

PENGEMBANGAN APLIKASI “ITERA-VISION-GPR-UI (ITERA VISION SYSTEM FOR GPR UTILITIES INFERENCE)” BERBASIS COMPUTER VISION DENGAN MODEL YOLOV12 UNTUK DETEKSI UTILITAS PADA METODE GROUND PENETRATING RADAR

Raka Putra Pratama Rumansah* , Andri Yadi Paembonan , Putu Pradnya Andika

Teknik Geofisika, Fakultas Teknologi Industri, Institut Teknologi Sumatera, Jl. Terusan Ryacudu Way Hui, Kecamatan Jati Agung, Lampung Selatan 35365, Indonesia

*email: rekatea26@gmail.com

ABSTRAK

Permasalahan dalam interpretasi data Ground Penetrating Radar (GPR) adalah ketergantungan terhadap keahlian operator, waktu analisis yang lama, dan potensi subjektivitas hasil. Untuk mengatasi hal tersebut, penelitian ini mengimplementasikan teknologi computer vision menggunakan model YOLOv12 (You Only Look Once) yang mampu mendeteksi pola hiperbolik secara otomatis. Pengembangan aplikasi berbasis computer vision bernama “ITERA-VISION-GPR-UI (ITERA Vision System for GPR Utilities Inference)” dirancang untuk mendeteksi utilitas bawah permukaan menggunakan citra radargram dari metode GPR. Dataset penelitian terdiri dari 82 citra radargram yang telah melalui tahapan pre-processing dan labelling menggunakan Roboflow. Pelatihan model dilakukan pada Google Colaboratory menggunakan varian YOLOv12s dengan parameter utama epochs 500, optimizer AdamW, input imsz 640 pixel (640 x 640 x 3), batch 1 dan patience 100. Dalam menguji performa model, matrix utama yang digunakan adalah Precision, Recall, Mean Average Precision pada ambang batas IoU 0,50 (mAP50), dan Mean Average Precision pada banyak ambang IoU 0,50-0,95 (mAP50-95). Pada epoch ke-292, hasil evaluasi matrix menunjukkan bahwa model mencapai nilai Precision sebesar (0,85), Recall (0,77), mAP50 (0,82), dan mAP50-95 sebesar (0,51) dengan tingkat akurasi keseluruhan mencapai 0,69. Nilai-nilai tersebut membuktikan bahwa model memiliki performa yang cukup baik dalam mendeteksi utilitas bawah permukaan. Meski terdapat beberapa kekurangan, Aplikasi yang dibangun kelak dapat membantu geophysicist dalam melakukan interpretasi citra radargram secara lebih efisien, cepat, dan akurat.

Kata Kunci: GPR; ITERA-VISION-GPR-UI; Computer vision; YOLOv12; Utilitas

ABSTRACT

[Title: Pengembangan Aplikasi “ITERA-VISION-GPR-UI (ITERA Vision System for GPR Utilities Inference)” Berbasis Computer Vision dengan Model YOLOv12 untuk Deteksi Utilitas pada Metode Ground Penetrating Radar] *The primary challenge in interpreting Ground Penetrating Radar (GPR) data lies in its strong dependence on operator expertise, the lengthy analysis time, and the potential subjectivity of the results. To address these limitations, this study implements computer vision techniques using the YOLOv12 (You Only Look Once) model, which is capable of automatically detecting hyperbolic patterns commonly associated with subsurface utilities. A web-based application named “ITERA-VISION-GPR-UI (ITERA Vision System for GPR Utilities Inference)” was developed to perform object detection on GPR radargram images for subsurface utility identification. The dataset used in this research consists of 82 radargram images that underwent pre-processing and labelling using Roboflow. Model training was conducted on Google Colaboratory using the YOLOv12s variant with key parameters including 500 epochs, the AdamW optimizer, an input image size of 640 pixels (640 × 640 × 3), batch size of 1, and a patience value of 100. Model performance was evaluated using four main metrics: Precision, Recall, Mean Average Precision at an IoU threshold of 0.50 (mAP50), and Mean Average Precision across multiple IoU thresholds from 0.50 to 0.95 (mAP50-95). At epoch 292, the model achieved a Precision of 0.85, Recall of 0.77, mAP50 of 0.82, and mAP50-95 of 0.51, with an overall accuracy of 0.69. These results indicate that the model demonstrates satisfactory performance in detecting subsurface utilities. Despite certain limitations, the developed application has the potential to assist geophysicists in interpreting radargram images more efficiently, rapidly, and accurately.*

Keywords: GPR; ITERA-VISION-GPR-UI; Computer vision; YOLOv12; Utilities

1. PENDAHULUAN

Penentuan posisi dan identifikasi utilitas bawah permukaan merupakan salah satu tantangan utama dalam proyek konstruksi, pemeliharaan jaringan bawah tanah, serta upaya mitigasi risiko terhadap kerusakan infrastruktur. Proses penggalian tanpa pemahaman yang akurat mengenai keberadaan pipa, kabel, atau saluran lainnya sering kali menyebabkan kerusakan utilitas aktif, peningkatan biaya perbaikan, hingga gangguan operasional yang berkepanjangan.

Seiring meningkatnya kebutuhan akan pemetaan utilitas bawah permukaan, berbagai metode geofisika terus dikembangkan untuk meminimalkan ketidakpastian dalam proses identifikasi struktur bawah permukaan. Metode *Ground Penetrating Radar* (GPR) telah lama dimanfaatkan untuk mendeteksi keberadaan utilitas di bawah permukaan melalui pengolahan data dan interpretasi. Namun, peran geofisika dalam proses interpretasi manual masih kurang efisien, karena memerlukan keahlian yang cukup baik, waktu analisis yang lama, serta rentan terhadap subjektivitas interpreter (Haron dkk., 2024). Hal ini menjadi kendala utama dalam penerapan GPR secara luas untuk kebutuhan interpretasi yang lebih cepat dan efisien.

Sebagai solusi terhadap keterbatasan tersebut, teknologi *computer vision* hadir sebagai pendekatan modern yang mampu melakukan analisis berdasarkan citra secara otomatis dan efisien. Dengan memanfaatkan algoritma deteksi objek terkini seperti arsitektur YOLOv12 (*You Only Look Once*), sistem dapat memproses citra radargram dan melakukan inferensi terhadap objek berbentuk hiperbolik secara akurat. Pendekatan ini menggantikan proses interpretasi manual menjadi sistem berbasis kecerdasan buatan yang lebih cepat, konsisten, dan dapat diandalkan.

Berdasarkan penelitian yang dilakukan oleh Cheng dkk. (2024), metode *computer vision* telah terbukti efektif dalam deteksi objek kompleks di berbagai bidang, termasuk pengolahan citra geospasial dan analisis data bawah permukaan. Selain itu, penelitian yang dilakukan oleh Hou dkk. (2025) menunjukkan keberhasilan dalam penerapan *computer vision* untuk deteksi objek pada citra radargram GPR dengan pola refleksi berbentuk hiperbolik, melalui penerapan berbagai arsitektur yang berbeda serta penggunaan metode komputasi yang lebih kompleks dan adaptif.

