum dapat membedakan buah jambu air dengan buah jambu air lainnya. Kesamaan antara penelitian ini dengan penelitian yang akan di lakukan yaitu ada pada deteksi objek menggunakan warna sedangkan perbedaan antara penelitian ini dengan penelitian yang akan dilakukan yaitu ada pada jenis objeknya dimana pada penelitian yang akan dilakukan yaitu menggunakan objek tanah dan juga pada penelitian yang akan dilakukan peneliti menggunakan algoritma *YOLOv8* untuk deteksi objek jenis tanah berdasarkan warna.

3. Penelitian dengan judul “Deteksi Gulma Berdasarkan Warna Hsv Dan Fitur Bentuk Menggunakan Jaringan Syaraf Tiruan” pada penelitian ini peneliti menggunakan metode jaringan syaraf tiruan dengan jumlah dataset yang dikumpulkan yaitu 149 data daun yang dimana data tersebut di bagi menjadi tiga bagian yaitu 70% sebagai *data training*, 15% *data validasi* dan 15% *data uji*, didapati akurasi pengujian sebesar 95.46%. dan cara pengambilan gambarnya menggunakan Kamera *RGB* (Fitriyah et al. 2021). Kesamaan dari penelitian ini dan penelitian yang akan dilakukan yaitu ada pada deteksi jenis objek berdasarkan warna sedangkan perbedaan penelitian ini dengan penelitian yang akan dilakukan terdapat pada perbedaan jenis objeknya dan juga metodenya.
4. Penelitian dengan judul “Deteksi Tingkat Kematangan Buah Pinang

Menggunakan Metode *Support Vector Machine* Berdasarkan Warna Dan Tekstur” pada penelitian ini peneliti menggunakan 90 dataset citra buah pinang dengan menggunakan metode *Support Vector Machine* (SVM) yang dimana hasil dari pengujian sistenya mendapatkan tiga nilai akurasi yaitu *Linear*, dengan tingkat *Accuracy* yang diperoleh sebesar 77,28%, *Gaussian* memperoleh hasil yang cukup baik dengan tingkat *Accuracy* yang sebesar 81,81%, dan terakhir *Polynomial* dengan hasil *Accuracy* yang diperoleh sebesar 90,90% (Maneno et al. 2023). persamaan antara penelitian ini dengan penelitian yang akan dilakukan yaitu berada pada deteksi objek berdasarkan warnanya sedangkan perbedaannya yaitu pada penelitian yang akan dilakukan menggunakan metode *YOLOv8* serta aplikasi akan diimplementasikan berbasis Android.

5. Penelitian dengan judul “Deteksi Kematangan Buah Manggis Berdasarkan Fitur Warna Citra Kulit Menggunakan Metode Transformasi Ruang Warna Hsv” pada penelitian ini peneliti menggunakan metode transformasi ruang warna HVS dengan menggunakan dataset sebanyak 60 data buah yang terdiri dari 45 sampel latih yang terdiri dari 15 citra buah manggis disetiap jenis kematangannya, dan 15 sampel uji yang terdiri dari 5 sampel buah manggis matang, 5 sampel buah manggis setengah matang dan 5 sampel buah manggis mentah dari hasil dataset 60 data didapatkan hasil akurasi paling tinggi yaitu sebesar 86.6% (Dalimunthe 2021).

persamaan antara penelitian ini dengan penelitian yang akan data yaitu berada pada deteksi objek menggunakan warna sedangkan perbedaan penelitian ini dengan penelitian yang akan dilakukan yaitu penelitian yang akan dilakukan menggunakan algoritma YOLOv8 dengan aplikasi yang di implementasikan berbasis android dan perbedaan pada objek yang di deteksi pada penelitian yang akan dilakukan peneliti menggunakan objek jenis tanah berdasarkan warna.

6. Penelitian dengan judul “Deteksi Kendaraan Secara *Real Time* Menggunakan Metode *YOLO Berbasis Android*” pada penelitian ini peneliti menggunakan metode yolo dengan menggunakan jumlah dataset sebanyak 200, 4 kelas, 10 batch, dan 200 epoch. Pada penelitian ini pelatihan dilakukan hingga 4.000 step. Pengujian yang dilakukan pada penelitian ini menggunakan smartphone xiaomi redmi 4X dengan resolusi video berukuran 768x432 piksel. Dengan hasil akurasi paling tinggi yaitu 83,3% (H, Matulatan, and Hayaty 2020). perbedaan pada penelitian ini dengan penelitian yang akan dilakukan yaitu ada pada objek deteksinya dan juga algoritma yolo pada penelitian sebelumnya menggunakan *YOLOv2* sedangkan penelitian yang akan di lakukan menggunakan *YOLOv8*.
7. Penelitian dengan judul “Deteksi Lampu Lalu Lintas Menggunakan *YOLO* untuk *Autonomous Car*”. Pada penelitian ini peneliti menggunakan metode *YOLO* untuk *Autonomous Car* dengan dataset yang dikumpulkan dari internet berupa gambar yang dibuat oleh

Morten Borno Jensen dan satu kolaborasi dengan *Mark Philip*, pada penelitian ini juga peneliti melakukan training data dan mendapat hasil training dengan jumlah akurasi paling tinggi yaitu 82,71% (Kusuma 2023). Perbedaan penelitian ini dengan penelitian yang akan dilakukan yaitu ada pada jenis objek yang di gunakan pada penelitian yang akan dilakukan peneliti menggunakan deteksi objek jenis tanah berdasarkan warna dan juga aplikasi ini di implementasikan berbasis android.

8. Penelitian dengan judul “Sistem Deteksi Pengendara Sepeda Motor Tanpa Helm Dan Kelebihan Penumpang Pada Dengan Menggunakan *Yolo V3*” pada penelitian ini peneliti menggunakan metode *YOLOv3* dengan jumlah dataset sebanyak 173 data yang nantinya akan diberi Pelabelan menggunakan *Labeling* yang hasil dari proses pelabelan tersebut adalah koordinat dalam bentuk *text*. Setelah itu peneliti melakukan training model setelah itu dilakukan proses pengujian. Nilai akurasi paling tinggi yang telah dilakukan dalam penelitian ini yaitu 84.6% dengan jumlah pengujian sebanyak 50 kali (Nasional, Riset, and No 2021). perbedaan yang terdapat pada penelitian ini dengan penelitian yang akan dilakukan yaitu dengan pada penelitian ini peneliti menggunakan *YOLOv3* sedangkan penelitian yang akan dilakukan menggunakan *YOLOv8* dan aplikasi ini diimplementasikan berbasis android.
9. Penelitian dengan judul “Deteksi Tanda Kehidupan Pada Korban Bencana Alam Dengan Algoritma *Yolo* Dan *Open Pose*” pada

penelitian ini peneliti menggunakan *algoritma yolo* dan *open pose*, *algoritma yolo* yang digunakan pada penelitian ini yaitu *Yolov4* yang merupakan pengembangan dari versi sebelumnya yaitu *yolov3*. Pada penelitian kali ini peneliti menggunakan dataset sebanyak 347 data dengan jumlah akurasi paling tinggi yaitu mencapai 100% (Aulia, Setianingsih, and Kallista 2021). perbedaan penelitian ini dengan penelitian yang akan dilakukan yaitu berada pada algoritma yang digunakan Algoritma yang digunakan dalam penelitian yang akan dilakukan yaitu menggunakan *YOLOv8* sedangkan algoritma yang digunakan dalam penelitian ini menggunakan *YOLOv4* dan perbedaan lainnya yaitu berbeda pada jenis objeknya. Objek yang digunakan dalam penelitian ini yaitu korban bencana alam sedangkan pada penelitian yang akan dilakukan yaitu menggunakan objek tanah.

- 10.** Penelitian dengan judul “Objek Deteksi Makanan Khas Palembang Menggunakan *Algoritma Yolo (You Only Look Once)*” pada penelitian ini peneliti menggunakan algoritma *Yolov3* dengan menggunakan dataset sebanyak 31 dataset gambar makanan khas Palembang dengan hasil akurasi rata – rata yaitu sebesar 96%. (Rahma et al. 2021) Perbedaan penelitian ini dengan penelitian yang akan dilakukan yaitu pada penelitian yang akan dilakukan menggunakan *YOLOv8* sedangkan penelitian yang ini menggunakan *YOLOv3* tidak hanya itu perbedaannya juga berada pada objek yang di deteksi pada penelitian ini peneliti menggunakan deteksi objek makanan khas Palembang

sedangkan pada penelitian yang akan dilakukan menggunakan deteksi objek jenis tanah berdasarkan warna.

Table 2.1 Perbandingan Penelitian

No	Fitur	Penelitian										
		PP	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	Menggunakan Deteksi objek Berdasarkan Warna	√	√	√	√	√	√		√			
2	Implementasi Berbasis android	√						√				
3	Implementasi berbasis web	√	√	√	√	√	√		√	√	√	√
3	Menggunakan algoritma Yolo	√						√	√	√	√	√
4	Hasil akurasi	√	√	√	√	√	√	√	√	√	√	√

keterangan :

1. PP : Penelitian Peneliti
2. P1-P5 : Penelitian jurnal terkait

Kolom yang terdapat checklist berarti fitur yang terdapat dalam penelitian maupun jurnal terkait.

2.3 Literatur Terkait

2.3.1 Tanah

Tanah merupakan kebutuhan hidup manusia yang sangat mendasar dan penting dalam kehidupan manusia, sehingga dalam melaksanakan aktivitas dan kegiatannya manusia selalu berhubungan dengan tanah. Begitu pentingnya tanah bagi kehidupan manusia, oleh karenanya tanah akan tetap merupakan tumpuan rakyat banyak guna melangsungkan kehidupan. (Rohaedi, H. Insan, and Zumaro 2019) Bentuk fisik tanah dapat ditentukan berbeda-beda antara satu tanah dengan tanah lainnya berdasarkan studi lapangan tentang *morfologi* tanah dan juga penentuan berdasarkan warna tanah yang ada. Tanah yang berbeda diberi nama berdasarkan ciri fisik dan juga warnanya, yang juga mewakili ciri utamanya (Ferdeanty et al. 2019).

2.3.2 Histosol (*Humus*)

Tanah Humus merupakan salah satu jenis tanah yang terbentuk dari pelapukan *bahan organik*. Akibat pelapukan bahan organik, jenis tanah ini produktif bagi hampir semua jenis tanaman. Tanah humus, terbentuk dari tumbuhan mati, disebabkan oleh pelapukan dan pembusukan *bahan organik*. Untuk pertanian, humus sangat bermanfaat. Kadar *bahan organik* yang signifikan membuat tanah menjadi hitam. Humus umumnya digunakan sebagai media tanam kakao, nanas, dan padi. Tersebar di Pulau Sumatera, Sulawesi, Jawa Barat, Kalimantan dan Papua. Tanah gambut merupakan tanah yang disebabkan oleh penguraian *bahan-bahan organik* (Rani et al.

2020).



Gambar 2.3 *Histosol/ Tamah Humus*

2.3.3 Aluvial

Menurut USDA, jenis tanah aluvial termasuk dalam hierarki Inceptisols. Sifat umumnya identik dengan tanah latosol. Aluvial merupakan tanah muda hasil pengendapan material halus dari pergerakan sungai. Ciri utama tanah aluvial adalah warna abu-abu dengan struktur agak gembur. Kesuburan aluvial sangat bergantung pada sumber material sungai. Hampir di seluruh wilayah Indonesia, jenis tanah aluvial banyak ditemukan di sungai-sungai besar seperti Jawa, Sumatera, Kalimantan, dan Papua. Aluvial

umumnya dimanfaatkan untuk tanaman padi, palawija, tebu, kokon, tembakau dan buah-buahan (Rani et al. 2020).



Gambar 2.4 *Tanah Alluvial*

2.3.4 Latosol

Latosol atau *insepticol* mempunyai lapisan solum (bagian atau struktur yang telah mengalami pelapukan) dengan ketebalan sedang hingga tinggi. Tebalnya dapat mencapai 130 cm hingga 5 meter atau lebih tebal lagi. Latosol umumnya berwarna merah, serta mempunyai tekstur liat. Konsistensinya gembur dengan struktur berupa remah, *pH-nya* berkisar antara 4,5 hingga 6,5 (kategori asam-agak asam). Tanah latosol merupakan jenis dengan kandungan mineral tertentu, cukup baik untuk ditanami berbagai

tanaman. Contohnya seperti tanaman Kakao, Tembakau, Vanili, Pala, dan Tebu. (UMSU 2022)



Gambar 2.5 Tanah Latosol (Darwadi 2020)

2.2.5 Deep Learning

Deep Learning Merupakan metode pembelajaran yang menggunakan jaringan saraf tiruan berlapis-lapis (*multi layer*), Jaringan Syaraf Tiruan ini dibuat mirip dengan otak manusia, dimana *neuron-neuronnya* saling terhubung membentuk *jaringan neuron* yang sangat kompleks (Raup et al. 2022). *Deep learning* juga merupakan bidang pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk mengimplementasikan masalah dengan kumpulan data besar. *Teknik deep learning* memberikan arsitektur yang sangat kuat untuk pembelajaran yang diawasi. Dengan menambahkan

lebih banyak lapisan, model pembelajaran dapat merepresentasikan data gambar berlabel dengan lebih baik. Dalam machine learning terdapat teknik penggunaan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasikan gambar dan mengenali suara. Namun cara ini masih mempunyai beberapa kekurangan baik dari segi kecepatan maupun akurasi (D Fermannah 2019).

2.3.6 Python

Python adalah bahasa pemrograman berbasis objek yang dapat berinteraksi secara interaktif. Bahasa ini memiliki struktur data tingkat tinggi. *Python* adalah bahasa pemrograman interpretatif yang memiliki banyak fungsi, dan dirancang dengan fokus pada kejelasan dan kemudahan pemahaman kode. *Python* dianggap sebagai bahasa yang menggabungkan kemampuan dan kejelasan sintaksis kode. Bahasa pemrograman *Python* dirancang khusus untuk memudahkan programmer dalam membuat program dengan efisiensi waktu, kemudahan pengembangan, dan kompatibilitas dengan sistem. *Python* dapat digunakan untuk membuat aplikasi mandiri atau pemrograman skrip (Triono, Budi, and Abdillah 2023).

