IMPLEMENTASI MODEL YOLOV7 DAN MASK R-CNN DALAM INSTANCE SEGMENTATION CITRA GUNUNG BERAPI



Disusun Oleh:

N a m a : Resa Zulfikar
NIM : 18523030

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2023

HALAMAN PENGESAHAN DOSEN PEMBIMBING

IMPLEMENTASI MODEL YOLOV7 DAN MASK R-CNN DALAM INSTANCE SEGMENTATION CITRA GUNUNG BERAPI

TUGAS AKHIR

Disusun Oleh:

N a m a	: Resa Zulfikar
NIM	: 18523030

Yogyakarta, 8 Mei 2023

Pembimbing,

(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

IMPLEMENTASI MODEL YOLOV7 DAN MASK R-CNN DALAM INSTANCE SEGMENTATION CITRA GUNUNG BERAPI

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 26 Juli 2023

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom. _____

Anggota 1

Irving Vitra Paputungan, S.T., M.Sc., Ph.D. _____

Anggota 2

Sri Mulyati, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Resa Zulfikar
NIM : 18523030

Tugas akhir dengan judul:

IMPLEMENTASI MODEL YOLOV7 DAN MASK R-CNN DALAM INSTANCE SEGMENTATION CITRA GUNUNG BERAPI

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 16 Mei 2023



(Resa Zulfikar)

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin, puji syukur kepada Allah SWT yang telah memberikan rezeki dan kesempatan pada penulis sehingga bisa mendapatkan kesempatan untuk belajar hingga jenjang pendidikan yang lebih tinggi. Hingga diberikan kemampuan untuk bisa menjalankan proses akhir dari jenjang pendidikan ini, yaitu penulisan penelitian skripsi.

Tulisan ini penulis persembahkan kepada kedua orang tua yang penulis sangat cintai, ibu Ika Nursiska dan bapak Wawan Kiswanto. Atas kerja keras keduanya sehingga memungkinkan bagi penulis dan dua saudaranya untuk bisa mengenyam pendidikan tinggi, dan bagi penulis sendiri diberikan kepercayaan untuk berkuliah dan memilih kampus tercinta ini, Universitas Islam Indonesia. Atas doa dan ridha dari kedua orang tua kami tersebut penulis bisa menyelesaikan penulisan skripsi ini. Semoga Allah SWT mengganti setiap jerih payah yang dilakukan orang tua penulis dalam berbagai bentuk kebaikan di dunia maupun di akhirat,
Amiin.

Terima kasih kepada ibu Arrie Kurniawardhani sebagai dosen pembibing yang selalu meluangkan waktu dan dengan sabar menjawab pertanyaan-pertanyaan dari penulis dan membimbing penulisan penelitian ini. Semoga Allah ganti setiap ilmu yang beliau berikan dengan kebaikan-kebaikan lain yang berlipat ganda,
Amiin.

HALAMAN MOTO

“Allah tidak akan menguji seseorang melainkan sesuai kesanggupannya”

QS Al-Baqarah: 286

“A thousand journeys begin with a single step”

Lao Tzu

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi ini dengan judul **"PENGGUNAAN MODEL YOLOV7 DAN *MASK* R-CNN DALAM *INSTANCE SEGMENTATION* CITRA GUNUNG BERAPI"**. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer Jurusan Informatika pada Fakultas Teknologi Industri Universitas Islam Indonesia. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT
2. Keluarga penulis, terutama kedua orang tua penulis Ibu Ika dan Bapak Wawan, yang memberikan dukungan, doa, dan kekuatan.
3. Ibu Arrie Kurniawardhani S.Si., M.Kom. sebagai Dosen Pembimbing penelitian ini.
4. Bapak Moh. Idris, S.Kom., M.Kom. sebagai Dosen Pembimbing Akademik.
5. Bapak Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia
6. Prof. Dr. Ir. Hari Purnomo., M.T., IPU., ASEAN.Eng. sebagai Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Dr. Raden Teduh Dirgahayu, S.T., M.Eng., Ph.D. selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
8. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Universitas Islam Indonesia.
9. Bapak, Ibu dosen jurusan Informatika Universitas Islam Indonesia yang telah memberikan banyak ilmu dan pengalaman berharga.
10. Teman-teman dan orang-orang yang penulis temui selama berkuliah di Yogyakarta.
11. Semua pihak yang secara langsung maupun tidak langsung memberikan bantuan dalam bentuk apapun.