Proses interpretasi data GPR hingga saat ini masih didominasi oleh analisis manual, yang memiliki sejumlah keterbatasan (Haron dkk., 2024). Di sisi

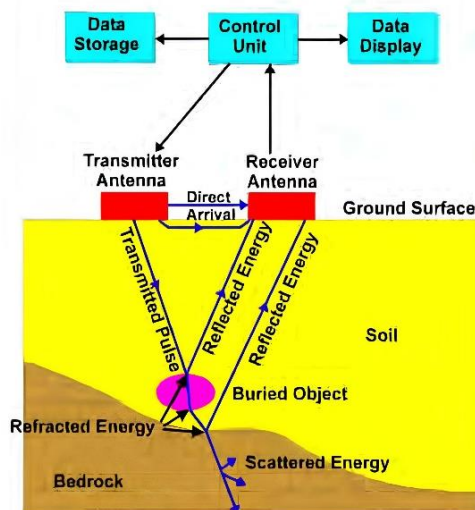
lain, meskipun perkembangan teknologi *computer vision* berlangsung sangat cepat, pemanfaatannya untuk mendeteksi pola hiperbolik pada radargram yang menjadi indikasi umum keberadaan utilitas bawah permukaan belum banyak diterapkan secara komprehensif. Sebagian besar penelitian sebelumnya hanya berfokus pada pengujian model tanpa mengembangkan aplikasi siap pakai untuk keperluan interpretasi. Kondisi tersebut memperlihatkan adanya celah berupa kebutuhan akan sistem otomatis yang mampu mempercepat dan meningkatkan keandalan proses interpretasi data GPR.

Penelitian ini dilakukan untuk menjawab kebutuhan akan sistem otomatis, yang berfokus pada pengembangan aplikasi berbasis web bernama "ITERA-VISION-GPR-UI (*ITERA Vision System for GPR Utilities Inference*)" yang dirancang untuk melakukan salah satu tugas *computer vision*, yaitu deteksi objek khususnya pada citra radargram GPR untuk mengidentifikasi keberadaan utilitas bawah permukaan. Penelitian ini bertujuan untuk mempermudah *geophysicist* dalam melakukan interpretasi yang cepat dan efisien khususnya pada metode GPR, serta menguji kelayakan model YOLOv12 yang digunakan dalam pelatihan dataset sebagai *inputan* model pada aplikasi ITERA-VISION-GPR-UI.

2. METODE

2.1 Metode *Ground Penetrating Radar* (GPR)

Metode GPR bekerja dengan memancarkan gelombang elektromagnetik (EM) yang merambat melalui lapisan dangkal bawah permukaan dan bereaksi terhadap variasi sifat elektromagnetik medium yang dilaluinya. Parameter permitivitas dielektrik material merupakan faktor utama dalam mengendalikan terbentuknya sinyal refleksi (Baker dkk., 2007). Gelombang EM dipantulkan kembali ketika menemui batas antara dua lapisan material dengan sifat elektromagnetik yang berbeda, dan pantulan tersebut diterima oleh antenna penerima. Hasil pantulan yang direkam menghasilkan informasi mengenai keberadaan serta karakteristik objek di bawah permukaan, yang kemudian ditampilkan dalam bentuk radargram (Mahesya dkk., 2011).



Gambar 1. Skema Kerja GPR (Mahesya dkk., 2011)

2.2 Pre-processing GPR

Pre-processing pada data GPR merupakan tahap awal yang bertujuan untuk memperbaiki serta meningkatkan mutu sinyal radar mentah yang diperoleh dari akuisisi lapangan. Tahapan ini memiliki peran penting dalam meningkatkan rasio *signal-to-noise* (SNR) sehingga proses identifikasi dan interpretasi fitur bawah permukaan dapat dilakukan dengan lebih tepat.

Mengacu pada buku *Reflexw* karya Sandmeier (1998) Umumnya *pre-processing* data GPR memiliki tahapan diantaranya:

1. *Staticcorrection (moving starttime)*.

Koreksi untuk menyamakan titik waktu awal sinyal refleksi pada setiap trace agar permukaan tanah atau antena memiliki acuan waktu yang konsisten.

2. *Dewow*

Menghilangkan komponen frekuensi sangat rendah atau bias DC (*baseline drift*) yang berasal dari sistem antena atau pengaruh permukaan.

3. *Gain*

Gain berfungsi untuk menghasilkan amplitude yang didistribusikan secara merata secara vertikal, biasanya dalam domain waktu.

4. *Filter Band-pass* Frekuensi

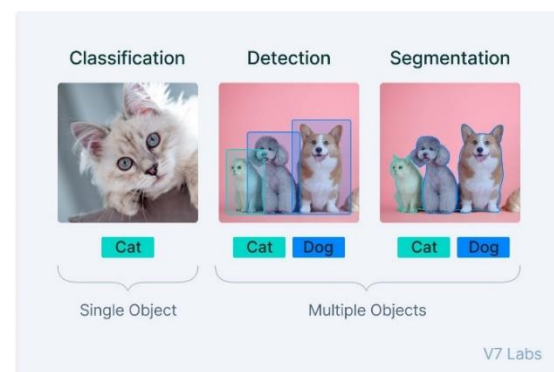
Penyaringan frekuensi untuk menghilangkan noise frekuensi sangat tinggi atau sangat rendah dengan menentukan 4 nilai frekuensi, frekuensi *low-cut*, awal *plateau*, akhir *plateau*, dan frekuensi *high-cut*.

5. *Running Average*

Perataan atau penghalusan sinyal melalui rata-rata dari beberapa *trace* berdekatan yang dapat dipilih untuk setiap langkah waktu (*time step*) yang digunakan untuk mengurangi fluktuasi acak dan memperjelas pola refleksi.

2.3 Computer Vision

Computer vision merupakan bidang ilmu yang mempelajari cara menginterpretasi dan memahami informasi dari gambar maupun video, sehingga komputer dapat melakukan analisis layaknya kemampuan visual manusia (Umam & Negara, 2016), Dengan kata lain, *computer vision* berusaha meniru cara kerja sistem penglihatan manusia.



Gambar 2. Gambaran dari tugas *Computer Vision* versi v7 (Bandyopadhyay Hmrishav, 2022)

Bidang ini memiliki hubungan erat dengan *image processing* (pengolahan citra) dan *machine vision* (visi mesin), sebab ketiganya menggunakan banyak metode dasar yang serupa serta diaplikasikan pada kasus-kasus yang hampir sama. Lebih jauh lagi, *computer vision* juga bersinggungan dengan berbagai disiplin ilmu lain seperti kecerdasan buatan (*artificial intelligence/ AI*), robotika, otomasi industri, pengolahan sinyal, optika fisik, hingga neurobiology (Dompeipen dkk., 2021).

Saat ini, penerapan *computer vision* semakin meluas dalam kehidupan sehari-hari. Beberapa contoh adalah pada sistem deteksi wajah (*face detection*), pengenalan ekspresi wajah (*facial expression recognition*), dan biasanya dikombinasikan dengan jaringan syaraf tiruan (*artificial neural network*) agar hasilnya lebih akurat (Dompeipen dkk., 2021).

Deteksi objek merupakan salah satu tugas dari *computer vision* digunakan untuk mengidentifikasi serta menentukan posisi objek tertentu di dalam sebuah citra digital. Ketika seseorang melihat sebuah gambar, otaknya secara otomatis mampu mengenali berbagai objek, menentukan posisinya, serta memahami hubungan atau interaksi antar objek tersebut. Sistem penglihatan manusia yang efisien dan presisi memungkinkan kita untuk menyelesaikan beragam tugas kompleks dengan cepat. Namun, jika kemampuan ini ingin diterapkan pada komputer, diperlukan algoritma yang memiliki kecepatan dan ketepatan serupa agar mesin mampu meniru proses

pengenalan visual manusia dan menangani tugas-tugas umum secara efektif (Dompeipen dkk., 2021).

2.4 Arsitektur YOLOv12

Model *You Only Look Once* (YOLO) merupakan arsitektur deteksi objek yang berbasis pada *Convolutional Neural Network* (CNN). YOLO mampu mendeteksi seluruh objek yang telah terdefinisi dan membuat kotak pembatas (*bounding box*) di sekelilingnya. YOLO memanfaatkan kemampuan CNN untuk memperkirakan posisi kotak pembatas serta menghitung probabilitas kelas dari setiap objek yang terdapat pada citra. Karena algoritma ini hanya memerlukan satu kali proses pengamatan untuk mengenali objek beserta lokasinya, maka dinamakan *You Only Look Once* (Redmon dkk., 2016).