2.3.7 Object Detection

Deteksi objek atau biasa disebut *Object Detection* merupakan suatu proses yang digunakan untuk mengetahui keberadaan suatu objek tertentu pada suatu citra digital. Proses pendeteksiannya dapat dilakukan dengan berbagai metode yang pada dasarnya melakukan pembacaan ciri-ciri seluruh

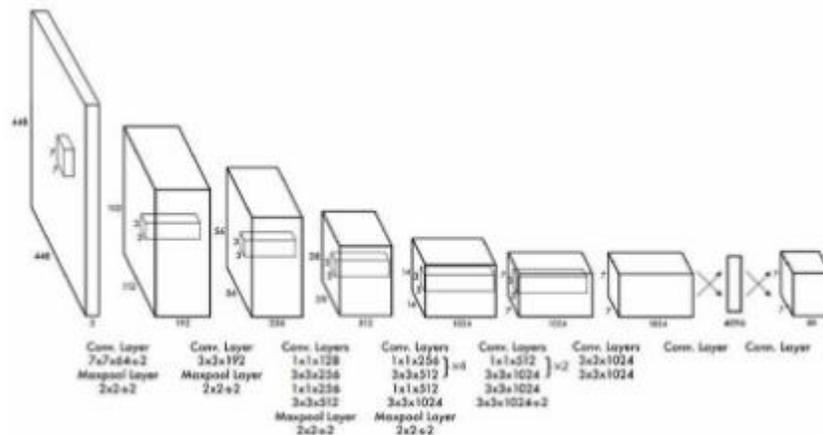
objek pada citra masukan. Fitur objek pada gambar akan dibandingkan dengan fitur dari objek referensi atau template kemudian dibandingkan dan ditentukan apakah objek yang terdeteksi termasuk objek yang ingin dideteksi atau Tidak (Rizkatama, Nugroho, and Suni 2021).

2.3.8 You Only Look Once

You Only Look Once (YOLO) adalah algoritma yang dikembangkan untuk mendeteksi suatu objek secara real time. Sistem pendeteksian dilakukan dengan menggunakan *reuse classifier* atau *localizer* untuk melakukan pendeteksian. Model diterapkan pada gambar di beberapa lokasi dan skala. Area dengan gambar dengan skor tertinggi akan dianggap sebagai deteksi. Sebelum dilakukan proses pelatihan perlu dilakukan proses anotasi terlebih dahulu untuk membentuk dataset. Untuk setiap data memiliki nama kelas, titik koordinat X objek, titik koordinat Y objek, panjang kotak pembatas, dan lebar kotak pembatas (Geraldly and Lubis 2020).

YOLO menggunakan pendekatan *jaringan saraf tiruan (JST)* untuk mendeteksi objek dalam gambar. Jaringan ini membagi gambar menjadi beberapa bagian dan memprediksi setiap kotak pembatas serta *probabilitas* untuk setiap wilayah. Kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi. *YOLO* mempunyai beberapa keunggulan dibandingkan *sistem berorientasi classifier*, terlihat dari seluruh gambar pada saat dilakukan pengujian dengan prediksi informasi global pada gambar tersebut (Geraldly and Lubis 2020).

YOLO mengimplementasikan arsitektur serupa *Convolutional neural networks*. *YOLO* hanya menggunakan lapisan *konvolusional* dan lapisan penggabungan. Untuk *convolutional layer* terakhir disesuaikan dengan jumlah kelas dan jumlah kotak prediksi yang diinginkan (Geraldly and Lubis 2020).



Gambar 2.6 Arsitektur Yolo

Arsitektur YOLO sebenarnya cukup sederhana. Pada awalnya, sistem menerima citra input dengan ukuran (448, 448, 3), yang berarti citra memiliki dimensi 448 x 448 dengan 3 *channel* warna. Kemudian, citra ini melewati satu proses *convolutional network* yang akan menghasilkan *output* dengan dimensi (7, 7, 30). Di sini, 7 x 7 mewakili ukuran *grid cell* ($S = 7$), dan 30 merupakan jumlah nilai yang dihasilkan dari perkalian antara jumlah kotak pembatas B dan jumlah komponen dalam satu kotak B , ditambah jumlah kelas ($B \times 5 + C$, dengan $B=2$ dan $C=20$) (Geraldly and Lubis 2020).

2.3.9 Google Colaboratory

Google colaboratory atau biasa juga disebut *Google Colab* merupakan produk dari *Google* yang dapat digunakan untuk menulis dan mengedit program. Produk *Google Colab* ini biasanya digunakan oleh profesi yang pekerjaannya memerlukan lingkungan *coding*, seperti *programmer*, *developer*, atau *software engineer*. Sedangkan *Python* merupakan bahasa pemrograman yang biasa digunakan untuk membangun situs, aplikasi, atau mengotomatisasi tugas dan melakukan analisis data. Jika digabungkan, *Google Colab Python* adalah lingkungan *Python* yang memiliki format *notebook* untuk menulis, menyimpan, dan berbagi kode program secara online (Digitalskola 2023).

2.3.10 Confusion Matrix

Confusion Matrix merupakan sebuah tabel yang menggambarkan jumlah data uji yang diklasifikasikan secara benar dan salah. Tabel ini berguna bagi pengembang dan analisis untuk menilai kinerja model klasifikasi dan memberikan pemahaman tentang tingkat akurasi serta jenis kesalahan yang terjadi dalam proses klasifikasi data (Normawati and Prayogi 2021) Pada Tabel 2.2 di bawah dapat dilihat tabel kriteria klasifikasi menggunakan confusion matrix.

Tabel 2.2 confusion matrix

Kelas	Disklasifikasi Sebagai Positif	Disklasifikasi Sebagai Negatif
	+	True Positive (TP)
-	True Negative (TN)	False Negative (FN)

Tabel 1 merupakan tabel dari konsep klasifikasi *confusion matrix*, *True Positive (TP)* terjadi ketika model klasifikasi dengan tepat mengidentifikasi jumlah kasus positif. *True Negative (TN)* adalah ketika model dengan benar mengidentifikasi jumlah kasus negatif. *False Positive (FP)* terjadi ketika model salah mengidentifikasi kasus negatif sebagai positif. *False Negative (FN)* adalah ketika model salah mengidentifikasi kasus positif sebagai negatif. Akurasi merupakan ukuran kinerja model klasifikasi yang menunjukkan seberapa banyak data yang diprediksi dengan benar dari total data, diukur dalam persentase (Purwono et al. 2024). Berikut merupakan rumus untuk menghitung nilai akurasi.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Pengukuran kinerja model mencakup *precision*, *recall*, dan *F1-Score*. *Precision* merupakan ukuran keakuratan positif, yang merupakan perbandingan antara prediksi positif yang benar dan total prediksi positif yang dilakukan oleh model. Berikut merupakan rumus untuk menghitung nilai

precision

$$Recall = \frac{TP}{TP + FP}$$

Recall mengukur seberapa baik model dapat mengidentifikasi data yang sebenarnya berlabel positif. Hal ini menunjukkan seberapa baik model dapat menangkap atau mengingat data yang positif secara efektif. Rumus untuk menghitung nilai recall dapat dilihat di bawah ini.

$$Precision = \frac{TP}{TP + FN}$$

F1-Score adalah ukuran yang menggabungkan hasil *presisi dan recall* melalui rata-rata harmonik (jenis perhitungan nilai tengah), sehingga memberikan tinjauan yang seimbang dari kedua metrik tersebut.

$$F1 - Score = 2 \frac{precision * Recall}{Precision + Recall}$$

2.3.11 ONNX (Open Neural Network Exchange)

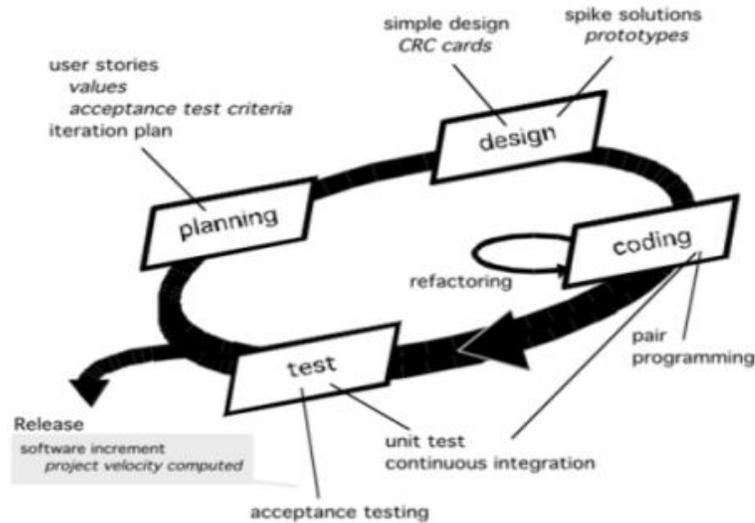
Open Neural Network Exchange (ONNX) adalah format yang dirancang untuk merepresentasikan model pembelajaran mendalam, sehingga model tersebut dapat dijalankan di berbagai perangkat tanpa khawatir tentang implementasinya. Hal ini membantu para peneliti, pengembang, dan insinyur untuk fokus pada masalah inti tanpa terjebak dalam permasalahan seputar infrastruktur, seperti kerangka kerja yang digunakan atau ketersediaan perangkat keras khusus. *ONNX* dapat digunakan di berbagai perangkat, mulai dari perangkat seluler hingga server besar serta didukung oleh beragam

runtime dan alat. Dengan *ONNX*, gesekan dalam memindahkan model pembelajaran mesin antar alat dan kerangka kerja dapat diminimalkan. Sebagai contoh, *ONNX* sangat berguna ketika ingin menerapkan model pembelajaran mendalam dalam aplikasi *iOS*. Model yang dibuat dalam berbagai kerangka kerja seperti *TensorFlow*, *PyTorch*, atau *MXNet* dapat dengan mudah dikonversi menjadi format *ONNX*. Kemudian, *ONNX* memungkinkan ekspresi model tersebut ke dalam format *CoreML* yang digunakan dalam aplikasi *iOS*, tanpa perlu melakukan porting model secara manual atau melatih ulang model dari awal. Ini membuat proses penerapan model lebih sederhana, efisien, dan praktis, serta menghemat waktu dan energi pengembang. Dengan demikian, *ONNX* membantu dalam mengatasi masalah kompatibilitas dan transfer model pembelajaran mesin antar kerangka kerja dan *platform* (Shridhar, Tomson, and Innes 2020).

2.3.12 *Extreme programming*

Extreme Programming (XP) merupakan proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil. Media dan metode ini juga cocok jika tim dihadapkan pada persyaratan yang tidak jelas atau terjadi perubahan persyaratan yang sangat cepat. Sedangkan menurut Ferdiana, *Extreme Programming (XP)* dikenal dengan istilah metode atau “cara teknis”. tim teknis mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik praktis pengembangan perangkat lunak.

XP adalah dasar bagaimana tim bekerja setiap hari (Sahputra, Defriani, and Hermanto 2023).



Gambar 2.7 *Extreme programming (Sahputra et al. 2023)*

2.3.13 *Android studio*

Android Studio adalah *Integrated Development Environment (IDE)* untuk pengembangan aplikasi *Android*, berdasarkan *IntelliJ IDEA*. Selain menjadi editor kode dan alat pengembang *IntelliJ* yang tangguh, *Android Studio* menawarkan lebih banyak fitur. Ini digunakan untuk meningkatkan produktivitas saat membuat aplikasi *Android*, *android studio* juga menggunakan *build system*, *emulator*, *template code*, dan *integrasi ke Github berbasis Gradle* (Wibowo 2019).

2.3.14 *Android*

Android adalah sistem operasi untuk ponsel pintar dan komputer tablet

berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi pengembang untuk membuat aplikasi sendiri untuk berbagai kegunaan perangkat seluler. Awalnya *Google Inc* membeli *Android Inc*, pendatang baru yang membuat *software* untuk ponsel. Kemudian seiring perkembangan *Android*, dibentuklah *Open Handset Alliance*, sebuah *konsorsium* yang terdiri dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Qualcomm, Motorola, Nvidia* dan *T-Mobile* (Widodo, Widayanto, and Sardianto 2021).

2.3.15 PyTorch

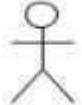
Pytorch adalah kerangka pembelajaran mesin *open source* berdasarkan perpustakaan *Torch* yang digunakan untuk aplikasi di bidang *visi komputer*, pemrosesan bahasa alami. *Pytorch* dikembangkan oleh *Meta AI*, sebuah divisi dari perusahaan *Meta* (yang disebut *Facebook*) yang merupakan perusahaan teknologi asal Amerika Serikat yang dapat digunakan dalam berbagai bahasa pemrograman, seperti *Python dan C++* (Helnawan, Attamimi, and Irfansyah 2023).

2.3.16 Use case diagram

Diagram use case merupakan salah satu jenis diagram *UML (Unified Modeling Language)* yang menggambarkan fungsi, ruang lingkup, dan interaksi pengguna dengan sistem. *Diagram use case* memvisualisasikan interaksi antara pengguna (aktor) dan sistem (*use case*), serta tindakan apa yang dapat dilakukan aktor terhadap *use case* secara detail. Seperti gambar

dibawah ini yaitu merupakan gambar dari simbol-simbol *Use Case* (Muhammad 2023).

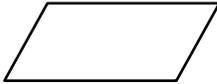
Tabel 2.3. Simbol – simbol *Use Case*

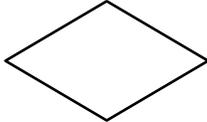
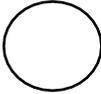
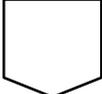
Simbol	Deskripsi
<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi / <i>extend</i></p> <p><<<i>extend</i>>></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.</p>
<p>Generalisasi</p> 	<p>Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> yang mana fungsi satu lebih umum dari yang lainnya.</p>
<p>Menggunakan / <i>include</i> / <i>use</i></p> <p><<<i>include</i>>> uses</p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya</p>

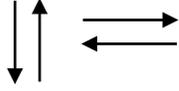
2.3.17 Flowchart

Flowchart merupakan gambaran mengenai langkah-langkah yang akan dilakukan dalam suatu sistem. *Flowchart* berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis. Seperti gambar dibawah ini merupakan gambar untuk simbol-simbol *flowchart* (Setiawan 2021b).

Tabel 2.4. simbol – simbol *Flowchart*

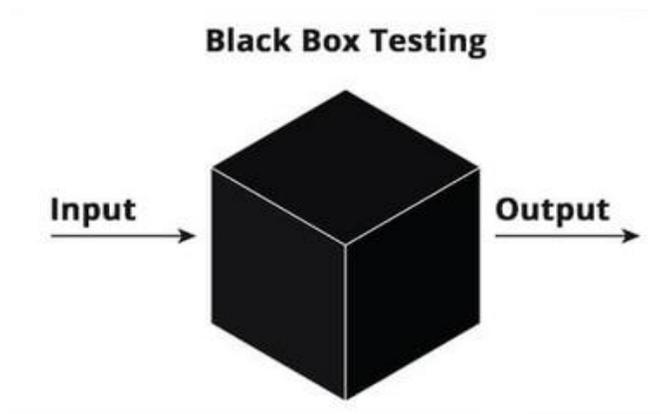
No	Simbol	Nama	Fungsi
1.		Terminal	Awal atau akhir suatu program.
2.		<i>Input/Output</i>	Proses <i>input</i> atau <i>output</i> terlepas dari jenis perangkat.
3.		<i>Process</i>	Proses operasional komputer.