Untuk semua bantuan yang diberikan, sebagai penulis, saya ingin mengucapkan terima kasih. Jika saya mengerjakan tugas akhir ini, saya tidak akan lupa meminta maaf kesalahan atau kekurangan. Penulis berharap artikel ini bermanfaat bagi pembaca dan dapat mengembangkan penelitian ini di masa yang mendatang.

Yogyakarta, 6 Mei 2023



(Resa Zulfikar)

SARI

Indonesia terletak di daerah geografis yang memiliki banyak gunung api aktif serta memiliki aktivitas vulkanisme yang tinggi, seperti gunung Sinabung di Sumatera Utara dan gunung Merapi di Yogyakarta. Hal ini tentu menjadikan gunung api sebagai objek pengamatan yang penting untuk diperhatikan mengingat dampak dari aktivitas vulkanik bisa mempengaruhi kehidupan manusia. Kegiatan pengamatan yang bisa dilakukan mulai dari mempelajari aktivitas dan dampaknya serta penelitian yang mencoba untuk melakukan pengawasan aktivitas gunung api demi ilmu pengetahuan, upaya pencegahan, atau mengantisipasi dampak aktivitas vulkanik pada manusia. Pemantauan yang dilakukan secara visual yaitu melihat atau mengobservasi tampilan luar dari objek yang diamati. Perubahan fisik yang terjadi pada gunung api selama aktivitas vulkanisnya dapat memberikan informasi berharga dan informasi ini didapatkan melalui monitoring dari gunung api. Pengamatan gunung api dapat dilakukan dengan penglihatan manusia dan/atau dibantu dengan alat pengawasan. Maka dari itu, dilakukan monitoring aktivitas gunung api dengan memanfaatkan teknik *instance segmentation*. Konsep segmentasi dari model ini nantinya diharapkan dapat dikembangkan untuk membantu pengamatan dari aliran lava yang keluar dari kawah gunung api atau pun fenomena vulkanisme lainnya seperti kemunculan awan panas. Penelitian ini akan berfokus pada pengembangan model deteksi lava dengan menggunakan model *Mask* Region-Based Convolutional Neural Network (*Mask* R-CNN) dan You Only Look Once Version7 (YOLOV7). Dari penelitian ini diperoleh hasil akurasi sebesar 53% untuk *Mask* R-CNN dan 87% untuk YOLOv7.