Menurut Asshiddiqie (2020) terdapat tiga tahapan YOLO dalam mendeteksi suatu objek. Pada roses awal, citra dibagi menjadi beberapa *grid* berukuran $s \times s$ untuk keperluan proses deteksi objek. Setiap *bounding box* melakukan prediksi terhadap masing-masing *grid* dan nilai *confidence*. Nilai *confidence* ini merepresentasikan tingkat keyakinan bahwa *bounding box* tersebut benar-benar mengandung objek sesuai dengan perencanaan dan akurasi prediksi. Persamaan matematis dari nilai *confidence* dapat dituliskan pada Persamaan (1):

$$\text{conf}(\text{class}) = \Pr(\text{Class}) \times \text{IOU} \frac{\text{Truth}}{\text{Pred}} \quad (1)$$

dengan:

$\text{conf}(\text{class})$: *Confidence score*

$\Pr(\text{Class})$: Probabilitas prediksi kelas

$\text{IOU} \frac{\text{Truth}}{\text{Pred}}$: *Intersection over Union* rasio tingkat tumpang tindih antara kotak prediksi dengan kotak *ground truth*

Semakin besar nilai IOU yang dihasilkan, maka semakin tinggi pula tingkat akurasi dalam mendeteksi objek. Persamaan matematis dari perhitungan IOU dapat dituliskan pada Persamaan (2):

$$\text{IOU} \frac{\text{Truth}}{\text{Pred}} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2)$$

Proses kedua setiap *bounding box* memiliki lima variable yaitu x , y , w , h , dan c . x dan y merupakan nilai koordinat dari titik tengah *bounding box* objek yang terdeteksi. w dan h adalah nilai dari ukuran lebar dan tinggi. c adalah *confidence* dari *bounding box*. Proses ketiga Setiap *grid* melakukan prediksi terhadap nilai probabilitas kelas ketika terdapat objek di dalamnya. Nilai probabilitas kelas tersebut dikalikan

dengan nilai *confidence* dari *bounding box*, sehingga dihasilkan nilai *confidence* akhir untuk setiap *bounding box* pada masing-masing kelas secara spesifik, sebagaimana ditunjukkan pada Persamaan (3):

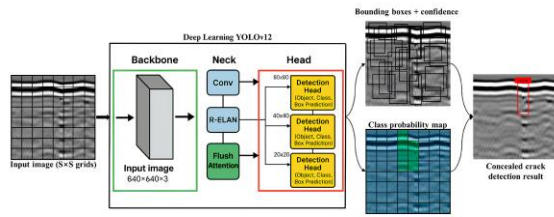
$$\begin{aligned} & \Pr(\text{Class}_i | \text{Object}) \times \Pr(\text{Object}) \\ & \times \text{IOU} \frac{\text{Truth}}{\text{Pred}} \\ & = \Pr(\text{Class}_i) \\ & \times \text{IOU} \frac{\text{Truth}}{\text{Pred}} \end{aligned} \quad (3)$$

dengan:

$\Pr(\text{Object})$: Probabilitas prediksi objek

YOLOv12 adalah versi terbaru dari keluarga YOLO yang merupakan terobosan terbaru dalam evolusi teknologi deteksi objek, yang membawa perubahan arsitektur besar dengan tujuan merevolusi sistem deteksi objek secara *real-time* (Yunji Tian, dkk 2025). YOLOv12 memiliki 3 komponen struktural utama diantaranya adalah *Backbone*, *Neck*, dan *Head*. *Backbone* berperan penting dalam mentransformasikan data citra mentah menjadi *feature map* multiskala yang menjadi dasar untuk proses deteksi berikutnya. Komponen inti dari bagian ini adalah *Residual Efficient Layer Aggregation Network* (R-ELAN), yang mengintegrasikan lapisan konvolusional dalam dengan koneksi residual yang dirancang secara strategis. Struktur ini mampu mengatasi masalah *gradient vanishing* sekaligus memaksimalkan pemanfaatan kembali fitur, sehingga memperkuat kemampuan model dalam mengenali detail objek yang kompleks dengan berbagai ukuran dan bentuk (Rabbani & Hussain, 2025).

Bagian *Neck* berfungsi sebagai penghubung antara *backbone* dan *head*, yang bertugas menggabungkan serta memperhalus fitur multiskala. Salah satu inovasi utamanya adalah penerapan mekanisme *area attention* yang dipercepat menggunakan *FlashAttention*, memungkinkan model lebih fokus pada area penting dalam gambar yang memiliki banyak elemen atau gangguan visual. Sementara itu, bagian *Head* pada YOLOv12 bertanggung jawab untuk mengubah peta fitur hasil proses *neck* menjadi keluaran akhir berupa koordinat *bounding box* dan skor klasifikasi. Peningkatan yang dilakukan mencakup penyederhanaan jalur deteksi multiskala serta penerapan fungsi *loss* yang dioptimalkan untuk menyeimbangkan antara akurasi lokalisasi dan klasifikasi secara lebih efektif. (Rabbani & Hussain, 2025). Ilustrasi cara YOLOv12 bekerja ditampilkan pada Gambar 3. dengan input adalah citra radargram dan output berupa citra radargram yang sudah di lakukan prediksi oleh Liu dkk., (2021).



Gambar 3. Deteksi retakan tersembunyi pada perkerasan aspal berbasis citra B-scan GPR menggunakan model YOLOv12 (Liu dkk., 2021)

Dalam YOLO atau YOLOv12 terdapat *loss function* yang digunakan untuk melakukan analisis hasil pelatihan pada kinerja model, dengan dibutuhkannya data hasil dari model evaluasi yaitu *confusion matrix*. proses ini digunakan untuk menghitung tingkat akurasi sistem, guna menilai efektivitas dan performa model dalam mendeteksi objek secara keseluruhan. Dalam melakukan pengukuran tingkat akurasi dibutuhkan beberapa parameter diantaranya adalah (Dompeipen dkk., 2021):

1. *True Positive* (TP) menggambarkan kondisi ketika sistem berhasil mengenali objek target dengan benar.
2. *False Positive* (FP) adalah situasi di mana sistem secara keliru mengidentifikasi objek lain sebagai target.
3. *False Negative* (FN) adalah kondisi ketika objek yang sebenarnya ada tidak terdeteksi oleh sistem.
4. *True Negative* (TN) adalah jumlah sampel yang memang bukan termasuk dalam kelas target dan secara tepat diklasifikasikan sebagai bukan kelas target oleh model (Chicco & Jurman, 2020).
5. *Recall* adalah rasio antara jumlah prediksi positif yang benar terhadap total keseluruhan data yang seharusnya positif, termasuk data yang tidak berhasil terdeteksi oleh model. *Recall* dapat dilihat melalui persamaan (4):

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

6. *Precision* mengukur tingkat ketepatan prediksi model terhadap objek yang benar terdeteksi dibandingkan dengan total dari prediksi positif. *Precision* dapat dilihat melalui persamaan (5):

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

7. *Mean Average Precision* (mAP) merupakan *matrix* utama yang digunakan untuk mengukur kinerja deteksi objek pada model, termasuk YOLOv12. *Average Precision* (AP) adalah luas area di bawah kurva PR untuk satu kelas objek.

$$AP = \int_0^1 P(R) dR \quad (6)$$

Kemudian, mAP didapat dengan mengambil rata-rata nilai AP yang ada dari semua kelas objek:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (7)$$

dengan:

N : Jumlah kelas

8. *F1 Score* adalah nilai yang didapat untuk menunjukkan performa algoritma yang diterapkan. Rumus dari perhitungan f-score dapat dilihat melalui persamaan (8):

$$F1 \text{ Score} = \frac{2x (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (8)$$

9. Akurasi menurut Chicco, D. (2020) adalah nilai yang menunjukkan perbandingan antara jumlah prediksi yang benar (baik positif maupun negatif) terhadap total keseluruhan data yang diuji. Akurasi dapat dilihat melalui persamaan (9):

$$Akurasi = \frac{TP + TN}{TP + FP + FN + TN} \quad (9)$$

2.5 Prosedur Penelitian

Penelitian ini dilakukan mulai dari bulan Juli – Oktober 2025 yang berlokasi di Laboratorium Pengolahan Data Geofisika, Gedung Laboratorium Teknik 3 Institut Teknologi Sumatera (ITERA). Penelitian dan pengambilan data dilakukan menggunakan citra radargram dalam format PNG/JPEG/JPG dari data GPR yang sudah diolah. Bahan dan alat dalam penelitian ini adalah *device* laptop HP *Elite Book* 820 G4, citra radargram GPR, ReflexW, bahasa pemrograman *python* v3.10.11, *Visual Studio Code* (VSCode), *Roboflow*, *Google Colaboratory*, dan *Streamlit*. Metode dan prosedur penelitian, secara lengkap akan dijelaskan dalam beberapa poin yang berhubungan dengan proses pengerjaan.

1. Pre-processing Data GPR

Sebelum dilakukan proses *computer vision*, tahap awal yang harus dilakukan adalah *pre-processing* data GPR guna meningkatkan kualitas citra radargram dan mengurangi gangguan sinyal. Beberapa *filter* yang diterapkan adalah *static correction* (*moving start time*), *filter dewow* dengan *timewindow* 4 ns, gain *Automatic Gain Control* (AGC) dengan *window length* sebesar 3, *filter bandpass frequency* dengan *lower cutoff* 180 hz, *lower plateau* 300 hz, *upper plateau* 450 hz, *upper cutoff* 600 hz, *filter running average* dengan parameter yang digunakan adalah rata-rata dari 3

trace. Parameter yang sama diterapkan keseluruhan data. Penulis memiliki 6 lintasan yang diolah untuk dijadikan sebagai dataset, pada citra lain gambar radargram diekspor menggunakan *software ReflexW* dari berbagai sumber, serta menambah dataset melalui data *opensource*.

2. Dataset

Dataset merupakan kumpulan data yang tersusun secara terstruktur dan digunakan sebagai dasar dalam proses pelatihan, validasi, maupun pengujian model pembelajaran mesin. Dataset dapat berisi berbagai jenis data seperti, teks, gambar, suara atau angka yang telah dilabeli. Pada penelitian ini dataset berupa citra radargram dengan ukuran pixel yang berbeda-beda dengan total jumlah 82 gambar, 62 gambar untuk pelatihan, 12 gambar validasi, 8 gambar untuk pengujian + 2 gambar untuk inferensi di luar dataset. 82 gambar untuk *train/val/test* dalam format PNG/JPG/JPEG diinput pada *tools Roboflow*.

3. Labelling

Proses *labelling* adalah tahap pemberian label atau anotasi pada setiap objek yang muncul di dalam citra. Tujuan utamanya adalah mengaitkan setiap objek dengan kelas tertentu sekaligus menentukan koordinat objek tersebut di dalam gambar. Pada sistem deteksi objek, hasil proses ini berupa *bounding box* yang menggambarkan letak dan dimensi objek yang akan dikenali oleh model (Redmon & Farhadi, 2018).

Menurut Redmon, J., & Farhadi, A. (2018), setiap *bounding box* direpresentasikan oleh lima komponen utama:

$$(x, y, w, h, c) \tag{10}$$

dengan:

- x : Koordinat titik tengah kotak pembatas pada sumbu horizontal
- y : Koordinat titik tengah pada sumbu vertikal
- w : Lebar *bounding box*
- h : Tinggi *bounding box*
- c : Kelas atau label objek.

Dalam arsitektur YOLO, koordinat ini dinormalisasi terhadap dimensi gambar agar memiliki skala yang seragam, dengan persamaan (11):

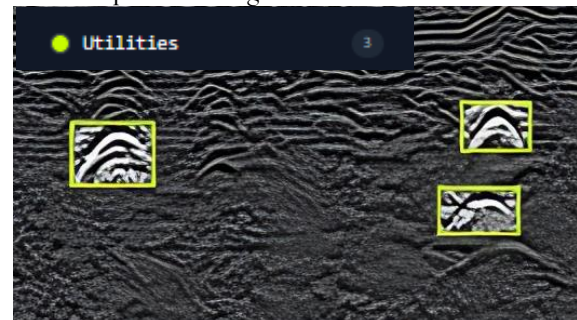
$$x_n = \frac{x}{W}, y_n = \frac{y}{H}, w_n = \frac{w}{W}, h_n = \frac{h}{H} \tag{11}$$

dengan:

- W : Lebar *bounding box*
- H : Tinggi *bounding box*.

Setiap hasil *labelling* disimpan dalam format .txt, berisi nilai (c, x_n, y_n, w_n, h_n) untuk tiap objek pada gambar. Informasi ini digunakan oleh model YOLO dalam tahap pelatihan untuk mengenali dan mendeteksi objek sesuai dengan kelas yang telah ditentukan.

Pada penelitian ini, proses *labelling* dilakukan menggunakan *platform Roboflow*, dengan hanya satu kelas target yaitu "*utilities*", yang didefinisikan sebagai objek dengan pola refleksi hiperbolik pada citra radargram. Tahapan *labelling* dimulai dengan pemilihan fitur anotasi citra pada dataset yang telah disiapkan. Setiap pola hiperbolik pada radargram diberi *bounding box* dengan referensi koordinat (x, y, w, h) , batas atas (*upper bound*) dari parameter h diletakkan tepat di atas puncak hiperbolik, pada batas bawah (*lower bound*) mencakup seluruh bagian buntut hiperbolik, sementara batas kanan dan kiri (*width, w*) ditempatkan sedemikian rupa agar menutupi keseluruhan lebar pola hiperbolik secara presisi. Pendekatan ini memastikan bahwa setiap area anotasi mencakup pola refleksi utilitas secara utuh, sehingga model dapat belajar mengenali karakteristik bentuk hiperbolik dengan maksimal.



Gambar 4. Tampilan proses *labelling* data GPR dengan klasifikasi 1 kelas "*Utilities*"

4. Splitting

Splitting merupakan proses pembagian dataset atau citra menjadi bagian yang terpisah biasanya adalah 3 bagian, seperti pelatihan, validasi, dan pengujian model. Tujuannya adalah untuk memastikan bahwa model belajar dari satu subset lalu dievaluasi oleh subset lain yang belum pernah terlihat, agar hasil model dapat belajar dengan baik dan dapat dievaluasi. Pada penelitian ini *splitting* dilakukan menggunakan *platform Roboflow* dengan subset yang terdiri dari 75% pelatihan, 15% validasi, dan 10% pengujian dengan rincian seperti pada Table 1.

Table 1. Rincian *splitting* dari seluruh dataset

Pelatihan	Validasi	Pengujian
62	12	8

5. Pelatihan Model YOLOv12

Tujuan utama dari pelatihan model adalah untuk membuat model YOLOv12 mampu mengenali dan mendeteksi objek baru secara *real-time* dengan tingkat akurasi yang tinggi, bahkan pada data yang belum pernah dilihat sebelumnya. Proses yang terjadi pada saat model dilatih adalah gambar dan label berupa koordinat objek beserta kelasnya di input, model menghitung posisi dan kelas objek menggunakan komponen struktur utama YOLOv12 (*Backbone*, *neck*, dan *head*), perhitungan *loss* dengan hasil prediksi dibandingkan dengan label sebenarnya menggunakan fungsi *loss* gabungan yang terdiri dari klasifikasi *loss*, lokalisasi *loss*, dan objek *loss*, selanjutnya model melakukan *backpropagation* untuk memperbarui bobot agar *loss* semakin kecil, lalu iterasi (*epoch*). pada proses ini iterasi dilakukan berulang kali agar model bisa belajar dengan baik, yaitu ketika akurasi terus meningkat dan *loss* stabil.