No	Simbol	Nama	Fungsi
4.		<i>Decision</i>	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan ya/tidak.
5.		<i>Connector</i>	Koneksi penghubung proses ke proses pada halaman yang sama.
6.		<i>Offline Connector</i>	Koneksi penghubung dari satu proses ke proses lain di halaman lain.
7.		<i>Predefined Process</i>	Melakukan atau melaksanakan suatu bagian prosedur yang disebut dengan sub bagian.
8.		<i>Punched Card</i>	<i>Input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
9.		<i>Punch Tape</i>	<i>Input</i> atau <i>output</i> yang menggunakan pita kertas berlubang.
10.		<i>Document</i>	Mencetak <i>output</i> dalam format dokumen.

No	Simbol	Nama	Fungsi
11.		<i>Flow</i>	Menyatakan jalannya arus suatu proses.

2.3.18 Black Box Testing

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh *user* (Arifita Arwaz et al. 2019).



Gambar 2.8 *Black Box Testing* (Setiawan 2021a).

2.3.19 Usability Testing

Usability testing adalah metode evaluasi yang digunakan dalam desain pengalaman pengguna (*user experience/UX*) dan pengembangan perangkat lunak untuk menguji usability suatu produk dengan melibatkan pengguna sungguhan. Tujuan utama dari usability testing adalah untuk mengidentifikasi masalah atau kesulitan yang mungkin dihadapi pengguna saat berinteraksi dengan produk tersebut, serta untuk mengumpulkan umpan balik yang dapat digunakan untuk meningkatkan desain dan usability produk tersebut (Anendya 2022). *Usability testing* terdiri dari lima kriteria, yaitu : *learnability, efficiency, memorability, errors* dan *satisfaction*.

1. *Learnability*

Kemudahan bagi pengguna untuk belajar dan menguasai penggunaan produk atau sistem baru. Semakin mudah suatu produk dipelajari oleh pengguna baru, semakin tinggi tingkat *learnability*-nya.

2. *Efficiency*

Kemampuan suatu produk atau sistem untuk digunakan secara efisien oleh pengguna dalam mencapai tujuan tertentu. Produk yang efisien memungkinkan pengguna untuk menyelesaikan tugas dengan cepat dan tanpa menggunakan banyak sumber daya.

3. *Memorability*

Kemampuan bagi pengguna untuk mengingat cara

menggunakan suatu produk atau sistem setelah periode waktu tertentu tanpa harus belajar ulang. Produk yang memiliki memorability yang baik akan lebih mudah digunakan kembali oleh pengguna setelah absen dalam penggunaannya.

4. *Errors*

Frekuensi dan tingkat kesalahan yang dibuat oleh pengguna saat menggunakan suatu produk atau sistem. Produk yang memiliki kesalahan yang rendah akan memberikan pengalaman pengguna yang lebih baik dan meminimalkan risiko kebingungan atau frustrasi.

5. *Satisfaction*

Tingkat kepuasan pengguna setelah menggunakan suatu produk atau sistem. Kepuasan pengguna dipengaruhi oleh berbagai faktor, termasuk kualitas produk, kemudahan penggunaan, dan sejauh mana produk memenuhi kebutuhan dan harapan pengguna. Produk yang menyebabkan tingkat kepuasan yang tinggi cenderung mendapatkan *loyalitas* dari pengguna dan rekomendasi positif.

Berikut ini merupakan rumus dari *usability testing*:

$$Y = \frac{\sum X}{N} \times 100 \quad (1) \text{ Skor Ideal}$$

Keterangan :

Y = Nilai persentase yang dicari (..%)

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi ($\sum = N \times R$)

N = Nilai dari setiap jawaban.

R = Frekuensi.

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel.

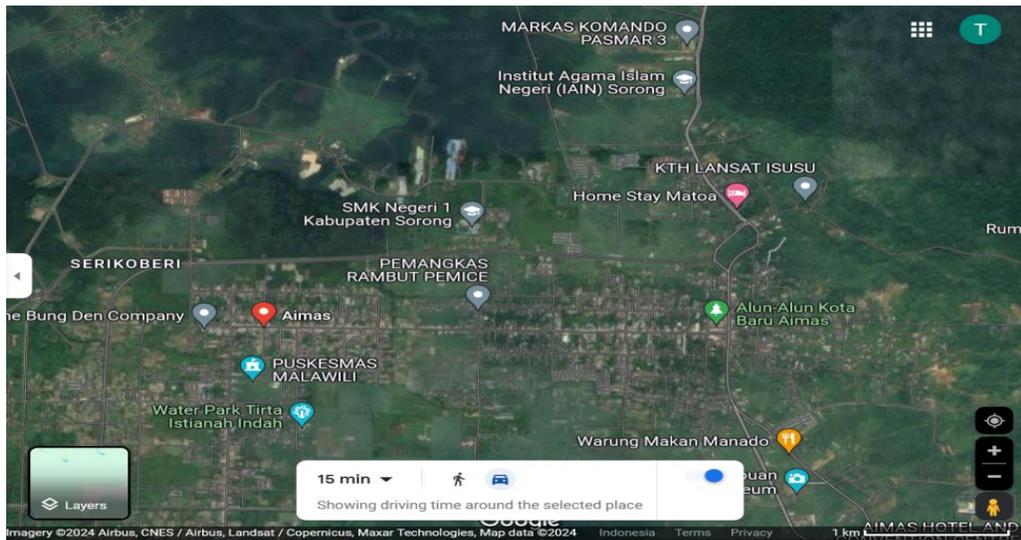
BAB III

METODE PENELITIAN

3.1 Lokasi dan Waktu Penelitian

3.1.1 Lokasi Penelitian

Lokasi yang dijadikan tempat penelitian yaitu kecamatan aimas, dimana kecamatan aimas berada di kabupaten sorong papua barat daya.



Gambar 3.1 *Lokasi Penelitian*

3.1.1 Waktu Penelitian

Waktu penelitian yang akan dilakukan yaitu pada bulan maret sampai bulan mei 2024.

Tabel 3.1 Waktu Pelaksanaan Penelitian

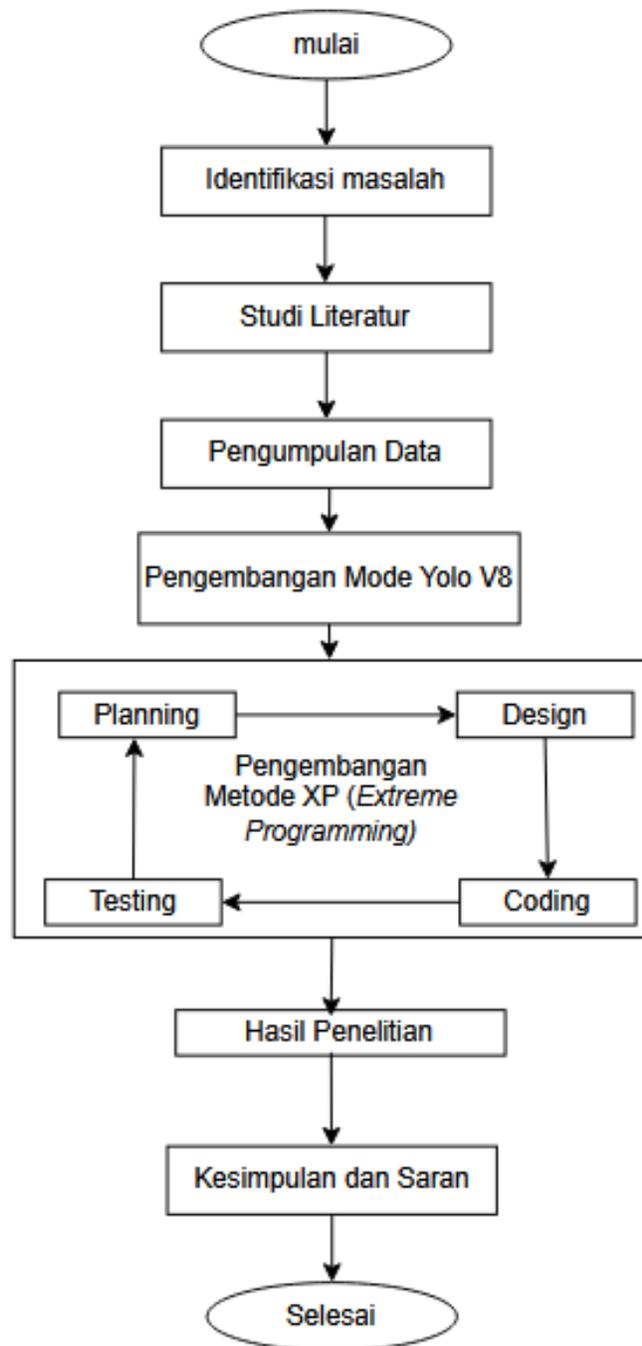
No	Aktifitas	Waktu																			
		Februari				Maret				April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengajuan Judul																				
2.	Penyusunan Proposal																				
3.	Pengajuan Proposal																				
4.	Pengumpulan Data																				
5.	Pembuatan Sistem																				
6.	Analisa dan Pembahasan																				
7.	Implementasi dan Pengujian																				
8.	Seminar Hasil																				
9.	Seminar Tutup																				

3.2 Tahapan Penelitian

3.2.1 Alur Penelitian

dalam melakukan ini, untuk mempermudahnya maka dijabarkan langkah-langkah apa saja yang akan dilakukan pada penelitian ini. Untuk alur

penelitian nya dapat dilihat dari gambar dibawah ini:



gambar 3.2 Alur Penelitian

3.3 Metode Pengumpulan Data

Berikut ini adalah metode pengumpulan data yang digunakan dalam penelitian ini :

a. *Study Literature*

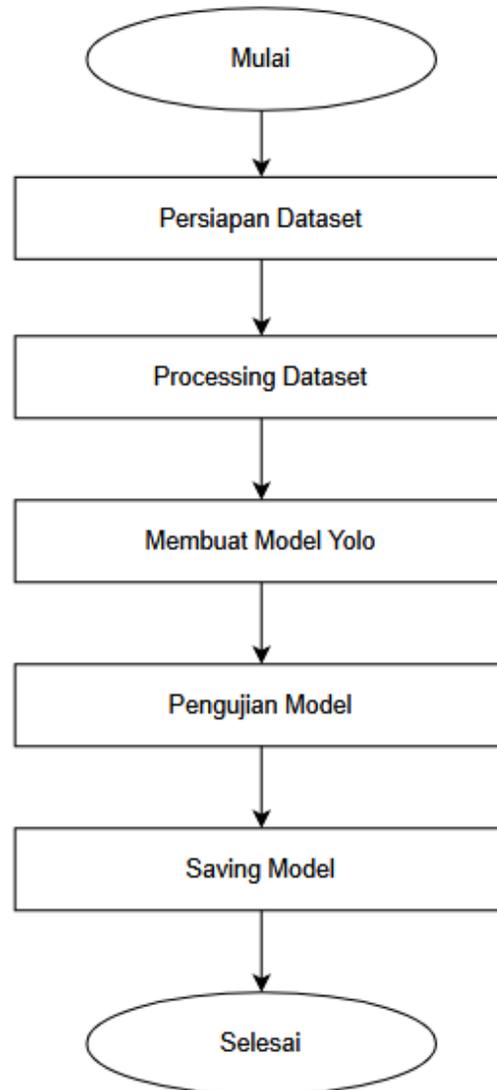
Study literature atau *study* kepustakaan dilakukan dengan mengumpulkan data melalui teks tertulis ataupun melalui edisi *softcopy*, seperti jurnal, *ebook* dan diperoleh dari *internet*.

b. *Observasi*

Observasi dilakukan dengan mengamati secara langsung pada tempat penelitian untuk mengumpulkan data yang mau dipakai.

3.4 Pengembangan Model *You Only Look Once*

Pada tahap ini peneliti akan membangun model *You Only Look Once* untuk membuat *implementasi deep learning* dalam identifikasi *jenis tanah pada kabupaten sorong* sesuai dengan perancangan sistem sebelumnya. Model ini dibuat dengan menggunakan algoritma *You Only Look Once Versi 8*, Bahasa *python* sebagai bahasa pemrograman yang dipakai, *roboflow* sebagai *library image data generator* untuk menganotasi data dan *google colaboratory* sebagai *notebook* untuk menjalankan program.



Gambar 3.3 *Flowchart You Only Look Once*

Berdasarkan alur dari Model pada gambar 3.3 dapat dijelaskan sebagai berikut :

A. Dataset

Pada penelitian ini peneliti menggunakan 3 *class* jenis tanah yang di kumpulkan dengan cara mencari data dengan cara turun lapangan dan mengambil gambar secara langsung. Kemudian Dataset akan diunggah ke

Google drive yang berfungsi sebagai penyimpanan pada *google collaborator*.

A. *Processing Data*

Pada tahap ini peneliti melakukan *processing* data seperti mengubah ukuran seperti yang diinginkan dan mengubah rotasi gambar. *Processing* data ini menggunakan *library image data generator* yang ada pada *roboflow*.

B. *Membuat Model Yolo*

Pada tahap ini peneliti akan membangun model untuk deteksi gambar jenis tanah menggunakan *You Only Look Once*. Peneliti juga akan mencoba menggunakan arsitektur model seperti *Tensorflow / tf lite* pada sistem ini.

C. *Pengujian Model*

Pada tahap ini peneliti akan menguji model yang telah dibuat dengan cara membuat *script* yang dapat membantu untuk melihat apakah prediksi yang dilakukan sudah benar atau tidak. Dan data yang digunakan untuk pengujian merupakan data yang berbeda dari dataset latih dan validasi.

D. *Saving model*

Setelah model telah dilatih dan dievaluasi, hasil penyimpanan dari model ini akan berbentuk *model.pt* kemudian data pelatihan di *convert* menjadi *model.tflite* agar memudahkan dalam pemodelan aplikasi pada android. Setelah di *convert* lalu dilakukan

pemodelan aplikasi menggunakan *android studio*.

3.5 Pengembangan Sistem Extreme Programming

Pada tahap ini peneliti melakukan pengembangan sistem. Pengembangan sistem ini adalah sebagai berikut :

A. *Planing*

Kebutuhan fungsional adalah dari rumusan masalah yang telah didapatkan selanjutnya dilakukan analisa terhadap kebutuhan fungsional dan *non fungsional* dari sistem yang nantinya akan dikembangkan. Kebutuhan fungsional dibutuhkan untuk mengetahui proses apa saja yang dapat dilakukan oleh sistem, serta siapa saja yang dapat menggunakan sistem yang dibangun. Adapun penjelasannya sebagai berikut :

1. Analisan Kebutuhan Fungsional

Fungsi-fungsi yang harus dipenuhi oleh aplikasi yang dirancang, diantaranya yaitu :

- a.* User berasal dari mahasiswa jurusan pertanian dan dapat membuka aplikasi deteksi jenis tanah.
- b.* Sistem dapat menampilkan hasil prediksi dan hasil akurasi pada user.