Kata Kunci: Deteksi Citra, Gunung Api, Lava, *Mask* R-CNN, Segmentasi, YOLOv7

GLOSARIUM

Image Annotation	Proses memberikan label atau tag pada gambar dengan sekumpulan kata kunci berdasarkan isi dari gambar.
Computer Vision	Salah satu cabang dari *deep learning* yang berkembang pesat dan pemanfaatannya digunakan sebagai rekaman data gambar hingga ekstraksi pola pada gambar sampai mampu menginterpretasikan informasi pada gambar.
Mask R-CNN	Metode yang bekerja untuk menyelesaikan permasalahan instance segmentation pada bidang visi komputer.
Mean Average Precision (mAP)	Parameter yang digunakan untuk mengevaluasi kinerja model secara keseluruhan, yang merupakan nilai rata - rata dari Average Precision (AP) keseluruhan kelas.
Preprocessing	Salah satu tahapan penting dari proses persiapan data, yaitu proses pengolahan data mentah menjadi format yang lebih mudah dan efektif diproses.
Image Segmentation	Proses pemberian label pada tiap pixel yang ada di sebuah citra/gambar yang mana tiap pixel berlabel tersebut merepresentasikan bentuk objek di dunia nyata.
YOLO (You Only Look Once)	Object detection network yang dibuat oleh Joseph Redmon di tahun 2016, dimana metode ini termasuk salah satu metode yang sering digunakan untuk melakukan deteksi objek pada sebuah gambar.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ixx
GLOSARIUM x
DAFTAR ISI xii
DAFTAR TABEL xiiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 3
1.6 Metodologi Penelitian 4
1.7 Sistematika Penulisan 4
BAB II KAJIAN PUSTAKA 6
2.1 Aktivitas Vulkanik Gunung Api 6
2.1.1 Lava 6
2.1.2 Awan Vulkanik 7
2.2 Visi Komputer 8
2.3 Image Segmentation 8
2.4 *Mask* R-CNN 9
2.5 YOLO 11
2.6 Phyton 15
2.7 Mean Average Precision (mAP) 15
2.8 Penelitian Terdahulu 17

2.9 Pemilihan Model Segmentasi 21
BAB III METODOLOGI PENELITIAN 25
3.1 Pengumpulan Data 25
3.2 Anotasi Dataset 28
3.2.1 Anotasi Dataset Model YOLOv7 29
3.2.2 Anotasi Dataset Model *Mask* R-CNN 33
3.3 Data Latih dan Validasi 35
3.4 Data Uji 37
3.5 Persiapan Model Segmentasi 38
3.5.1 Konfigurasi YOLOv7 38
3.5.2 Konfigurasi Mask R-CNN 41
3.6 Evaluasi Hasil 42
BAB IV HASIL DAN PEMBAHASAN 44
4.1 Konfigurasi Tensorflow dan CUDA 44
4.2 Implementasi Model *Mask*-RCNN 44
4.3 Implementasi Model YOLOv7 53
4.4 Penyandingan Kedua Model 59
4.5 Hasil Evaluasi Validasi 60
4.6 Hasil Pengujian 60
4.7 Analisis Kekurangan dan Kelebihan Model 65
BAB V KESIMPULAN DAN SARAN 66
5.1 Kesimpulan 66
5.2 Saran 66
DAFTAR PUSTAKA 67

DAFTAR TABEL

Tabel 2.1 Analisis Penelitian Terdahulu 23
Tabel 3.1 Tangkapan Frame Data Uji Video 25
Tabel 3.2 Metadata Gambar 27
Tabel 3.3 Metadata Video Data Uji 38
Tabel 3.4 Finetuning Model YOLOv7 38
Tabel 3.5 Parameter Settings YOLOv7 39
Tabel 3.6 Finetuning Model Mask R-CNN 40
Tabel 3.7 Parameter Setting Mask R-CNN 41
Tabel 4.1 Konfigurasi Tensorflow dan CUDA 44
Tabel 4.2 Hasil mAP Mask R-CNN dan YOLOv7 60
Tabel 4.3 Hasil Penyandingan Data Uji Video 61
Tabel 4.4 Analisis Kelebihan dan Kekurangan Model 65