Pelatihan pada penelitian ini menggunakan *Google Colaboratory*. Jenis model yang digunakan adalah YOLOv12s, dengan “s” diartikan sebagai ukuran varian *small* dalam keluarga YOLOv12 yang dirancang untuk menyeimbangkan kecepatan inferensi tinggi dengan akurasi deteksi yang baik. Parameter utama dalam pelatihan (proses *Hyperparameter Tuning*) diantaranya adalah *batch*=-1, *epochs*=500, *optimizer*=AdamW, input *imgsz* 640 *pixel* (640 x 640 x 3), dan *patience*=100. Hasil pelatihan akan memberikan output berupa *project* baru yaitu model yang sudah selesai dilakukan pelatihan dalam bentuk *best.pt*.

6. Model Evaluasi dan Inferensi

Model evaluasi (*matrix* evaluasi) pada YOLOv12 berfokus mengukur seberapa baik model dalam mendeteksi objek, yaitu lokalisasi (*bounding box*) dan klasifikasi (kelas objek) serta performanya secara keseluruhan. Beberapa *matrix* utama yang sering digunakan adalah *Precision* yang merupakan proporsi deteksi yang benar dari keseluruhan deteksi (hindari banyak *false positives*). *Recall* adalah proporsi objek nyata yang berhasil dideteksi dari keseluruhan objek yang ada (hindari banyak *false negatives*). *Mean Average Precision* (mAP) adalah rata-rata dari *Average Precision* (AP) di berbagai *Intersection Over Union* (IoU) *threshold*. Contoh: mAP50 (IoU $\geq 0,50$) atau mAP50-95 (IoU dari 0,50 hingga 0,95). AP sendiri adalah area di bawah kurva *precision-recall* untuk satu kelas (Tian dkk., 2025).

Matrix evaluasi didapat ketika pelatihan model selesai, Apabila hasil *confusion matrix* menunjukkan performa yang sesuai dengan kriteria yang telah

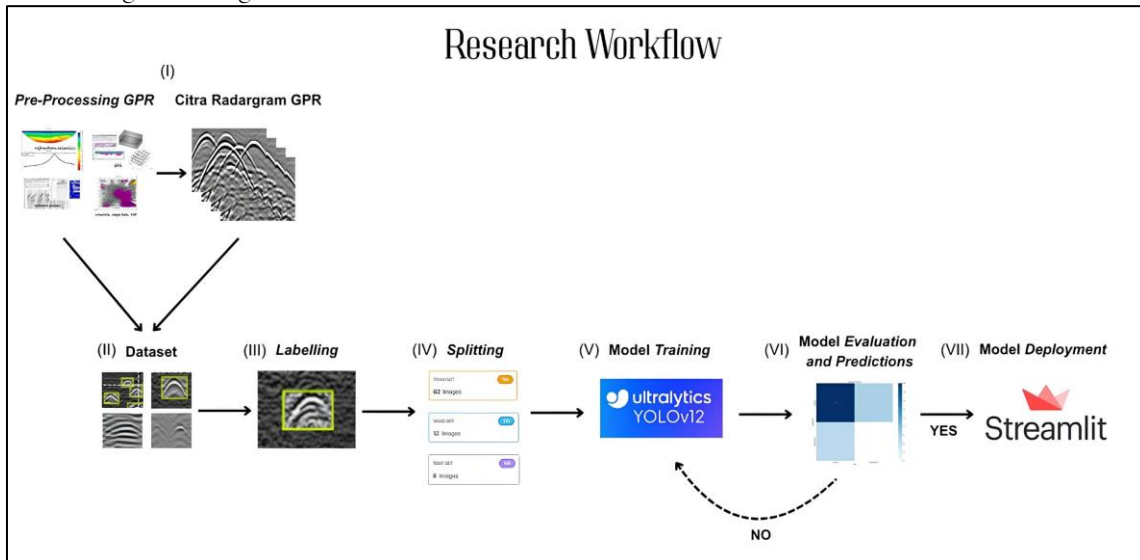
ditetapkan, maka model terbaik (*best.pt*) yang dihasilkan akan diunduh untuk digunakan pada tahap *inference* dan diintegrasikan ke dalam aplikasi ITERA-VISION-GPR-UI. Namun, apabila hasil evaluasi belum memenuhi harapan, proses pelatihan akan diulang dengan melakukan *tuning* terhadap parameter-parameter pelatihan hingga diperoleh model dengan performa yang maksimal. Untuk memastikan apakah model sudah mendapatkan hasil terbaik dilakukan inferensi model menggunakan citra radargram diluar dari *train/val/test set* dengan tujuan memastikan apakah model layak untuk digunakan. Inferensi merupakan fase saat model yang sudah dilatih digunakan untuk mendeteksi dan mengenali objek pada gambar atau video baru (Tian dkk., 2025). Pada tahap ini kembali menuju proses *labelling* menjadi peran yang cukup penting ketika performa model masih belum maksimal setelah dilakukan pelatihan berulang dengan parameter yang berbeda, karena pada penelitian ini acuan utama dalam menilai hasil model sudah cukup baik adalah *matrix* utama yakni nilai mAP 50, mAP50–95, *Recall*, dan *Precision*.

7. Deployment aplikasi menggunakan Streamlit dan inferensi model.

Streamlit adalah pustaka *Python* *opensource* yang memfasilitasi pembuatan aplikasi berbasis web apps yang bisa diimplementasikan untuk pembelajaran mesin dan ilmu data. *Streamlit* menyediakan berbagai *widget* bawaan untuk masukan pengguna seperti pengunggahan gambar, penggeser, masukan teks, dan elemen *hypertext markup language* (HTML) lain yang sudah dikenal seperti *checkboxes* dan *radio buttons*. *Streamlit* merupakan aplikasi gratis yang tidak perlu memiliki pengetahuan pengembangan *front-end* web yang mahir untuk pengoperasiannya (Prapas dkk., 2021)

Pada penelitian ini *deployment* dilakukan di *localhost* dengan menginstall beberapa pustaka penting seperti *Streamlit*, *Ultralytics*, *Pytorch*, *supervision*, *numpy*, dan *matplotlib* untuk kebutuhan menjalankan aplikasi serta melakukan inferensi. *Script* dibuat menyesuaikan kebutuhan penulis dengan outputnya nanti adalah pemilihan model, input gambar by PNG/JPG/JPEG dengan banyak gambar, deteksi objek dengan range *confidence score* 0–100, hasil deteksi disertai *confidence score* pada setiap objek dan setiap gambar yang diinput, menambahkan nama aplikasi, Input path model *best.pt* yang sudah dilakukan pelatihan kedalam *script*. Jika sudah sesuai, menjalankan aplikasi adalah langkah akhir dengan perintah “*streamlit run* (nama *file*).py” agar aplikasi dapat dijalankan dan ditampilkan pada *localhost device*. Inferensi dilakukan pada aplikasi yang sudah

dijalankan dengan data citra terbaru diluar *train/val/test* guna mengetahui hasil deteksi dan performa aplikasi.