2. Analisa Kebutuhan *Non-Fungsional*

Kebutuhan *non fungsional* ini dibagi menjadi 2 (dua) bagian yaitu : Kebutuhan Perangkat Lunak (*Software*)

dan Kebutuhan Perangkat Keras (*Hardware*).

a. Kebutuhan *Software* Perangkat lunak yang digunakan antara lain :

1. Sistem operasi *Windows 11, 64-bit*
2. *Visual Studio Code* Sebagai *text editor*
3. *Google Chrome* sebagai *browser*
4. *google colaboratory* sebagai *notebook* untuk menjalankan program.

b. Kebutuhan *Hardware*

Perangkat keras digunakan yang digunakan yaitu Acer Aspire 5 dengan *Spesifikasi* sebagai berikut:

1. *Processor intel core i3-1115G4, 14 FDH Acer Comfyview, 8GB, DDR4*
2. *Ram 8.0 GB*

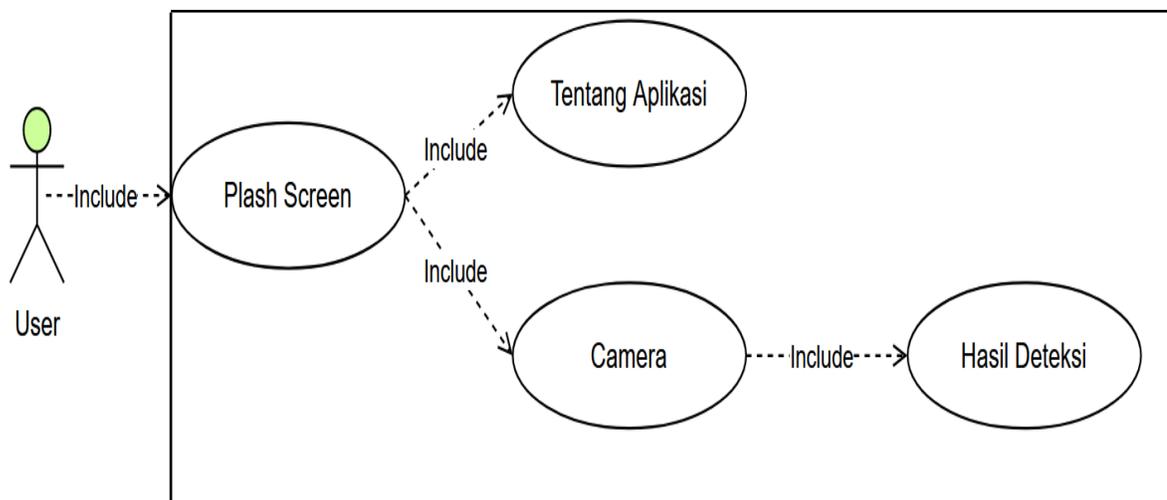
B. Design

1. Perancangan Analisis Desain

Pada tahap perancangan ini akan dibuat *visualisasi flowchart diagram* dan *use case diagram*, bertujuan untuk membantu menjelaskan gambaran dari alur penelitian yang dilakukan. Berikut adalah gambar dari sistem dalam bentuk *use case* dan *flowchart* :

2. Use Case Diagram Sistem

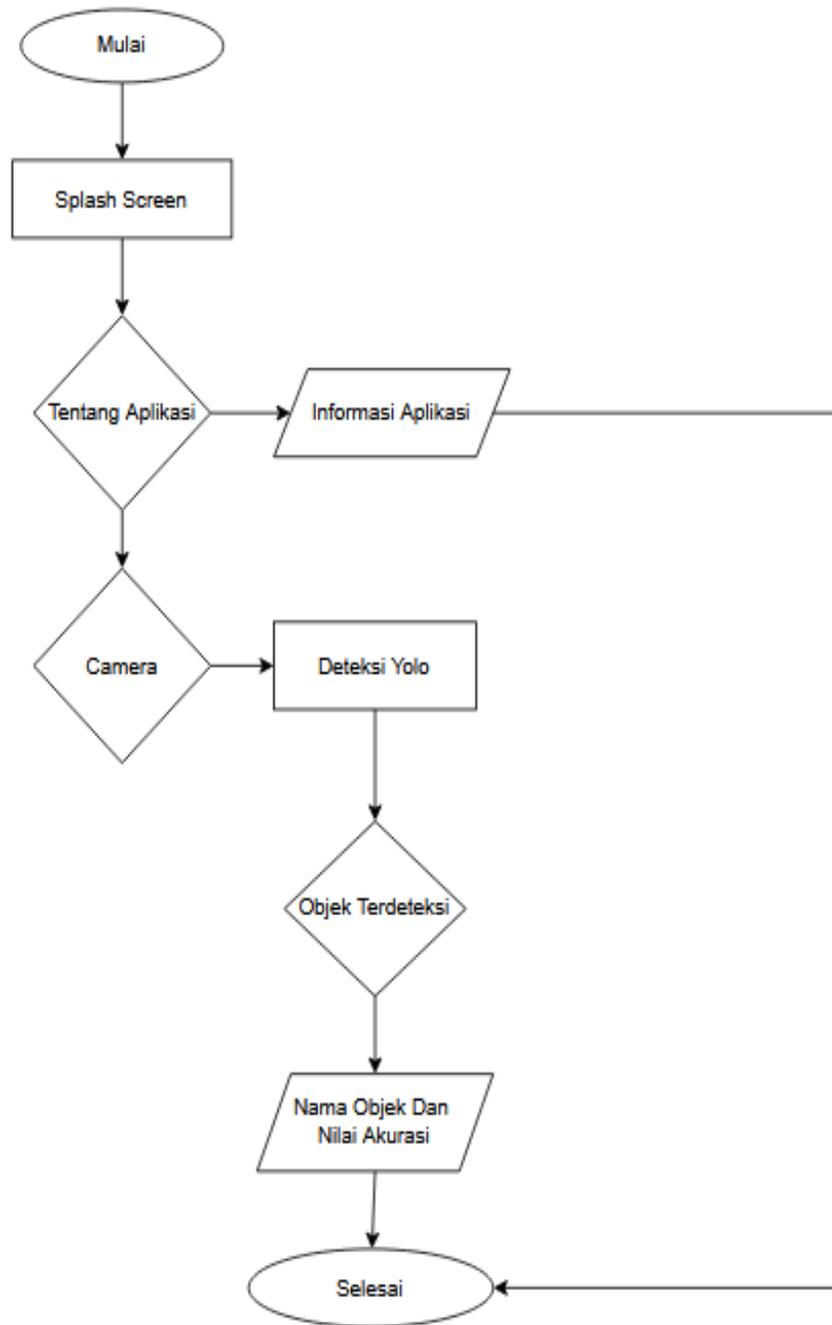
Pada gambar dibawa dapat dilihat bahwa *user* dapat melakukan penginputan data yang berupa gambar penyakit kulit wajah, selanjutnya akan di proses oleh model yang telah dibuat sebelumnya dan akan menghasilkan hasil prediksi dan nilai akurasi dari prediksi.



Gambar 3.4 Usecase Diagram

3. Flowchar Sistem

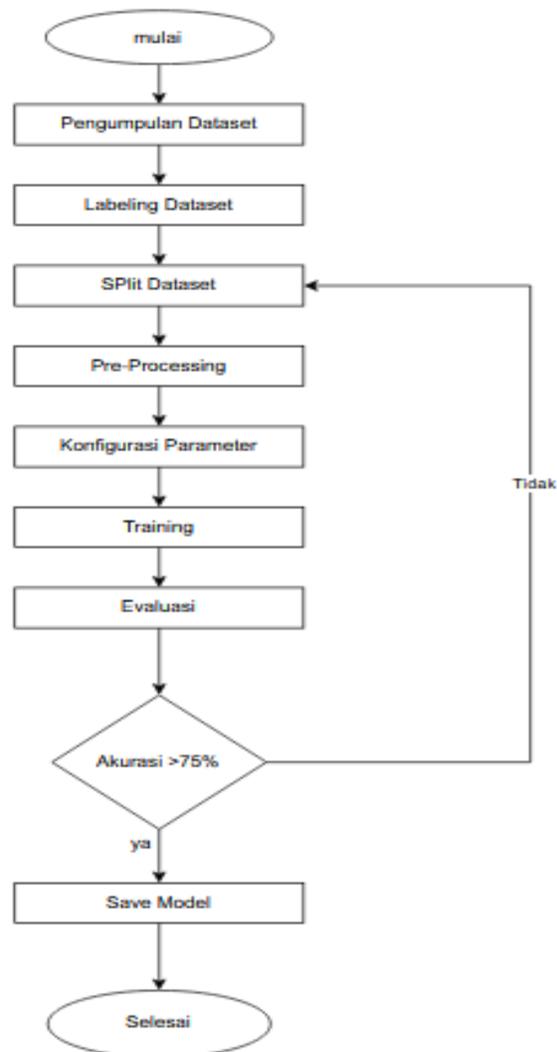
Pada flowchart ini dapat dilihat bahwa terdapat 1 proses yaitu proses pengulangan, proses pengulangan ini terjadi apabila gambar yang di inputkan oleh *user* tidak sesuai dengan ketentuan sistem dalam hal ini camera dapat diarahkan ulang pada objek.



Gambar 3.5 *Flowchart* Sistem Identifikasi

4. Flowchart Training

Flowchart training atau *flowchart* pelatihan digunakan untuk menggambarkan proses dalam melakukan pelatihan sebuah data. Pada gambar 3.6 dibawah ini merupakan *flowchart training* dalam penelitian ini :



Gambar 3.6 *Flowchart Training*

5. *Desain Tampilan Sistem*

Berikut adalah rancangan tampilan sistem yang akan dibuat :

A. **Tampilan plash**

pada gambar 3.7 di bawah ini merupakan tampilan awal pada saat membuka aplikasi yang terdiri beberapa gambar tanah kemudian digabung menjadi logo dari aplikasi ini, kemudian pada bagian lingkaran tanah terndapat keterangan “Deteksi Jenis Tanah” .



gambar 3.7 Tampilan Splash Screen

B. Tampilan Tentang Aplikasi

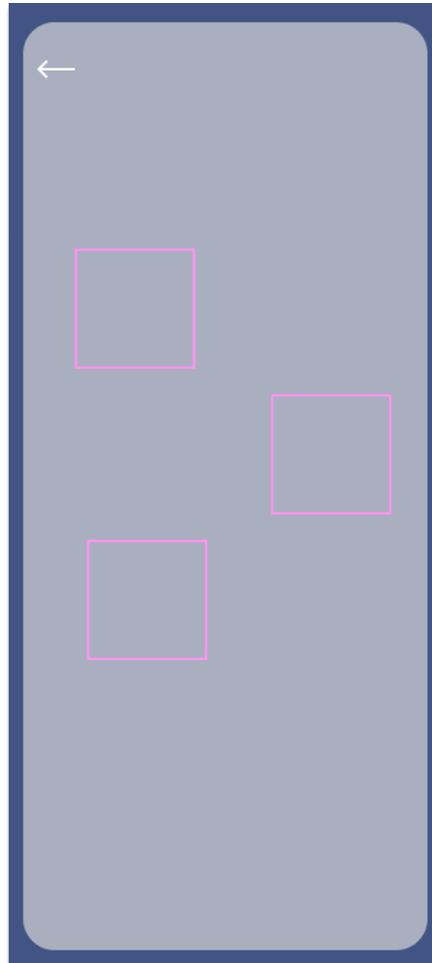
pada gambar 3.8 di bawah ini merupakan tampilan tentang aplikasi yang dimana pada tampilan ini akan menampilkan informasi terkait aplikasi ini



Gambar 3.8 *Tentang Aplikasi*

C. Tampilan Camera

Pada gambar 3.9 di bawah merupakan tampilan camera untuk mendeteksi jenis tanah, selain itu terdapat juga tanda panah untuk kembali ke menu utama.



Gambar 3.9 *Tampilan Camera*

6. *Pembuatan Coding*

Pada tahap ini peneliti membangun aplikasi sesuai dengan perancangan sistem sebelumnya. Aplikasi ini dibuat dengan

menggunakan model *You Only Look Once (Yolov8)* yang telah dirancang sebelumnya. Untuk membuat tampilan *interface* peneliti menggunakan bantuan library *flask python*. Untuk menuliskan *script coding* ini peneliti menggunakan *text editor visual studio* dan untuk pengembangan menggunakan metode *extreme programing* yang sering digunakan untuk pengembangan aplikasi pada skala kecil.

7. Pengujian *Black Box Testing*

Pengujian sistem adalah langkah yang penting dilakukan sebelum sistem dapat *release* untuk digunakan oleh *user*. Pada tahap ini dilakukan evaluasi terhadap sistem, apakah sistem yang dibuat sudah layak digunakan atau masih memiliki masalah, dalam hal ini peneliti akan melakukan pengujian menggunakan *Black Box Testing* untuk mengetahui apakah fitur telah sesuai

Tabel 3.2 Langkah – Langkah Pengujian sistem

Uji Kasus	Skenario Uji Coba	Hasil Yang Didapatkan
Menjalankan sistem	Menjalankan dan mengakses aplikasi jenis tanah	Apakah berhasil di akses dengan lancar.
Menekan tombol camera	Menyalakan camera dengan cara mengklik fitur camera	Aplikasi dapat mengakses camera sebagai input data yang ingin dideteksi

Uji Kasus	Skenario Uji Coba	Hasil Yang Didapatkan
Mengenali objek	Mendeteksi jenis tanah	Aplikasi dapat melakukan deteksi jenis tanah
Output	Menampilkan hasil deteksi jenis tanah	Sistem dapat menampilkan hasil berupa bounding box sesuai dengan objek yang dideteksi.
Fitur tentang aplikasi	Menampilkan informasi tentang aplikasi deteksi jenis tanah	Sistem dapat menampilkan informasi tentang aplikasi deteksi jenis tanah.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Dataset

Pada penelitian ini saya menggunakan dataset yang berisi 3 jenis tanah yaitu Tanah Histosol, Tanah Alluvial, dan Tanah Latosol. Dataset yang digunakan merupakan hasil dari foto langsung yang di dapat dari tempat penelitian.