DAFTAR GAMBAR

Gambar 2.1 Aliran lava 7
Gambar 2.2 Awan Vulkanik 8
Gambar 2.3 *Mask* R-CNN Framework 10
Gambar 2.4 Arsitektur YOLOv7 12
Gambar 2.5 Hasil Segmentasi Multiple Rice Grain 18
Gambar 2.6 Hasil Segmentasi Buah 19
Gambar 2.7 Hasil Segmentasi Longsor 19
Gambar 2.8 Hasil Segmentasi Hewan Ternak 20
Gambar 2.9 Hasil Segmentasi Pohon 21
Gambar 3.1 Alur Penelitian 25
Gambar 3.2 Sampel Dataset Gunung Berapi. (a) Sumber Google Image. (b) Sumber Website MAGMA 27
Gambar 3.3 Hasil Polygon Labelling 29
Gambar 3.4 Proses Anotasi Citra pada Roboflow 30
Gambar 3.5 Folder Hasil Anotasi Roboflow 30
Gambar 3.6 (a) Struktur Folder Train Hasil Anotasi. (b) Isi Folder Labels 31
Gambar 3.7 Isi Folder Images 31
Gambar 3.8 Isi File .txt pada Folder Labels 32
Gambar 3.9 File YAML Hasil Anotasi Roboflow 32
Gambar 3.10 (a) Pembuatan Label Kelas. (b) Proses Labelling 33
Gambar 3.11 Isi Folder Hasil Anotasi Makesense.ai 34
Gambar 3.12 File JSON Hasil Anotasi Makesense.ai 34
Gambar 3.13 Pembagian Dataset Pada Mask R-CNN 35
Gambar 3.14 Setting Default Pembagian Dataset Mask R-CNN 36
Gambar 3.15 Pembagian Dataset pada Roboflow 36
Gambar 3.16 Tangkapan Frame Data Uji Video. (a) Video 1 – video_merapi.mp4. (b) Video 2 – Krakatoa_lava.mp4 38
Gambar 3.17 Baris Kode Setting pada YOLOv7 39
Gambar 3.18 Sumber Pre-Trained Weight YOLOv7 40
Gambar 3.19 Baris Kode Settings pada Mask R-CNN 41
Gambar 3.20 Flowchart perhitungan nilai mAP 42

Gambar 4.1 Source Code Koneksi Google Drive 44
Gambar 4.2 Source Code Download Import Library Mask RCNN 45
Gambar 4.3 Isi File maskrcnn_colab_commit_10 45
Gambar 4.4 Source Code Ekstrak Dataset 46
Gambar 4.5 Source Code Display Image 46
Gambar 4.6 Gambar Data Citra dan Labelnya 47
Gambar 4.7 Source Code Memuat Konfigurasi dan Model 47
Gambar 4.8 Source Code Training Citra Model Mask R-CNN 48
Gambar 4.9 Source Code Testing Model Terbaik 49
Gambar 4.10 Hasil Riwayat Training 50
Gambar 4.11 Source Code Evaluasi Mean Average Precision (mAP) 50
Gambar 4.12 Fungsi model_evaluation 51
Gambar 4.13 Pendefinisian Nilai IoU 51
Gambar 4.14 Hasil Evaluasi Mean Average Precision (mAP) 52
Gambar 4.15 Hasil Loss Value Model 53
Gambar 4.16 Hasil Loss Validation 35
Gambar 4.17 Struktur Folder YOLOv7 54
Gambar 4.18 Source Code Change Directory 54
Gambar 4.19 Source Code Download Hyperparameter YOLOv7 55
Gambar 4.20 Isi dari Hyperparameter Default YOLOv7 55
Gambar 4.21 Source Code Instalasi Library PyYAML 56
Gambar 4.22 Source Code Pengaturan Model YOLOv7 42
Gambar 4.23 Hasil Training YOLOv7 57
Gambar 4.24 Hasil Evaluasi Loss Training Model YOLOv7 58
Gambar 4.25 Hasil Evaluasi mAP Validasi Model YOLOv7 59

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal terletak di daerah dengan geografis yang memiliki banyak gunung api aktif, misalnya beberapa gunung api yang terkenal seperti gunung Sinabung di Sumatera Utara dan gunung Merapi di Yogyakarta yang memiliki aktivitas vulkanisme yang tinggi. Hal ini tentu menjadikan gunung api sebagai objek pengamatan yang penting untuk diperhatikan mengingat dampak dari aktivitas vulkanis bisa mempengaruhi kehidupan manusia. Kegiatan pengamatan yang bisa dilakukan mulai dari mempelajari aktivitas dan dampaknya serta penelitian yang mencoba untuk melakukan pengawasan aktivitas gunung api demi ilmu pengetahuan, upaya pencegahan, atau mengantisipasi dampak aktivitas vulkanik pada manusia.