Gambar 5. Alur Kerja Penelitian

3. HASIL DAN PEMBAHASAN

3.1. Hasil Pelatihan dan Model Evaluasi YOLOv12

Pada bagian ini dipaparkan hasil pelatihan model YOLOv12 beserta analisis performanya selama proses pelatihan. Evaluasi dilakukan untuk menilai sejauh mana model mampu mempelajari karakteristik pola hiperbolik pada citra radargram. Pada Gambar 6. ditunjukkan perubahan nilai *matrix* pada setiap *epoch*. Sumbu x merepresentasikan jumlah *epoch*, sedangkan sumbu y menggambarkan performa *matrix*. Nilai *matrix* yang mendekati 1 menandakan bahwa model semakin jarang melakukan kesalahan deteksi (*false positive*). Model YOLOv12 pada penelitian ini menggunakan empat *matrix* utama, yaitu *Precision*, *Recall*, *mAP50*, dan *mAP50–95* selama proses pelatihan hingga mencapai 392 *epoch*. Setiap grafik menggambarkan peningkatan performa model seiring bertambahnya iterasi pelatihan. Model awal dirancang untuk berlatih hingga 500 *epoch*, namun proses pelatihan berhenti otomatis pada *epoch* ke-392. Hal ini disebabkan oleh pengaturan parameter *tuning patience* sebesar 100, yang berarti pelatihan dihentikan lebih awal yang biasa disebut *early stopping* jika tidak ada peningkatan dalam performa validasi selama 100 *epoch* berturut-turut. Dari hasil evaluasi, performa terbaik (*best model weights*) diperoleh pada *epoch* ke-292, yang kemudian digunakan sebagai dasar dalam analisis *confusion matrix*. *Early stopping* ini menunjukkan bahwa model telah mencapai titik stabil dan tidak mengalami

peningkatan pada *matrix* evaluasi setelah *epoch* tersebut, sehingga menghindari terjadinya risiko *overfitting*.

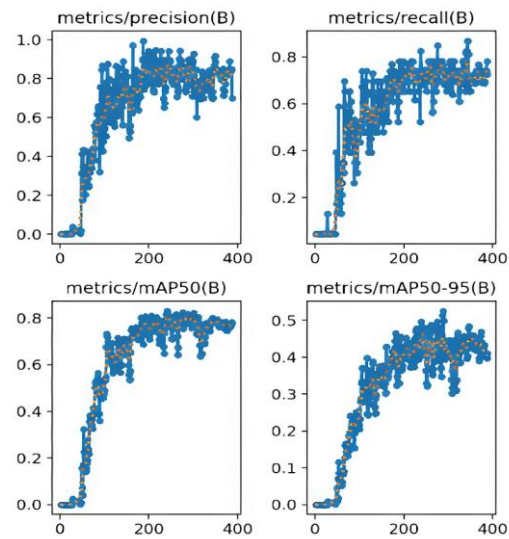
Berdasarkan hasil pelatihan model YOLOv12 diperoleh hasil *confusion matrix* yang menggambarkan performa model. Pada Gambar 7. *confusion matrix* dihitung menggunakan dataset validasi dan pengujian. Pada *epoch* ke-292 diketahui sebanyak 18 citra utilitas yang merupakan TP berhasil terdeteksi dengan benar sebagai *Utilities* oleh model, sebanyak 3 citra FP yang seharusnya merupakan *background* justru terdeteksi sebagai *Utilities*, dan sebanyak 5 citra FN yang seharusnya termasuk *Utilities*, tidak terdeteksi oleh model dan diklasifikasikan sebagai *background*. Model ini menunjukkan kemampuan deteksi yang cukup baik dengan jumlah TP(18) yang dominan dibandingkan kesalahan prediksi. Namun, masih terdapat beberapa kesalahan klasifikasi baik dalam bentuk FP (3) maupun FN (5), yang menandakan bahwa model terkadang masih keliru dalam membedakan antara objek utilitas dan latar belakang (*background*). Pada gambar 6. tren grafik perubahan *matrix per-epoch* untuk *matrix Precision* memperlihatkan peningkatan tajam pada awal pelatihan hingga mencapai kestabilan sekitar 0,8 setelah *epoch* ke-200. Sehingga pada akhirnya model mampu mencapai 0,85 pada *epoch* ke-292. Nilai ini memperlihatkan bahwa model mampu melakukan deteksi utilitas dengan cukup baik pada tingkat kesalahan *false positive* yang rendah sebagian besar deteksi benar-benar merupakan objek utilitas yang sesungguhnya.

Pada nilai *Recall*, grafik menunjukkan nilai 0,77 pada *epoch* ke-292 walau sempat menyentuh titik tertinggi sebesar 0,85 disekitar *epoch* 300-392. Hal ini dikarenakan keraguan model dalam memprediksi TP dan FN pada beberapa citra. Nilai *recall* yang tinggi menandakan bahwa model memiliki sensitivitas yang baik dalam mengenali keberadaan objek utilitas di bawah permukaan tanpa banyak kehilangan objek yang seharusnya terdeteksi (*false negative* rendah).

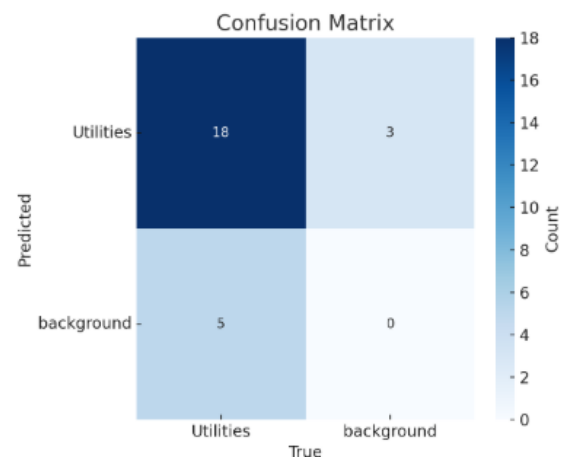
Matrix mAP50 menunjukkan nilai yang cukup tinggi sebesar 0,82 pada *epoch* ke-292. Tren grafik perubahan *matrix* per-*epoch* pada *matrix* ini memperlihatkan peningkatan pada awal pelatihan hingga mencapai kestabilan yang berkelanjutan. Nilai tersebut menandakan bahwa model memiliki akurasi yang cukup tinggi dalam mendeteksi posisi serta bentuk objek utilitas dengan tingkat tumpang tindih (*overlap*) yang memadai antara prediksi dan label sebenarnya, dikarenakan ambang IoU pada mAP50 memiliki batas toleransi yang cukup longgar dengan minimal 0,50 oleh sebab itu nilainya relatif tinggi.

Pada *matrix* mAP50-95 nilainya relatif lebih rendah, yaitu sekitar 0,51 untuk *epoch* ke-292, namun tren kenaikannya stabil hingga mencapai konvergensi. *Matrix* ini menunjukkan kemampuan model dalam mempertahankan akurasi deteksi pada tingkat ketelitian yang tinggi oleh sebab itu nilainya relatif lebih rendah karena dihitung pada banyak ambang IoU. *F1 Score* dan akurasi jika dihitung menurut persamaan (8) dan persamaan (9) masing-masing memiliki nilai 0,8 dan 0,69. Hal ini menandakan bahwa model memiliki kemampuan yang cukup baik dalam mengidentifikasi objek utilitas dengan tingkat kesalahan prediksi yang relatif rendah. Nilai akurasi sebesar 0,69 mengindikasikan bahwa sebagian besar prediksi yang dilakukan model berada pada kategori yang benar.

Hasil dari nilai *matrix* utama dan perhitungan akurasi mengindikasikan bahwa model mampu mendeteksi utilitas secara efisien sehingga model layak digunakan untuk inferensi pada aplikasi yang dibangun, walaupun model ini masih jauh dari sempurna. Hal ini dapat dikaitkan dengan penelitian sebelumnya yang dilakukan oleh Hou dkk., (2025), bahwa nilai *matrix* yang tinggi tidak selalu menjamin bahwa model bisa memprediksi semua pola hiperbolik terutama pada target yang tumpang tindih dan pola yang tidak terlihat dengan jelas.