Tabel 4. 1 Jumlah Dataset

NO	Nama Kelas	Jumlah
1	Tanah Histosol	300 Gambar
2	Tanah Alluvial	300 Gambar
3	Tanah Latosol	300 Gambar
Jumlah		900 Gambar

Berdasarkan tabel 4.1 di atas, dataset terbagi menjadi 3 *class* yaitu Tanah Histosol, Tanah Alluvial, dan Tanah Latosol yang masing – masing classnya berisi 300 gambar dataset. Total keseluruhan dataset yang digunakan yaitu 900 data gambar yang dimana data tersebut di bagi menjadi 3 yaitu 630 *data training* (70%), 180 *data Valid* (10%), dan 90 *data Test* (20%).

4.2 Implementasi Algoritma YOLO v8

4.2.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari 900 gambar dataset kemudian di bagi menjadi 3 *class* dan di anotasi di *roboflow* setelah itu data tersebut di *traning* menggunakan *google colaboratory*. Berikut

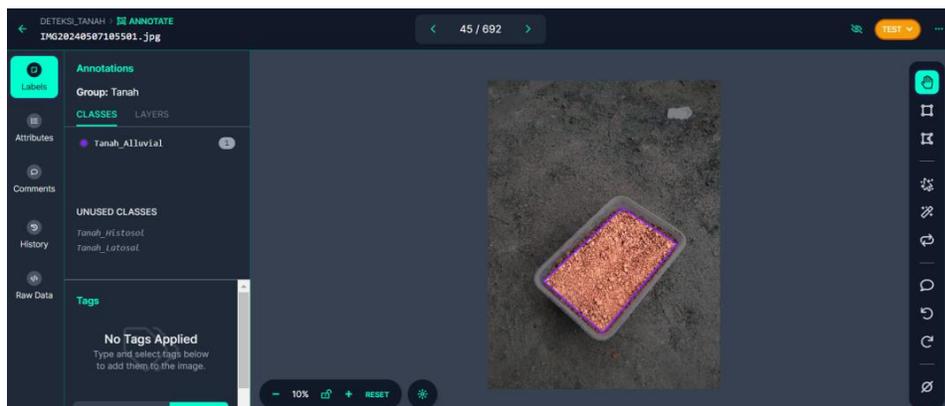
gambar dataset yang akan digunakan pada penelitian ini :



Gambar 4.1 Jenis Tanah

4.2.2 Labeling Dataset

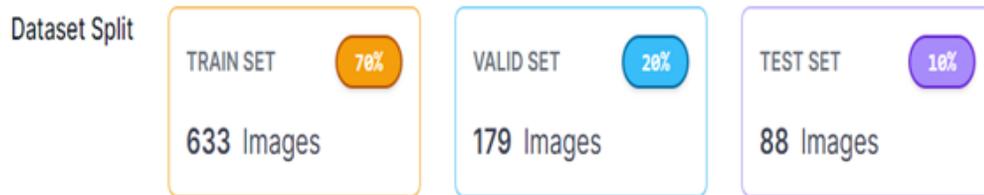
Pada tahapan ini dilakukan *anotasi atau labeling* dataset pada *roboflow* untuk pembuatan *bounding box* berdasarkan *class* jenis tanah yang bertujuan untuk mempermudah *machine learning* dalam melakukan *objek deteksi*. Pada tahapan ini juga dilakukan pembagian dataset yang akan digunakan untuk training model yang terdiri dari *data training, test, dan validation*. Berikutnya tahapan labeling dataset pada *roboflow* :



Gambar 4.2 Proses Pelebelan/Anotasi Gambar

4.2.3 Split Dataset

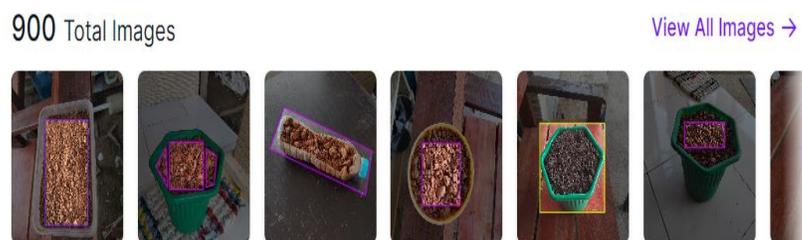
Pada tahapan ini, setelah dilakukan anotasi pada gambar kemudian gambar akan di bagi ke dalam *folder train*, *test*, dan *valid*. Hasil dari pembagian ini bisa dilihat pada gambar 4.3 di bawah.



Gambar 4.3 Split Data

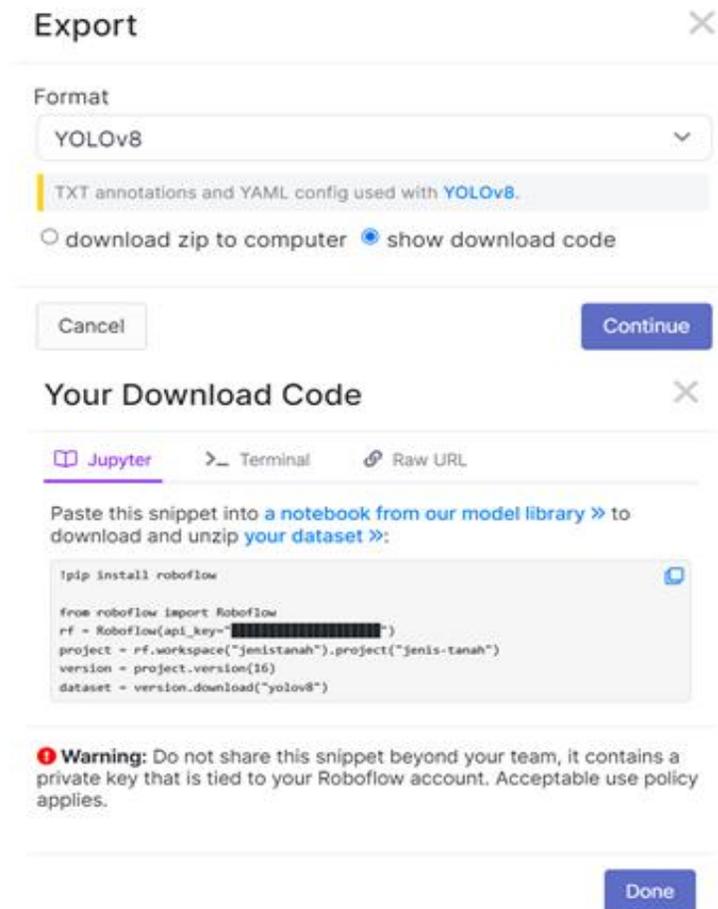
4.2.4 Preprocessing

Pada tahapan ini, setelah dilakukan *split data* selanjutnya yaitu dilakukan *preprocessing* pada dataset, yang dapat dilihat pada gambar 4.4 dibawah :



Gambar 4.4 Split Data

Selanjutnya dilakukan juga pemilihan versi model *yolo V8* kemudian pilih *show download code*, lalu setelah itu salin *code* yang diberikan, kemudian *code* tersebut yang nantinya akan dipakai untuk pengujian model pada *google colab*. Tahapan tersebut dapat dilihat pada gambar 4.5 di bawah :



Gambar 4.5 Pemilihan versi model yolo

4.2.5 Konfigurasi Parameter

Pada tahapan ini, sebelum dataset dilatih akan dilakukan penentuan *hyperparameter* yaitu pada *epoch*, *batch*, dan *img*. Hasil *konfigurasi*

parameter bisa dilihat pada Gambar 4.6 di bawah ini.

```
# Train YOLOv8n on COCO8
!yolo train model=yolov8m.pt data=/content/drive/MyDrive/deteksi/jenis-tanah-18/data.yaml epochs=50 imgsz=640
```

Gambar 4.6 *Konfigurasi Parameter*

4.2.6 Training

Pada tahapan ini, model yang telah di rancang kemudian akan dilatih menggunakan data pelatihan. Proses pelatihan dilakukan dengan iterasi melalui *epoch* dan *batch*. Pada setiap *epoch* mewakili satu siklus melalui seluruh data pelatihan. Dalam setiap *epoch*, data pelatihan akan dibagi menjadi *batch-batch* kecil hal ini bertujuan untuk mengoptimalkan proses pelatihan dengan memproses data secara bertahap dari pada sekaligus. Proses ini menggunakan jumlah 50 *epoch*. Hasil training dapat dilihat pada gambar 4.6 di bawah:

```
50 epochs completed in 1.252 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 52.0MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.2.45 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	179	182	0.892	0.882	0.921	0.769
Tanah-Histosol	62	62	0.857	0.87	0.928	0.789
Tanah_Alluvial	59	62	0.897	0.846	0.869	0.726
Tanah_Latosal	58	58	0.922	0.931	0.967	0.792