Pemantauan visual adalah suatu cara pengamatan dengan melihat atau mengobservasi tampilan luar dari objek yang diamati. Monitoring sendiri merupakan proses pengamatan perubahan yang terjadi pada waktu tertentu. Hasil monitoring sendiri dapat memberikan informasi berharga yang memperkaya pengetahuan mengenai objek yang diamati. Dalam konteks pengawasan gunung api, monitoring sangat penting dilakukan untuk mengamati perubahan fisik selama aktivitas vulkanik. Monitoring gunung api dapat dilakukan dengan penglihatan manusia atau dengan bantuan alat pengawasan. Objek pengamatan mencakup aktivitas geologis dan vulkanis yang terlihat dan dirasakan, seperti munculnya lava, terjadi getaran seperti gempa bumi, atau munculnya awan/abu vulkanik. Informasi yang diperoleh dari monitoring ini sangat membantu para ilmuwan dalam memprediksi dan mengantisipasi kemungkinan erupsi gunung api yang akan terjadi.

Lava merupakan lelehan batu yang mencair dan berpijar yang muncul ke permukaan sebagai akibat dari aktivitas vulkanik. Berbeda dengan magma yang merupakan kondisi lain dari lelehan batu yang belum muncul ke permukaan dan keluar dari kawah gunung. Dalam fenomena alam, aliran lava adalah salah satu fenomena yang paling umum dan mudah untuk diidentifikasi sebagai hasil dari aktivitas erupsi gunung api. Dari hasil peninjauan dampak dan risiko gunung api, apapun yang terdampak oleh aliran lava dipastikan akan mengalami kerusakan parah atau hancur total (Jenkins, Day, Faria, & Fonseca, 2017). Pengaruh aliran lava terhadap tingkat kerusakan juga bertambah parah dengan tingginya tingkat kepadatan

penduduk yang ada di daerah berisiko terdampak aktivitas vulkanik (Negro, Cappello, & Bilotta, 2020). Meskipun vulkanologi telah menjadi cabang ilmu pengetahuan alam dan dapat menciptakan kebutuhan akan ahli dalam bidang ini mengingat dampak yang ditimbulkan gunung api cukup besar. Namun, kemampuan dan jumlah ahli untuk melakukan pemantauan hingga saat ini masih belum mencukupi jika dilihat dari jumlah gunung api yang harus diawasi. Negara Indonesia memiliki 129 gunung api aktif, dan dari jumlah gunung api yang masih aktif tersebut hanya terdapat 69 gunung yang terpantau, maka masih banyak gunung api yang aktif yang tidak terpantau secara baik yaitu sekitar 40% (Virkhansa, Setiawan, & Dewi, 2019). Dampak destruktif dari pergerakan lava dan kurangnya sumber daya yang memadai untuk melakukan pengamatan menjadi dua alasan utama mengapa diperlukannya upaya untuk memenuhi kekosongan tersebut. Hal ini tentu menjadi alasan diperlukannya upaya pemantauan aliran lava untuk mendukung upaya mitigasi kebencanaan gunung api dan melindungi masyarakat. Berkat kemajuan di bidang teknologi seperti perangkat lunak dan keras, sudah banyak dikembangkan berbagai model dan metode yang dapat digunakan untuk menggantikan tenaga manusia. Misal, pengamatan atau monitoring dapat menggunakan kamera yang bisa bekerja 24 jam dan pemantauan dilakukan di tempat yang jauh, tetapi masih membutuhkan tenaga manusia yang harus memantau hasil rekaman. Teknologi lain yang memungkinkan untuk digunakan dengan tenaga manusia paling minimal adalah deteksi objek dari sebuah gambar yang dilakukan oleh sebuah program sehingga mesin yang akan mendeteksi ada atau tidaknya objek yang diamati, dan salah satu teknologi yang bisa digunakan adalah image segmentation atau segmentasi citra. Image segmentation adalah proses pemberian label pada tiap pixel yang ada di sebuah citra/gambar yang mana tiap pixel berlabel tersebut merepresentasikan bentuk objek di dunia nyata. image segmentation bisa diartikan juga sebagai proses mendefinisikan batas-batas antar entitas yang terpisah dalam sebuah gambar (Ghosh, Das, Das, & Maulik, 2019).