Gambar 5. Perubahan *matrix* per-*epoch*



Gambar 6. *Confusion matrix*

3.2. Aplikasi Deteksi Objek “ITERA-VISION-GPR-UI (ITERA Vision System for GPR Utilities Inference)” dan Hasil Inferensi

Aplikasi “ITERA-VISION-GPR-UI” dijalankan pada *localhost device* untuk melakukan inferensi dan menguji model pada data yang benar-benar baru. *Localhost device* membuat aplikasi ini hanya memungkinkan untuk dijalankan pada *device* penulis. Gambar 8. merupakan tampilan dari “ITERA-VISION-GPR-UI”. Aplikasi ini memiliki beberapa fitur yang bisa dibidang *using friendly* dan visualisasi yang cukup menarik, fitur dan penjelasan mengenai visualisasi tersebut meliputi:

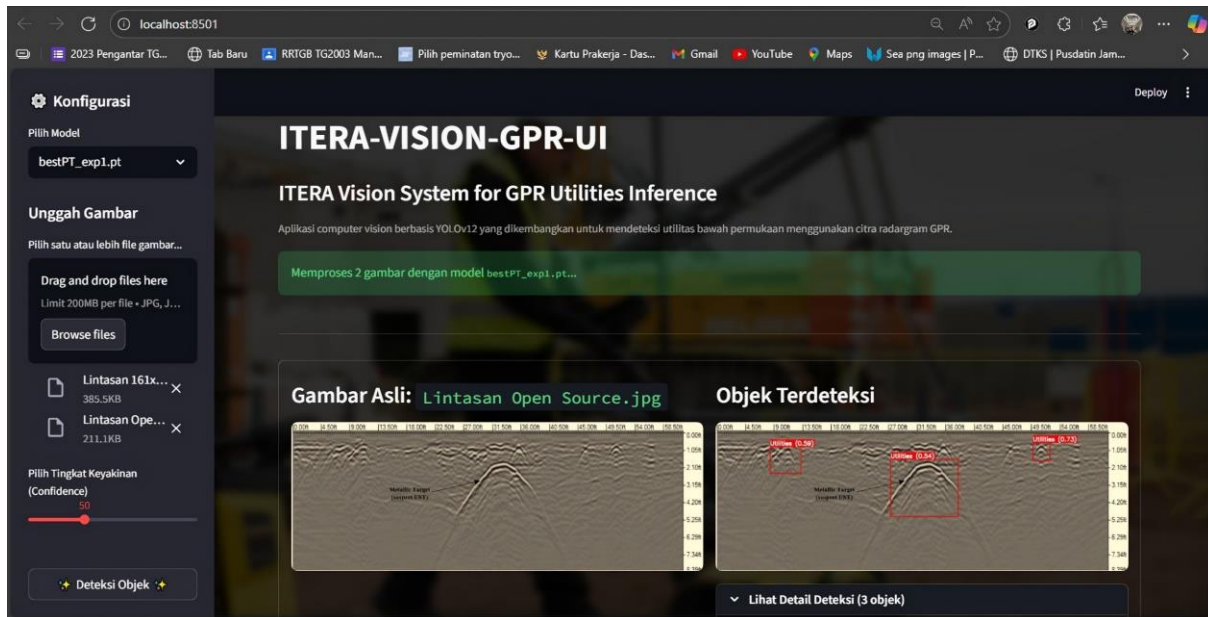
1. Header Utama

Menampilkan nama aplikasi “ITERA-VISION-GPR-UI: ITERA Vision System for GPR Utilities Inferensi” dan dibawahnya terdapat deskripsi singkat bahwa aplikasi ini merupakan sistem deteksi berbasis *computer vision* dengan menggunakan model YOLOv12.

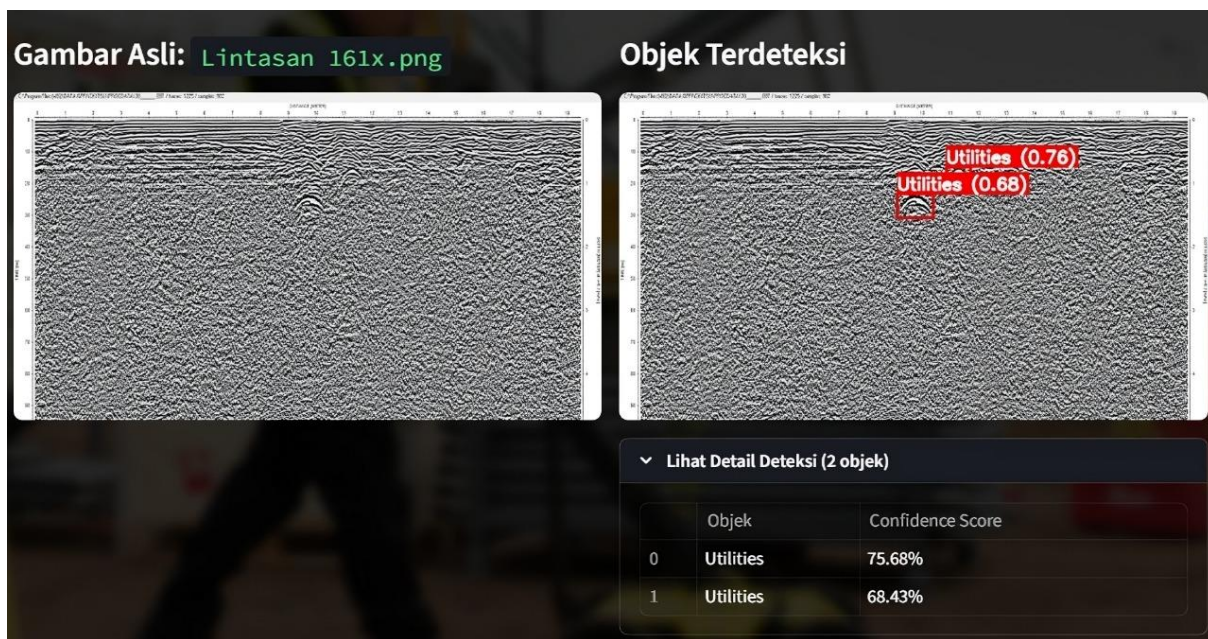
2. Menu Tampilan Model dan Unggah Gambar
Terletak dibagian panel konfigurasi sebelah kiri aplikasi. Fitur ini memungkinkan pengguna untuk memilih model hasil pelatihan YOLOv12, karena model yang *confidence* hanya satu maka kita memilih model “*bestPT_exp1.pt*”. Sedangkan pada fitur unggah gambar, memungkinkan pengguna untuk mengunggah citra GPR dari perangkat lokal (format umum: .JPG/JPEG/PNG). Fitur unggah gambar memungkinkan pengguna untuk mengunggah citra radargram sebanyak mungkin tanpa batasan jumlah kecuali pada ukuran file sebesar 200 mb.
3. Tombol “Deteksi Objek” dan Tingkat *Confidence*
Berkfungsi untuk memulai proses inferensi model YOLOv12 terhadap citra yang telah diunggah dengan fitur pengaturan seberapa besar tingkat keyakinan sistem kepada objek utilitas yang terdeteksi. Saat tombol ditekan, sistem menjalankan proses prediksi secara otomatis. Output dari proses ini berupa *bounding box* dan label klasifikasi 1 kelas yakni “*utilities*” pada area citra yang terdeteksi sebagai utilitas.
4. Kolom “Gambar Asli”
Menampilkan citra GPR mentah yang diunggah oleh pengguna. Berkfungsi sebagai pembandingan terhadap hasil deteksi agar pengguna dapat melihat perubahan visual antara citra yang diunggah dengan hasil inferensi model.
5. Kolom “Objek Terdeteksi”
Menampilkan citra hasil deteksi objek dari model YOLOv12. Area yang teridentifikasi sebagai utilitas diberi *bounding box* berwarna dengan label dan tingkat kepercayaan (*confidence score*). Misalnya, label “*utilities*” dengan nilai *confidence* 0,82 menunjukkan bahwa model memperkirakan 82% kemungkinan area tersebut merupakan utilitas.