```
Speed: 0.3ms preprocess, 8.7ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/train2
Learn more at https://docs.ultralytics.com/modes/train
```

Gambar 4.7 *Training*

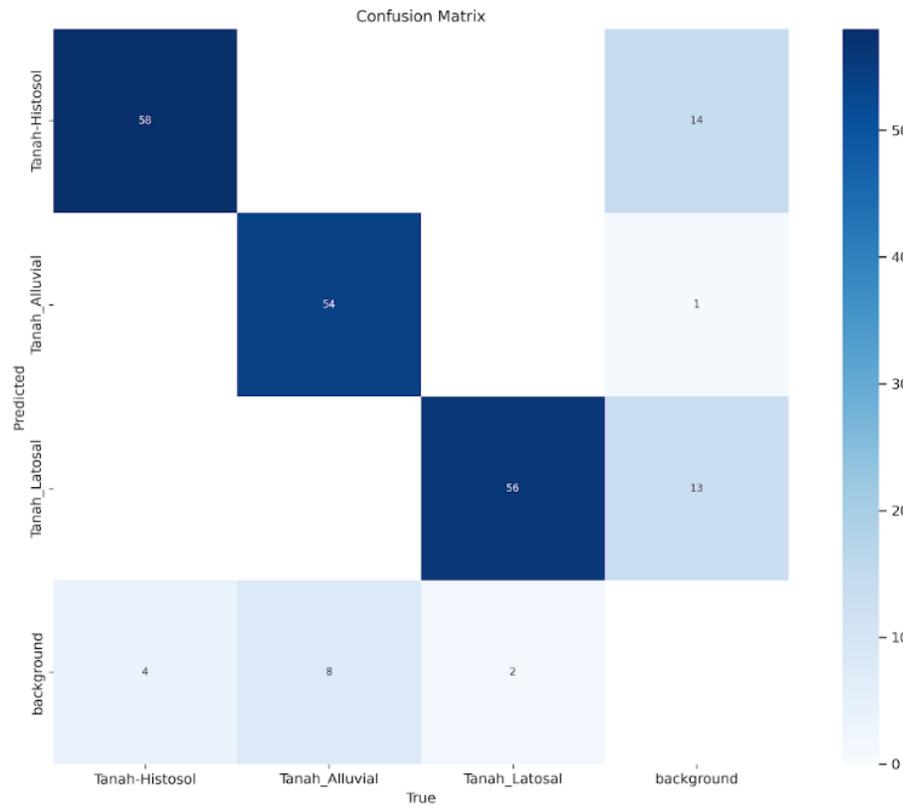
Pada gambar 4.7 di atas dapat dilihat proses *training data* dengan *input* gambar sebesar 640 piksel. Model ini mampu melakukan training dengan estimasi waktu yang kurang lebih 1 jam 30 menit. Semakin tinggi *epoch* maka akan membutuhkan waktu yang cukup lama untuk melakukan *training* model.

4.2.7 Evaluasi

Pada tahapan ini akan dilakukan evaluasi hasil dari *training* dengan melihat *confusion matrix*, *precision curve*, *precision-recal curve*, dan *recal curve*.

4.2.7.1 Confusion Mantrix

Confusion matrix digunakan untuk memberikan pemahaman tentang kinerja model dalam melakukan tugas deteksi objek. Dalam hal ini, *confusion matriks* berfungsi untuk memberikan informasi tentang sejauh mana model dapat mengenali dan membedakan objek yang hadir dalam gambar. *True Positive* dan *True Negative* memberikan informasi ketika *classifier* melakukan pendeteksian data bernilai benar, sedangkan *False Positive* dan *False Negatif* memberikan informasi salah dalam mendeteksi. Berikut merupakan *confusion matrix* yang terdapat pada gambar 4.8 di bawah ini:



Gambar 4.8 Confusion matrix

Berdasarkan gambar 4.8 diatas memperlihatkan *Confusion Matrix* hasil prediksi, berikut ini merupakan tabel dari *Confusion Matrix* di atas :

Tabel 4.2 Confusion Matrix

Predicted Value	Actual Value			
	Aluvial	Histosol	Latosol	Background
Aluvial	58	0	0	14
Histosol	0	54	0	1
Latosol	0	0	56	13
Background	4	8	2	0

Pengukuran kinerja confusion matrix menggunakan 4 istilah yang akan menunjukkan hasil dari proses klasifikasi yaitu:

1. *True Positive* (TP) = Total dari data yang positif dan terklasifikasi oleh sistem dengan benar.
2. *True Negative* (TN) = Total dari data yang positif dan terklasifikasi oleh sistem dengan salah.
3. *False Positive* (FP) = Total dari data yang negatif dan terklasifikasi oleh sistem dengan benar.
4. *False Negative* (FN) = Total dari data yang negatif dan terklasifikasi oleh sistem dengan salah.

Accuracy merupakan hasil dari akurasi untuk mendefinisikan tingkat dari dekatnya nilai aktual dengan nilai hasil prediksi. Jika semakin tinggi tingkat akurasi, maka semakin baik pula kinerja metode. Adapun rumus untuk melakukan perhitungan akurasi seperti pada rumus dibawah ini :

$$\begin{aligned}
 Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\
 &= \frac{58+54+56+0}{168+0+28+14} = \frac{168}{210} = 0,80
 \end{aligned}$$

Recall merupakan tingkat keberhasilan dari sistem untuk mendapatkan informasi. Adapun rumus perhitungannya dapat dilihat di bawah ini :

$$\begin{aligned}
 Recall &= \frac{TP}{TP+FP} \\
 &= \frac{168}{168+28} = \frac{168}{196} = 0,85
 \end{aligned}$$

Precision merupakan tingkat keakuratan data yang diminta dengan data yang diberikan oleh model. *Precision* merupakan klasifikasi TP dari keseluruhan data yang diprediksi positif menjadi nilai kelas positif. Adapun rumus perhitungannya dapat dilihat dibawah ini :

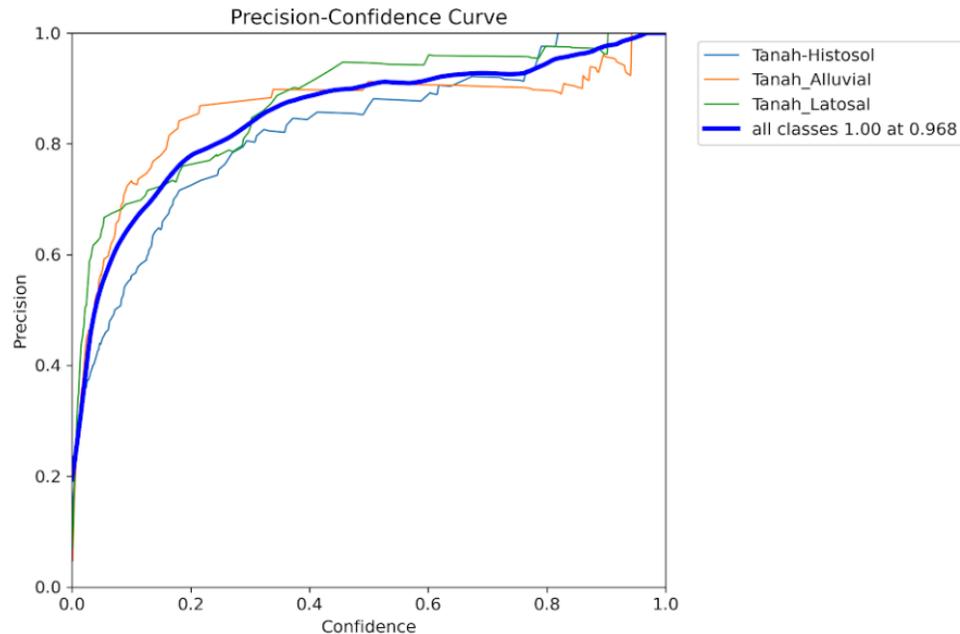
$$\begin{aligned} Precision &= \frac{TP}{TP+FN} \\ &= \frac{168}{168+14} = \frac{168}{182} = 0,92 \end{aligned}$$

F1 - Score merupakan sebuah perbandingan *Precision* dan *Recall*. Adapun rumusnya perhitungan seperti dibawah ini :

$$\begin{aligned} F1 - Score &= 2 \frac{precision*Recall}{Precision+Recall} \\ &= 2 \frac{0,92*0,85}{0,92+0,85} = 2 \frac{0,78}{1,77} = 0,88 \end{aligned}$$

4.2.7.2 Precision Curve

Grafik Precision Curve menunjukkan bagaimana nilai *precision* dari model klasifikasi berubah seiring dengan variasi ambang batas kepercayaan yang digunakan untuk mengklasifikasikan data. Hasil dari Gambar 4.9 di bawah menunjukkan *Precision Curve* setelah proses pelatihan dilakukan.



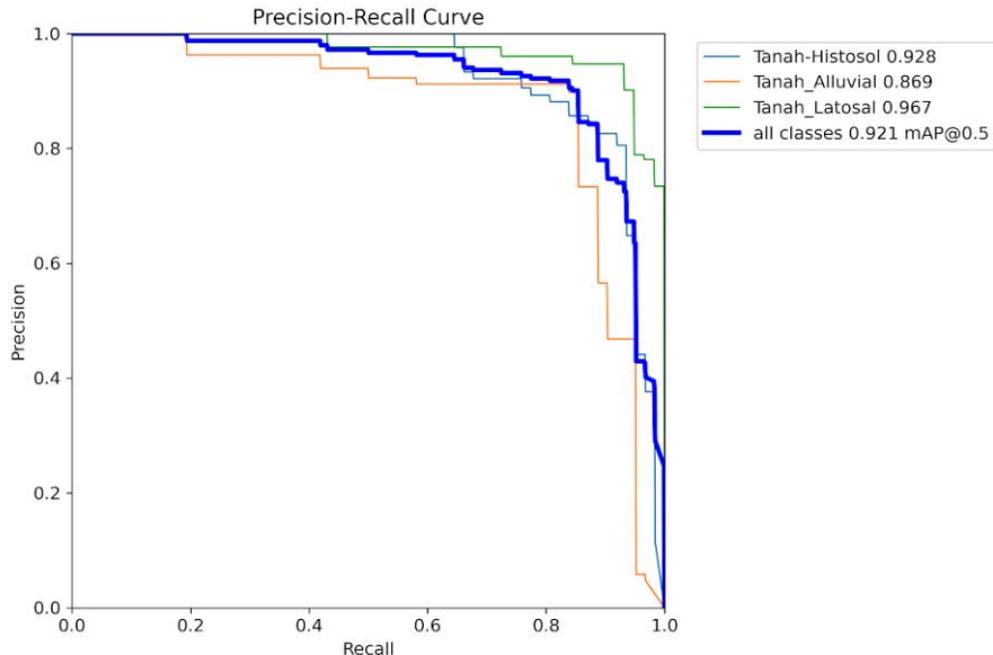
Gambar 4.9 *Precision*

Pada gambar diatas memberikan gambaran tentang bagaimana nilai presisi dari model berubah seiring dengan variasi ambang batas yang digunakan untuk mengklasifikasikan data, yang menghasilkan nilai all class yaitu 1.00 at 0.968, hal ini menandakan kondisi yang baik di mana model dapat mengklasifikasikan semua prediksi positif secara sempurna tanpa kesalahan pada ambang batas 1.000

4.2.7.3 *Precision-Recall Curva*

Pada Gambar 4.10 di bawah menampilkan *hasil kurva precision-recall* setelah proses pelatihan dilakukan. *Precision* menggambarkan seberapa tepat model dalam mengidentifikasi hasil *positif*, sementara *recall* mengukur seberapa banyak hasil *positif* yang

berhasil dideteksi oleh model. Ketika ambang batas kepercayaan ditingkatkan, presisi cenderung meningkat namun recall cenderung menurun, dan sebaliknya.

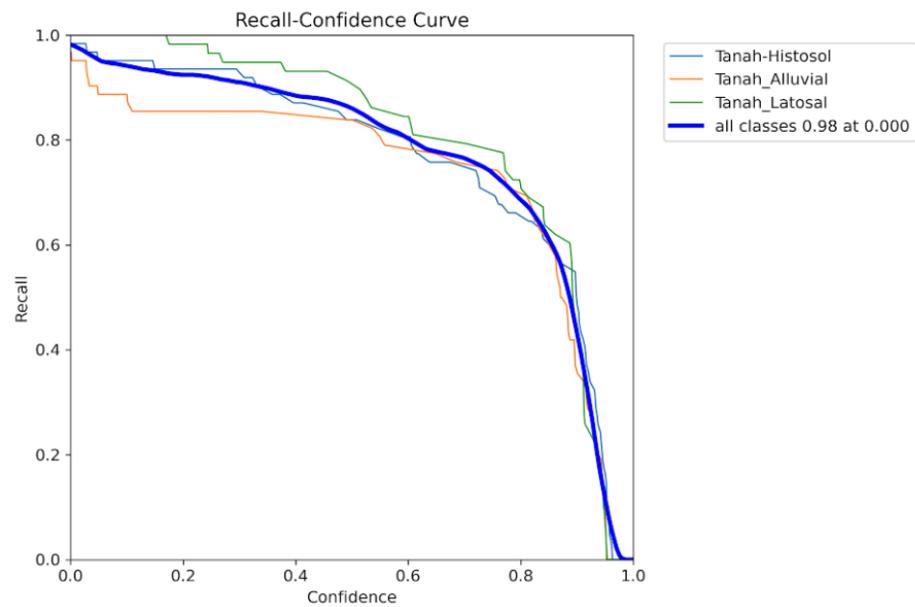


Gambar 4.10 *Precision-Recall*

Pada gambar diatas menunjukkan tingkat *precision-recall* yang tinggi untuk setiap kategori yang diteliti, Tanah Histosol (0.928), Tanah Alluvial (0.869), dan Tanah Latosal (0.967). Rata-rata *precision-recall* untuk semua kelas adalah 0.921, Hal ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengklasifikasikan jenis tanah dengan tingkat akurasi yang tinggi. Nilai mAP ini dihitung pada ambang batas (*threshold*) sebesar 0.5, yang berarti model efektif dalam mendeteksi kasus positif dengan *presisi dan recall* yang tinggi pada ambang batas tersebut.

4.2.7.4 Recall Curve

Kurva recall membantu dalam mengevaluasi kinerja model, dalam mendeteksi objek positif serta menentukan ambang batas kepercayaan yang optimal untuk kebutuhan aplikasi. *Kurva recall* yang baik akan menunjukkan tingkat *recall* yang tinggi, peningkatan yang stabil dan teratur seiring dengan peningkatan ambang batas, penurunan yang lambat, dan memberikan pemahaman yang jelas tentang keseimbangan antara *recall* dan *precision*. Gambar 4.11 di bawah menampilkan *kurva recall* setelah proses pelatihan model dilakukan.



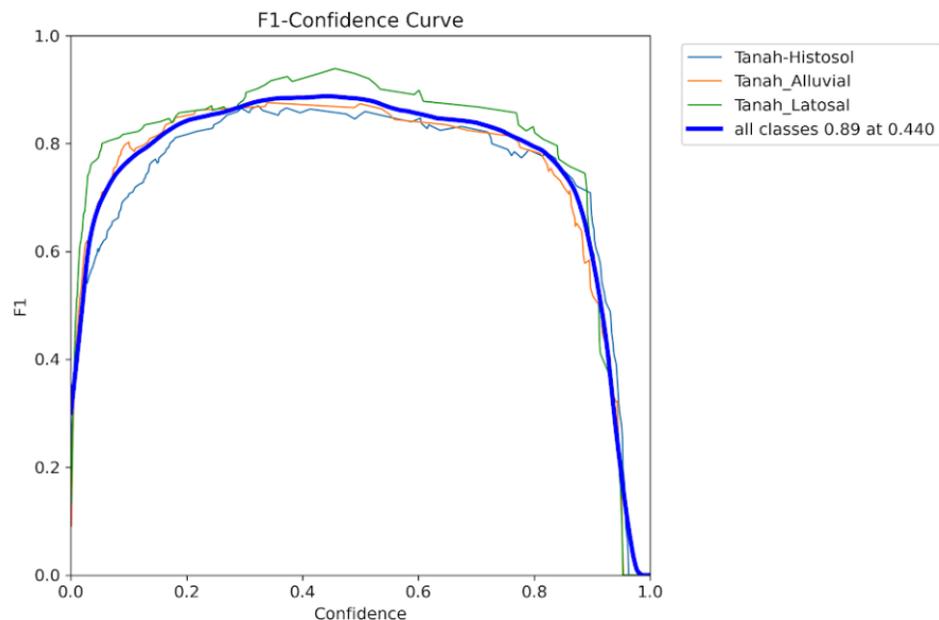
Gambar 4.11 *Recall*

Pada gambar diatas, terlihat bahwa model klasifikasi mencapai tingkat maksimum sebesar 0.98 saat menggunakan ambang batas

(*threshold*) sebesar 0.000. Ini berarti model mampu mengidentifikasi sebagian besar dari semua kasus yang sebenarnya positif dengan tingkat kesalahan yang rendah pada ambang batas yang sangat rendah tersebut. Hasil ini menunjukkan bahwa model memiliki kinerja yang sangat baik dalam menangkap semua kasus yang sebenarnya positif.

4.2.7.5 *F1-Score Curve*

F1-Score adalah sebuah matrix evaluasi yang menggabungkan hasil *precision* dan *recall* melalui rata-rata *harmonik* (jenis perhitungan nilai tengah). Dengan cara ini, *F1-Score* memberikan tinjauan yang seimbang dari kedua metrik. Gambar 4.12 di bawah menampilkan *kurva F1* setelah proses pelatihan model dilakukan.



Gambar 4.12 *F1-Score*

4.2.7.6 Mean Average Precision (mAP)

Mean Average Precision atau *mAP* dapat menghitung rata-rata presisi di atas berbagai level *recall* yang berbeda, memberikan gambaran yang lebih lengkap tentang kinerja sistem deteksi objek dari pada hanya menggunakan *precision* atau *recall* saja. Dengan menggunakan *mAP*, penulis dapat mengevaluasi kinerja sistem secara *holistik* dan mengambil keseimbangan antara *precision* dan *recall*. Pada gambar 4.13 di bawah merupakan *hasil training data* dari *mAP*:

```

50 epochs completed in 1.252 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 52.0MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.2.45 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	179	182	0.892	0.882	0.921	0.769
Tanah-Histosol	62	62	0.857	0.87	0.928	0.789
Tanah_Alluvial	59	62	0.897	0.846	0.869	0.726
Tanah_Latosol	58	58	0.922	0.931	0.967	0.792

```

Speed: 0.3ms preprocess, 8.7ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/train2
🔗 Learn more at https://docs.ultralytics.com/modes/train

```

Gambar 4.13 Hasil Training mAP

4.2.8 Hasil Deteksi

Pada gambar di bawah merupakan gambar *source code* deteksi model yang telah dibuat dengan cara memasukkan gambar lalu di uji apakah gambar tersebut bisa terdeteksi dengan baik atau tidak. Hasil deteksi dapat di lihat pada gambar 4.14 di bawah :

```

!yolo predict model=/content/drive/MyDrive/Bismillah/runs/detect/train/weights/best.pt source='/content/drive/MyDrive/Skr
  Ultralytics YOLOv8.2.22 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
  Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs

  /usr/local/lib/python3.10/dist-packages/torch/nn/modules/conv.py:456: UserWarning: Plan failed with a cudnnException: Cl
    return F.conv2d(input, weight, bias, self.stride,
  image 1/1 /content/drive/MyDrive/Skripsi_Akhir/jenis_tanah/contoh_gambar/gambar_12.jpg: 640x480 1 Tanah_Latosal, 89.9ms
  Speed: 14.4ms preprocess, 89.9ms inference, 1745.7ms postprocess per image at shape (1, 3, 640, 480)
  Results saved to runs/detect/predict4
  Learn more at https://docs.ultralytics.com/modes/predict

```

Gambar 4.14 *Deteksi Model*



Gambar 4.15 *Hasil deteksi menggunakan handphone*

Pada gambar di atas merupakan hasil uji coba menggunakan handphone dengan nilai akurasi masing – masing jenis tanah yaitu tanah Alluvial mendapatkan nilai akurasi sebesar 0.49% dan tanah latosal yaitu 0.33%.



Gambar 4.16 Hasil Deteksi

Pada gambar di atas merupakan hasil uji coba menggunakan *google colab* dimana pada masing – masing jenis tanah memiliki nilai akurasi yang berbeda dimana tanah alluvian mendapatkan hasil akurasi sebesar 0.80%, tanah histosol mendapatkan hasil akurasi sebesar 0.93 dan untuk tanah latosol mendapatkan nilai akurasi sebesar 0.93%.