Maka dari itu, penelitian ini dilakukan sebagai upaya untuk mengusulkan penggunaan model *image segmentation* yang dapat melakukan deteksi aktivitas dari gunung api. Konsep deteksi dari model ini diharapkan dapat dikembangkan untuk membantu pengamatan, seperti kemunculan aliran lava yang keluar dari kawah gunung api. Penelitian ini akan berfokus pada pengembangan model segmentasi aktivitas gunung berapi dan sekitarnya.

1.2 Rumusan Masalah

Berdasarkan pemaparan permasalahan yang sudah disebutkan sebelumnya, maka rumusan masalah yang didapat ialah:

a. Bagaimana penerapan metode image segmentation *Mask* Region-Based Convolutional Neural Network (*Mask* R-CNN) dan You Only Look Once Version7 (YOLOv7) dalam segmentasi objek citra gunung berapi.
b. Bagaimana hasil kinerja dari model image segmentation *Mask* R-CNN dan YOLOv7 dalam segmentasi objek yang ada di sekitar gunung berapi.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

a. Model yang digukanan adalah YOLOv7 dan *Mask* R-CNN.
b. Dataset yang digunakan adalah citra gunung berapi yang menunjukkan aktivitas vulkanisme pada siang hari.
c. Hanya dilakukan implementasi model algoritma YOLOv7 dan *Mask* R-CNN untuk melakukan segmentasi objek.
d. Objek segmentasi terdiri dari 4 kelas, yaitu; lava, awan vulkanik, gunung, dan vegetasi.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. Menerapkan metode *image segmentation Mask* R-CNN dan YOLOv7 dalam segmentasi empat objek yang ada di sekitar gunung berapi.
b. Mengetahui hasil kinerja dari model *image segmentation Mask* R-CNN dan YOLOv7 dalam segmentasi empat kelas objek yang ada di sekitar gunung berapi.

1.5 Manfaat Penelitian

Manfaat dilakukannya penelitian dalam upaya untuk segmentasi empat objek yang ada di sekitar gunung berapi ini adalah sebagai berikut:

a. Membantu melakukan monitoring gunung api berkaitan dengan kemunculan aktivitas vulkanik.
b. Mengisi kekurangan sumber daya manusia dalam melakukan pengamatan gunung api yang tidak terpantau.

1.6 Metodologi Penelitian

Tahap metodologi penelitian yang akan dilakukan untuk mendukung proses penelitian mengikuti tahapan berikut:

a. Pengumpulan Data

Tahap ini menggunakan dua metode dan sumber pengambilan data yaitu studi literatur yang bersumber dari penelitian sebelumnya, dan metode dokumentasi yang merupakan kumpulan fakta yang menyimpan informasi di masa lampau seperti foto atau video.

b. Pengembangan Model

Pengembangan model terkait dengan penelitian ini akan menggunakan metode YOLOv7 dan *Mask* R-CNN yang merupakan salah metode yang sering digunakan untuk melakukan segmentasi objek pada sebuah gambar. Proses segmentasi objek akan dilakukan dengan menentukan terlebih dahulu lokasi dari setiap objek yang ingin dideteksi pada gambar.

c. Uji Model

Sebagai tahap yang akan melihat sebaik apa model yang sudah dikembangkan dalam mencapai tujuan dan menjawab rumusan masalah.