Berdasarkan hasil inferensi pada Gambar 8. terdapat tiga area utama yang terdeteksi sebagai utilitas oleh model dengan nilai *confidence score* masing-masing sebesar 0,59, 0,54, dan 0,73. Tingkat *confidence score* yang diatur adalah sebesar 0,50. Nilai *confidence score* yang diatur menunjukkan tingkat keyakinan model bahwa radargram yang *diinput* memiliki 3 utilitas meskipun terdapat 2 sampai 3 anomali hiperbolik yang tidak terdeteksi, hal ini dikarenakan pola hiperbolik pada area tersebut memiliki bentuk yang kurang *clear*, tidak terlihat jelas, dan memiliki pola sekitar yang memperlihatkan sinyal terattenuasi pada anomali yang diduga utilitas tersebut. Lolosnya beberapa pola hiperbolik dikarenakan pola untuk setiap dataset pelatihan hanya diberi *bounding box* pada anomali hiperbola yang *clear*. Sehingga model sebenarnya masih memiliki beberapa kelemahan dalam melakukan inferensi untuk pola anomali yang cukup kompleks. Pada Gambar 9. terdapat 2 utilitas terdeteksi dengan nilai *confidence score* sebesar 0,76 dan 0,68, dengan tampilan aplikasi tersebut persentase nilai *confidence* menunjukkan bahwa aplikasi dapat memproses inferensi disertai dengan tampilan yang informatif.

Hasil inferensi citra menunjukkan bahwa model mampu mendeteksi dan mengklasifikasikan utilitas bawah permukaan secara akurat dengan tingkat *confidence score* diatas 0,50. Meskipun memiliki beberapa kelemahan, aplikasi ini dikatakan cukup bisa membantu dan dapat mempermudah pekerjaan *geophysicist* dalam melakukan interpretasi yang cepat dan efisien. Aplikasi “*ITERA-VISION-GPR-UI*” yang dijalankan pada *localhost* tidak menutup kemungkinan untuk nantinya akan tersedia didalam *cloud Streamlit* dan repository *git.hub* agar bisa digunakan oleh pengguna yang membutuhkan khususnya *geophysicist* di kemudian hari.



Gambar 7. Visualisasi aplikasi ITERA-VISION-GPR-UI dan hasil inferensi menggunakan lintasan *opensource*



Gambar 8. Hasil inferensi menggunakan citra lintasan 161

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan bahwa aplikasi ITERA-VISION-GPR-UI kelak dapat mempermudah *geophysicist* dalam melakukan interpretasi data GPR khususnya dalam pemetaan utilitas, dengan fitur-fitur pada aplikasi dan proses inferensi yang efisien memungkinkan pengguna untuk mengunggah banyak citra radargram GPR dalam jumlah banyak dan melakukan inferensi, sehingga dapat membantu interpretasi dengan lebih cepat dan akurat.

Model YOLOv12 menunjukkan performa yang stabil dan cukup baik, dengan nilai Precision (0,85), Recall (0,77), mAP50 (0,82), mAP50-95 (0,51), dan hasil perhitungan akurasi sebesar 0,69 mengindikasikan bahwa model cukup layak digunakan pada aplikasi ITERA-VISION-GPR-UI. Sehingga pada hasil inferensi citra menunjukkan bahwa model mampu mendeteksi dan mengklasifikasikan utilitas bawah permukaan secara akurat dengan tingkat *confidence score* diatas 0,50, meskipun masih memiliki beberapa kekurangan.

Untuk penelitian selanjutnya, aplikasi ITERA-VISION-GPR-UI diharapkan dapat

dikembangkan lebih lanjut melalui peningkatan jumlah dataset, kualitas citra dan keragaman dataset, serta penerapan teknik data *augmentation*. Selain itu, penyempurnaan tampilan dan fitur aplikasi juga direkomendasikan agar aplikasi dapat lebih efisien dan dapat dideploy menggunakan server aps resmi.

UCAPAN TERIMAKASIH

Ucapan terima kasih kami ucapkan kepada semua yang terlibat atas penyediaan data-data yang diperlukan dalam penelitian ini, serta seluruh pihak yang membantu dalam penulisan dan pengolahan data penelitian ini.

DAFTAR PUSTAKA

- Baker, G. S., Jordan, T. E., & Pardy, J. (2007). An introduction to ground penetrating radar (GPR). *Special Paper of the Geological Society of America*, 432, 1–18. [https://doi.org/10.1130/2007.2432\(01\)](https://doi.org/10.1130/2007.2432(01))
- Bandyopadhyay Hmrishav. (2022, June 9). From self driving cars, through defect detection to medical imaging—here’s how computer vision is helping modern businesses to solve complex visual tasks. *V7*. Diakses pada 29 Januari 2025 <https://www.v7labs.com/blog/what-is-computer-vision>
- Cheng, K. (2024). A Survey of Deep Learning-Based Object Detection. *Advances in Engineering Technology Research*, 11(1), 800. <https://doi.org/10.56028/aetr.11.1.800.2024>
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1). <https://doi.org/10.1186/s12864-019-6413-7>
- Dompeipen Tresya Anjali, Sompie Sherwin R.U.A, & Najoan Meicsy E.I. (2021). Computer Vision Implementation for Detection and Counting the Number of Humans. *Jurnal Teknik Informatika*, 15(4)(Vol. 16 No. 1 (2021): Jurnal Teknik Informatika), 1–12. <https://doi.org/10.35793/jti.16.1.2021.31471>
- Hazreq bin Haron, A., Sarah binti Tengku Amran, T., Ridzuan Bin Ahmad, M., Azreen Masenwat, N., Sani, S., M., Ali bin Mohd Yusoff, M., & Aliff Daniel bin Mohsin, M. (2024). Subsurface utility detection using ground penetrating radar and electromagnetic locator – a comparative study. *IOP Conference Series: Materials Science and Engineering*, 1308(1), 012013. <https://doi.org/10.1088/1757899X/1308/1/012013>
- Liu, Z., Wu, W., Gu, X., Li, S., Wang, L., & Zhang, T. (2021). Application of Combining YOLO Models and 3D GPR Images in Road Detection and Maintenance. *Remote Sensing*, 13(6), 1081. <https://doi.org/10.3390/rs13061081>
- Mahesya, A., Karya, H. A., Asri, S., Bagus, W., Christian, H., Fitri, S., Hardeka, W., Lia, P., Khairum, T., Ma’arif, S., Pratama, W., & Sialagan, F. (2011). *MODUL EKSPLORASI ELEKTROMAGNETIK Teknik Geofisika 2011 Universtas Lampung*. https://www.academia.edu/39804469/Eksplorasi_elektromagnetik?auto=download
- Rabbani Alif Mujadded Al, & Hussain, M. (2025). *YOLOV12: A BREAKDOWN OF THE KEY ARCHITECTURAL FEATURES*. <https://arxiv.org/html/2502.14740v1>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*.
- Tian, Y., Ye, Q., & Doermann, D. (2025). *YOLOv12: Attention-Centric Real-Time Object Detectors*. <https://arxiv.org/abs/2502.12524v1>
- Umam, K., & Negara, B. S. (2016). Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, 2(2), 31. <https://doi.org/10.24014/coreit.v2i2.2391>