4.2.9 Saving Model

Pada tahapan ini model yang sudah di buat dan dilakukan training akan di simpan yang nantinya model tersebut akan di gunakan pada *android*. Pada gambar 4.16 di bawah merupakan *source code* yang digunakan untuk menyimpan model.

```

yolo mode=export model=/content/drive/MyDrive/titing/runs/detect/train/weights/best.pt format=onnx imgsz=640
┌───┐
│ Ultralytics YOLOv8.2.18 Python-3.10.12 torch-2.3.0+cu121 CPU (Intel Xeon 2.00GHz) │
│ Model summary (fused): 218 layers, 25841497 parameters, 0 gradients, 78.7 GFLOPs │
│                                                                              │
│ PyTorch: starting from '/content/drive/MyDrive/titing/runs/detect/train/weights/best.pt' with input shape (1, 3, 640, 640) BCHW and output shape(s) (1 │
│ requirements: Ultralytics requirement ['onnx>=1.12.0'] not found, attempting AutoUpdate... │
│ Collecting onnx>=1.12.0 │
│   Downloading onnx-1.16.0-cp310-cp310-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (15.9 MB) │
│ ─────────────────────────────────────────────────────────────────────────────────── 15.9/15.9 MB 288.1 MB/s eta 0:00:00 │
│ Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from onnx>=1.12.0) (1.25.2) │
│ Requirement already satisfied: protobuf>=3.20.2 in /usr/local/lib/python3.10/dist-packages (from onnx>=1.12.0) (3.20.3) │
│ Installing collected packages: onnx │
│ Successfully installed onnx-1.16.0 │
│                                                                              │
│ requirements: AutoUpdate success 8.9s, installed 1 package: ['onnx>=1.12.0'] │
│ requirements: ⚠ Restart runtime or rerun command for updates to take effect │
│                                                                              │
│ ONNX: starting export with onnx 1.16.0 opset 17... │
│ ONNX: export success 12.7s, saved as '/content/drive/MyDrive/titing/runs/detect/train/weights/best.onnx' (98.8 MB) │
│                                                                              │
│ Export complete (17.2s) │
│ Results saved to /content/drive/MyDrive/titing/runs/detect/train/weights │
│ Predict: yolo predict task=detect model=/content/drive/MyDrive/titing/runs/detect/train/weights/best.onnx imgsz=640 │
│ Validate: yolo val task=detect model=/content/drive/MyDrive/titing/runs/detect/train/weights/best.onnx imgsz=640 data=/content/drive/MyDrive/ti

```

Gambar 4.17 *Save Model*

Pada gambar di atas merupakan tahapan untuk meng *convert model best.pt*, menjadi *file .onnx* yang kemudian file tersebut akan diimplementasikan pada *android studio*.

4.3 Implementasi Pada Android

4.3.1 Planning

Sebelum adanya aplikasi ini cara menentukan jenis tanah menggunakan kandungan yang ada pada tanah tersebut. Hal ini membutuhkan waktu yang cukup lama untuk mengetahui jenis tanah, namun dengan adanya aplikasi ini diharapkan dapat membantu mempersingkat waktu untuk mengetahui jenis tanah.

4.3.2 Design

Aplikasi ini menggunakan *Bahasa pemrograman java* untuk diimplementasikan pada *android* dan *bahasa pemrograman python* untuk melakukan *training* pada *google colab*. Proses ini melibatkan fitur – fitur utama seperti *antarmuka pengguna*, *integrasi kamera*, *pengolahan data*, dan *integrasi model jenis tanah*.

A. Tampilan plash

pada gambar 4.17 di bawah ini merupakan tampilan awal pada saat membuka aplikasi yang terdiri beberapa gambar tanah kemudian digabung menjadi logo dari aplikasi ini, kemudian pada bagian lingkaran tanah terndapat keterangan “Deteksi Jenis Tanah” .



Gambar 4.18 Tampilan *Splash Screen*

B. Tampilan Tentang Aplikasi

pada gambar 4.19 di bawah ini merupakan tampilan tentang aplikasi yang dimana pada tampilan ini akan menampilkan informasi terkait aplikasi ini, tidak hanya itu pada gambar di bawah juga terdapat 2 icon yaitu :

a. *icon* pesan tanda tanya

Pada *icon* ini merupakan *icon* yang akan menampilkan penjelasan tentang aplikasi dan penggunaan aplikasi.

b. *Icon Camera*

Icon camera merupakan *icon* yang dimana ketika di tekan maka tampilan akan muncul berupa camera belakang yang dimana user dapat langsung mengarahkan camera yang telah aktif pada objek yang akan di deteksi.



Gambar 4.19 *Tentang Aplikasi*

C. Tampilan Camera

Pada gambar 4.20 di bawah merupakan tampilan camera untuk mendeteksi jenis tanah, selain itu terdapat juga tanda panah untuk kembali ke menu utama.



Gambar 4.20 Camera

4.3.3 Testing

Pada tahapan ini, sistem diuji menggunakan metode *blackbox testing* untuk memeriksa semua fitur yang telah dirancang pada aplikasi, serta memastikan bahwa semuanya berfungsi dengan baik sesuai dengan yang di harapkan. Hasil pengujian dengan metode *blackbox* dapat di lihat pada tabel

4.2 di bawah ini:

Tabel 4.2 Hasil *Black Box Testing*

Uji Kasus	Skenario Uji Coba	Hasil Yang Didapatkan	Kesimpulan
Menjalankan sistem	Menjalankan dan mengakses aplikasi jenis tanah	Apakah aplikasi berhasil di akses dengan lancar.	Berhasil
Menekan tombol camera	Menyalakan camera dengan cara mengklik fitur camera	Aplikasi dapat mengakses camera sebagai input data yang ingin dideteksi	Berhasil
Objek selain Tanah alluvia, latosal dan histosol	Mendeteksi objek	Aplikasi tidak dapat mendeteksi jenis tanah	Berhasil
<i>Output</i>	Menampilkan hasil deteksi jenis tanah	Sistem dapat menampilkan hasil berupa bounding box sesuai dengan objek yang dideteksi.	Berhasil
Fitur tentang aplikasi	Menampilkan informasi tentang aplikasi deteksi jenis tanah	Sistem dapat menampilkan informasi tentang aplikasi deteksi jenis tanah.	Berhasil

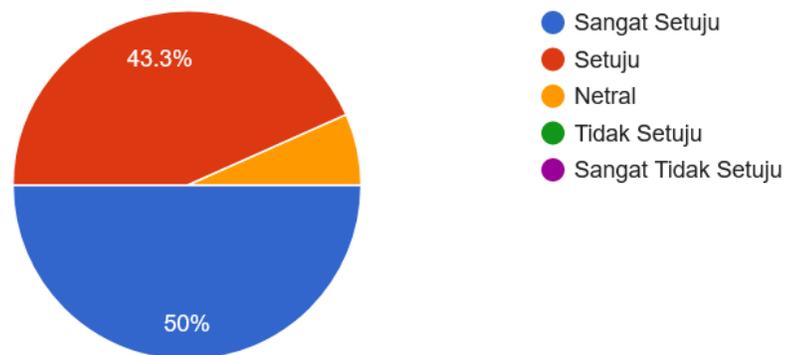
Berdasarkan tabel 4.2 hasil *blackbox testing* dapat disimpulkan bahwa sistem dapat berjalan dengan baik dan memenuhi kebutuhan yang sesuai dengan yang diinginkan.

4.4 Usability Testing

Berikut ini merupakan hasil *kuesioner* yang didapatkan dari 30 *responden* yang terdiri dari 11 laki-laki dan 19 perempuan. Data dikumpulkan menggunakan *google form* dengan presentasi yang diperoleh sebagai berikut :

Apakah aplikasi ini berjalan dengan baik?

30 responses

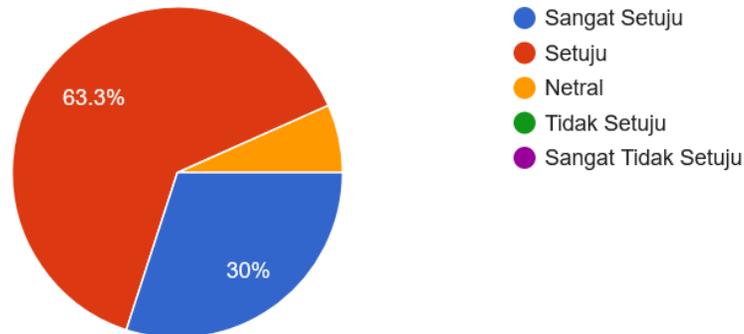


Gambar 4.21 *Diagram Pertanyaan 1*

Berdasarkan gambar 4.21 di atas pada pertanyaan “apakah aplikasi ini berjalan dengan baik?” mendapat 30 respon yang diantaranya memilih Sangat Setuju mencapai 50% atau 15 orang, pilihan Setuju mencapai 43,3% atau 13 orang dan pilihan Netral mencapai 6,7% atau 2 orang.

Apakah aplikasi ini mudah untuk dipahami?

30 responses

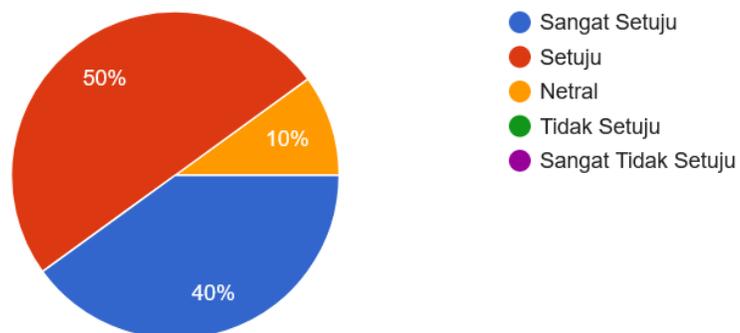


Gambar 4.22 Diagram Pertanyaan 2

Berdasarkan gambar 4.22 di atas pada pertanyaan “apakah aplikasi ini mudah untuk dipahami?” mendapat 30 respon yang diantaranya memilih Sangat Setuju mencapai 30% atau 9 orang, pilihan setuju mencapai 63,3% atau 19 orang dan pada pada pilihan Netral mencapai 6,7% atau 2 orang.

Apakah fitur pada aplikasi ini berjalan dengan baik?

30 responses

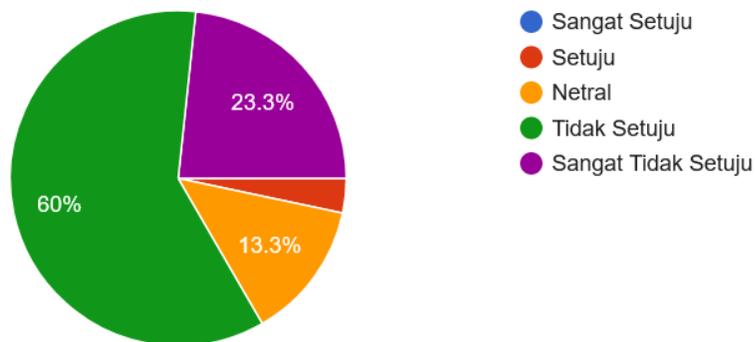


Gambar 4.23 Diagram pertanyaan 3

Berdasarkan gambar 4.23 di atas pada pertanyaan “apakah fitur pada aplikasi ini berjalan dengan baik?” mendapat 30 respon yang diantaranya memilih Sangat Setuju mencapai 40% atau 12 orang, pada pilihan setuju mencapai 50% atau 15 orang dan pada pilihan Netral mencapai 10% atau 3 orang.

Apakah proses deteksi aplikasi ini terdapat kesalahan?

30 responses

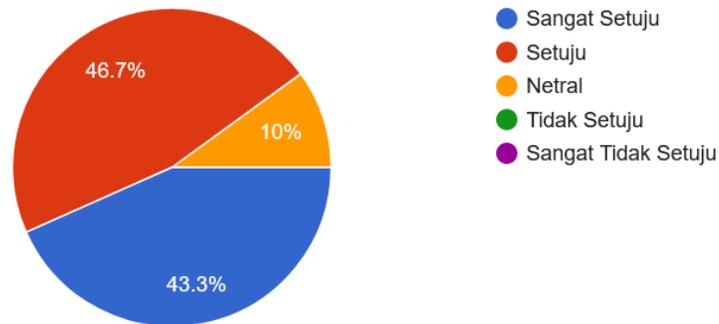


Gambar 4.24 Diagram Pertanyaan 4

Berdasarkan gambar 4.24 di atas pada pertanyaan “apakah Proses Deteksi Aplikasi Ini Terdapat kesalahan?” mendapat 30 respon yang diantaranya memilih pada pilihan setuju mencapai 3,3% atau 1 orang, pada pilihan Netral mencapai 13,3% atau 4 orang pada pilihan Tidak Setuju mencapai 60% atau 18 orang, dan Sangat Tidak Setuju mencapai 23,3% atau 7 orang.

Seberapa besar tingkat kepuasan anda dalam menggunakan aplikasi ini. Apakah sangat puas?

30 responses



Gambar 4.25 Diagram Pertanyaan 5

Berdasarkan gambar 4.25 di atas pada pertanyaan “Seberapa besar tingkat kepuasan anda dalam menggunakan aplikasi ini. Apakah sangat puas?” mendapat 30 respon yang diantaranya memilih Sangat Setuju mencapai 43.3% atau 13 orang, pada pilihan setuju mencapai 46,7% atau 14 orang, dan pada pilihan Netral mencapai 10% atau 3 orang.

Berdasarkan hasil kuisisioner di atas, selanjutnya hasil tersebut dihitung menggunakan *skala liker* (1-4) yang digunakan untuk mengukur *responden*. Tabel skor *skala likert* dapat dilihat pada tabel 4.3 di bawah ini :

Tabel 4.3 Skor Kuisisioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
S	Setuju	4
N	Netral	3

Skala Jawaban	Keterangan	Skor
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Berdasarkan data skor kuisioner pada tabel 4.3 didapat dicari persentasenya masing-masing jawaban dengan menggunakan rumus :

$$Y = \frac{x \text{ Skor Ideal}}{X} \dots \dots (1)$$

Keterangan :

Y = Nilai persentase yang dicari (..%).

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi ($\Sigma = N \times R$).

N = Nilai dari setiap jawaban.

R = Frekuensi.

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel ($5 \times 30 = 150$).

Setelah diperoleh nilai Y, maka selanjutnya mencari interval (rentang jarak) dan interpretasi persen menggunakan rumus di bawah ini :

Rumus Interval

$$I = 100/\text{Jumlah Skor (Likert)}$$

$$\text{Maka } I = 100 / 5 = 20$$

$$\text{Hasil (I) = 20}$$

Ini adalah interval jarak dari terendah 0% hingga tertinggi 100%.

Berikut kriteria interpretasi skor berdasarkan interval yaitu :

Angka 0% – 19,99% = Sangat Tidak Setuju

Angka 20% – 39,99% = Tidak Setuju

Angka 40% – 59,99% = Cukup Setuju

Angka 60% – 79,99% = Setuju

Angka 80% – 100% = Sangat Setuju

Tabel 4.4 Skor Pertanyaan 1

pertanyaan	Keterangan	Skor (N)	Responden ®	NxR
Aplikasi ini berjalan dengan baik?	Sangat Setuju	5	15	75
	Setuju	4	13	52
	Cukup Setuju	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
			30	133
		$Y = \frac{133}{150} \times 100$		88,6%

Tabel 4.5 Skor Pertanyaan 2

pertanyaan	Keterangan	Skor (N)	Responden ®	NxR
apakah aplikasi ini mudah untuk dipahami?	Sangat Setuju	5	9	45
	Setuju	4	19	76
	Cukup Setuju	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
			30	127

$Y = \frac{127}{150} \times 100$	84,6%
----------------------------------	-------

Tabel 4.6 Skor Pertanyaan 3

pertanyaan	Keterangan	Skor (N)	Responden ®	NxR
Apakah fitur ada aplikasi ini berjalan dengan baik?	Sangat Setuju	5	12	60
	Setuju	4	15	60
	Cukup Setuju	3	3	9
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
			30	129
$Y = \frac{129}{150} \times 100$				86%

Tabel 4.7 Skor Pertanyaan 4

pertanyaan	Keterangan	Skor (N)	Responden ®	NxR
Apakah proses deteksi aplikasi ini terdapat kesalahan?	Sangat Setuju	5	0	0
	Setuju	4	1	4
	Cukup Setuju	3	4	12
	Tidak Setuju	2	18	36
	Sangat Tidak Setuju	1	7	7
			30	59
$Y = \frac{59}{150} \times 100$				50,8%

Tabel 4.8 Skor Pertanyaan 5

pertanyaan	Keterangan	Skor (N)	Responden ®	NxR
Seberapa besar tingkat kepuasan anda dalam menggunakan	Sangat Setuju	5	13	65
	Setuju	4	14	56
	Cukup Setuju	3	3	9

aplikasi ini. Apakah sangat puas?	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
			30	130
$Y = \frac{130}{150} \times 100$				86,6%

Berikut merupakan perhitungan yang di dapatkan

$$\frac{88,6 + 84,6 + 86 + 50,8 + 86,6}{5} = 79,3$$

Berdasarkan hasil perhitungan rata rata menggunakan usability testing pada 5 kuisisioner di atas maka dapat di simpulkan bahwa pembuatan sistem berbasis android pada deteksi jenis tanah berdasarkan warna sudah sesuai dengan perolehan presentase mencapai 79,4% dengan keterangan **Setuju**

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Hasil implementasi sistem berbasis *Android* dengan menggunakan metode *you only look once V8* (YOLO V8) memperoleh skor *usability testing* sebesar 79,5%, sehingga dapat disimpulkan bahwa penelitian ini berhasil dilakukan dalam mendeteksi jenis tanah berdasarkan warna.
2. Penelitian ini juga mendapatkan hasil akurasi yang cukup tinggi dari pengujian *black box testing* yaitu sebesar 88% yang mana dapat di katakan bahwa penelitian ini berhasil.

5.2 Saran

Penulis menyadari bahwa penelitian ini masih jauh dari pada sempurna, oleh karena itu penulis menginginkan adanya pengembangan dalam aplikasi ini agar memperoleh hasil yang lebih baik lagi. Diantaranya saran dari penulis yaitu sebagai berikut :

1. Penelitian selanjutnya diharapkan dapat menggunakan metode yang baru atau algoritma yang lebih baru lagi.
2. Penelitian selanjutnya diharapkan dapat menggunakan jenis tanah yang lebih bervariasi.
3. Penelitian selanjutnya diharapkan dapat membuat sistem yang dapat

mendeteksi yang lebih cepat.

4. Penelitian selanjutnya diharapkan dapat mendeteksi jenis tanah secara langsung tanpa menggunakan wadah.
5. Penelitian selanjutnya diharapkan dapat menggunakan dataset yang lebih banyak lagi dan lebih bervariasi.
6. Penelitian selanjutnya diharapkan dapat menggunakan dataset yang tidak memiliki background.

DAFTAR PUSTAKA

- Anendya, Aorinka. 2022. “Pengertian Usability Testing: Metode Dan Manfaatnya Bagi Bisnis.” *Dewaweb*. Retrieved (<https://www.dewaweb.com/blog/pengertian-usability-testing/>).
- Arifita Arwaz, Aini, Kurnia Putra, Rediansyah Putra, Theja Kusumawijaya, and Aries Saifudin. 2019. “Jurnal Teknologi Sistem Informasi Dan Aplikasi Pengujian Black Box Pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions.” *Jurnal Teknologi Sistem Informasi Dan Aplikasi* 2(4):2654–4229.
- Aulia, Dhia Shafa, Casi Setianingsih, and Meta Kallista. 2021. “DETEKSI TANDA KEHIDUPAN PADA KORBAN BENCANA ALAM DENGAN ALGORITMA YOLO DAN OPEN POSE LIFE SIGN DETECTOR USING YOU ONLY LOOK ONCE (YOLO) AND OPEN POSE.” 8(5):6162–74.
- D Fermansah. 2019. “Cara Kerja Artificial Neural Network.” *Journal of Chemical Information and Modeling* 53(9):1689–99.
- Dalimunthe, Annafiah. 2021. “Deteksi Kematangan Buah Manggis Berdasarkan Fitur Warna Citra Kulit Menggunakan Metode Transformasi Ruang Warna HSV.” *Skripsi* 89.
- Darwadi, Salman Hakim. 2020. “Jenis-Jenis Tanah Dan Persebarannya.” *Pahamifi*. Retrieved (<https://pahamify.com/blog/pahami-materi/materi-ips/jenis-jenis-tanah-dan-persebarannya/>).
- Digitalaskola. 2023. “Definsi, Manfaat, Hingga Cara Menggunakan Google Colab Python.” Retrieved (<https://digitalaskola.com/blog/home/google-colab-python>).
- Ferdeanty, Sufardi, and Teti Arabia. 2019. “Karakteristik Morfologi Dan Klasifikasi Tanah Andisol Di Lahan Kering Kabupaten Aceh Besar Karakteristik Morfologi Dan Klasifikasi Tanah Andisol Di Lahan Kering Kabupaten Aceh Besar (Characteristics Morphology and Soil Classification of Andisol at Dry La.” *Jurnal Ilmiah Mahasiswa Peratanian* 4(4):666–76.
- Fitriyah, Hurriyatul, Rizal Maulana, Fakultas Ilmu Komputer, Universitas Brawijaya

- Malang, and Penulis Korespondensi. 2021. "Weed Detection in Hsv Color-Space Based on Shape Features Using." *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)* 8(5):929–38. doi: 10.25126/jtiik.202184719.
- Gerald, Calvin, and Chairisni Lubis. 2020. "Pendeteksian Dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once Dan Convolutional Neural Network." *Jurnal Ilmu Komputer Dan Sistem Informasi* 8(2):197. doi: 10.24912/jiksi.v8i2.11495.
- H, Junita Sri Wisna, Tekad Matulatan, and Nurul Hayaty. 2020. "Jurnal Sustainable : Jurnal Hasil Penelitian Dan Industri Terapan Deteksi Kendaraan Secara Real Time Menggunakan Metode YOLO Berbasis Android." 09(01).
- Helnawan, Aldy, Muhammad Attamimi, and Astria Nur Irfansyah. 2023. "Sistem Segmentasi Jalan Dan Objek Untuk Kendaraan Otonom Menggunakan Kamera RGB-D." *Jurnal Teknik ITS* 12(1). doi: 10.12962/j23373539.v12i1.110848.
- Kusuma, Dedy Hidayat. 2023. "Deteksi Lampu Lalu Lintas Menggunakan YOLO Untuk Autonomous Car." 1:21–27.
- Maneno, Regolinda, Budiman Baso, Patricia G. Manek, and Kristoforus Fallo. 2023. "Deteksi Tingkat Kematangan Buah Pinang Menggunakan Metode Support Vector Machine Berdasarkan Warna Dan Tekstur." *Journal of Information and Technology* 3(2):60–66. doi: 10.32938/jitu.v3i2.5323.
- Muhammad, Ariffud. 2023. "Use Case Diagram: Pengertian, Contoh, Simbol, Cara Membuatnya." *NIAGAHOSTER*. Retrieved (<https://www.niagahoster.co.id/blog/use-case-diagram-adalah/>).
- Nasional, Seminar, Terapan Riset, and X. X. No. 2021. "KELEBIHAN PENUMPANG PADA DENGAN MENGGUNAKAN YOLO V3 Jurusan Teknik , Politeknik Negeri Madiun , Jalan Serayu No . 84 , Madiun , 63133 PENDAHULUAN Berdasarkan Data Dari Badan Pusat Statistik Jumlah Korban Meninggal Dalam Kejadian Kecelakaan Di Indonesia Seb." 430–38.
- Normawati, Dwi, and Surya Allit Prayogi. 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.

- Novianto, Dian, and Tri Sugihartono. 2020. "Sistem Deteksi Kualitas Buah Jambu Air Berdasarkan Warna Kulit Menggunakan Algoritma Principal Component Analysis (Pca) Dan K-Nearest Neighbor (K-NN)." *Jurnal Ilmiah Informatika Global* 11(2):42–47. doi: 10.36982/jiig.v11i2.1223.
- Purwono, Nugroho, Danang Budi Susetyo, Seftiawan Samsu Rijal, Gina Koedatu Syaripah, and Siti Munawaroh. 2024. "Assessing Urban Growth in Greater Surabaya Using Google Earth Engine: An Evaluation of Built-Up Area Expansion in Indonesian Secondary Cities." *Journal of the Geographical Institute Jovan Cvijic SASA* 74(1):127–38. doi: 10.2298/IJGI230608004P.
- Rahma, Lusiana, Hadi Syaputra, A. Haida. Mirza, and Susan Dian Purnamasari. 2021. "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)." *Jurnal Nasional Ilmu Komputer* 2(3):213–32. doi: 10.47747/jurnalnik.v2i3.534.
- Rani, Prita, Manigopa K. Ashwani Anjani Amrit Kumar, and Awadhesh Arun Ashwani Anjani Amrit Kumar. 2020. "PENERAPAN ANALYTICAL HIERARCHY PROCESS UNTUK SISTEM PENDUKUNG KEPUTUSAN PENENTUAN JENIS TANAH TERBAIK BAGI TANAMAN PORANG (AMORPHOPHALLUS MUELLERI)." *Range Management and Agroforestry* 4(1):1–15. doi: 10.1016/j.fcr.2017.06.020.
- Raup, Abdul, Wawan Ridwan, Yayah Khoeriyah, Supiana Supiana, and Qiqi Yuliaty Zaqiah. 2022. "Deep Learning Dan Penerapannya Dalam Pembelajaran." *JIIP - Jurnal Ilmiah Ilmu Pendidikan* 5(9):3258–67. doi: 10.54371/jiip.v5i9.805.
- Rizkatama, Getsa Novandra, Anan Nugroho, and Alfa Faridh Suni. 2021. "Sistem Cerdas Penghitung Jumlah Mobil Untuk Mengetahui Ketersediaan Lahan Parkir Berbasis Python Dan YOLO V4." *Edu Komputika Journal* 8(2):91–99. doi: 10.15294/edukomputika.v8i2.47865.
- Rohaedi, Edi, Isep H. Insan, and Nadia Zumaro. 2019. "Mekanisme Pengadaan Tanah Untuk Kepentingan Umum." *Palar / Pakuan Law Review* 5(2):198–220. doi: 10.33751/.v5i2.1192.
- Sahputra, Marwian Aditya, Meriska Defriani, and Teguh Iman Hermanto. 2023.

- “Rancan Bangun Aplikasi Pelayanan E-Trayek Berbasis Mobile Menggunakan Metode Extreme Programming.” *Sudo Jurnal Teknik Informatika* 2(1):34–44. doi: 10.56211/sudo.v2i1.229.
- Saputra, Ade chandra, and Enny Dwi Oktaviyani. 2023. “Rancang Bangun Sistem Deteksi Kematangan Buah Kelapa Sawit Berdasarkan Deteksi Warna Menggunakan Algoritma K-Nn.” *Jurnal Teknologi Informasi: Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika* 7(2):222–29. doi: 10.47111/jti.v7i2.9232.
- Setiawan, Rony. 2021a. “Black Box Testing Untuk Menguji Perangkat Lunak.” *Dicoding Indonesia*. Retrieved (<https://www.dicoding.com/blog/black-box-testing/>).
- Setiawan, Rony. 2021b. “Flowchart Adalah: Fungsi, Jenis, Simbol, Dan Contohnya.” *DICODING*. Retrieved (<https://www.dicoding.com/blog/flowchart-adalah/>).
- Shridhar, Ayush, Phil Tomson, and Mike Innes. 2020. “Interoperating Deep Learning Models with ONNX.Jl.” *JuliaCon Proceedings* 1(1):59. doi: 10.21105/jcon.00059.
- Triono, Agus, Apri Setia Budi, and Rafi Abdillah. 2023. “Implementasi Peretasan Sandi Vigenere Chipper Menggunakan Bahasa Pemograman Python.” *Jurnal JOCOTIS - Journal Science Informatika and Robotics* 1(1):1–9.
- UMSU, Editor. 2022. “Tanah Latosol : Ciri, Jenis, Manfaat, Dan Berasal Dari?” *Fakultas Pertanian UMSU*. Retrieved (<https://faperta.umsu.ac.id/2022/03/20/tanah-latosol/>).
- Wibowo, Dimas catur. 2019. “Apa Itu Android Studio Dan Android SDK?” *Dicoding*. Retrieved (<https://www.dicoding.com/blog/apa-itu-android-studio-dan-android-sdk/>).
- Widodo, Andrian Eko, Aprih Widayanto, and Sardianto. 2021. “Rancang Bangun Sistem Informasi Manajemen Pemasaran Properti Berbasis Android.” *Indonesian Journal on Networking and Security* 10(2):91–98.

LAMPIRAN

Lampiran A. Form Validasi Pakar

FORM VALIDASI PAKAR
JENIS TANAH



Gambar 1. Jenis Tanah Latosol)



Gambar 2. Jenis Tanah Alluvial



Histosol
Gambar 3. Jenis Tanah Humus (Organosol)

No	Pertanyaan	Benar	Salah
1	Apakah Gambar 1 Merupakan Jenis Tanah Latosol?	✓	
2	Apakah Gambar 2 Merupakan Jenis Tanah Alluvia?	✓	
3	Apakah Gambar 3 Merupakan Jenis Tanah Humus (Organosol)?	✓	

PAKAR

Mira
MIRA U. SIBEKAMTO, SP, MP

Lampiran B. Penggunaan Aplikasi