1.7 Sistematika Penulisan

Susunan laporan tugas akhir ini dijelaskan dalam sistematika penulisan yang akan terbagi dalam lima bab dengan tujuan mempermudah dalam membahas dan menyampaikan penelitian yang dilakukan. Kelima bab tersebut adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab pendahuluan menjelaskan dasar-dasar penelitian dilakukan, seperti permasalahan yang ditemukan, dan sub bab yang mendukung terwujudnya penelitian. Sub bab itu sendiri terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penelitian.

BAB II KAJIAN PUSTAKA

Bab ini membahas mengenai konsep-konsep dasar yang digunakan sebagai landasan teori penelitian, di mana salah satunya menyangkut dengan gunung api dan sifat-sifatnya, dan juga penjelasan image segmentation serta penelitian terdahulu yang terkait dengan topik penelitian

BAB III METODOLOGI PENELITIAN

Bagian bab ini menjelaskan bagaimana metode yang digunakan untuk penelitian dilakukan, dimulai dari pengumpulan data, pengembangan model, dan pengujian model untuk melihat apakah metode yang digunakan sesuai dengan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Secara umum bab ini membahas hasil akhir yang didapatkan selama proses penelitian tugas akhir ini, seperti hasil dari proses pengembangan model untuk penelitian hingga pembahasan hasil pengujian.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisikan kesimpulan dari hasil penelitian yang telah dilakukan, serta saran yang ditujukan untuk pengembangan dan peningkatan lebih baik lagi demi kualitas penelitian kedepannya.

BAB II
KAJIAN PUSTAKA

2.1 Aktivitas Vulkanik Gunung Api

Gunung api adalah rekahan yang dekat dengan kerak bumi tempat keluarnya magma atau gas ke permukaan bumi. Gunung api cukup umum muncul di daerah yang secara geologis sangat aktif dan biasanya terdapat di area yang merupakan perbatasan lempeng bumi di mana kegiatan vulkanis yang muncul berada di dalam lempeng bumi namun tidak terkait dengan lokasi batas dan aktivitas lempeng bumi (Sun, Magee, & Jackson, 2020). Gunung meletus atau dikenal juga sebagai aktivitas erupsi gunung berapi merupakan peristiwa alam yang terjadi akibat endapan magma yang terakumulasi di dalam perut bumi yang terdorong keluar karena tekanan tinggi (Sun, Magee, & Jackson, 2020). Kejadian erupsi gunung api dapat menyebabkan bencana, seperti aliran lava dan awan panas (Sudarsono, Yuwono, & Ramadhan, 2019).

2.1.1 Lava

Lava adalah bebatuan yang meleleh dan berpijar yang berasal dari kedalaman bumi dan keluar dari gunung api melalui rekahan yang terbentuk akibat aktivitas vulkanik. Lava terdiri dari berbagai jenis mineral dan elemen yang membentuk batuan yang keras ketika dingin. Ketika lava masih dalam kondisi meleleh dan belum keluar dari gunung api, maka disebut sebagai magma. Aliran lava yang mengalir dari gunung api dapat sangat merusak, baik bagi lingkungan alam maupun infrastruktur buatan manusia. Oleh karena itu, lava sering menjadi ancaman bagi kehidupan manusia dan perlu dipantau dengan cermat oleh para ahli vulkanologi.

Dari proses keluarnya lava dari kawah gunung menyebabkan munculnya fenomena yang dikenal sebagai aliran lava. Di mana lava mengikuti kontur gunung dan turun melalui celah-celah atau struktur gunung seperti lereng atau sungai. Aliran lava memiliki komposisi yang sama dengan magma dan memiliki suhu antara 800ºC- 1200ºC yang secara visual tampak seperti lelehan atau cairan berwarana merah terang dan merah tua (Griffiths, 2000). Seperti ditampilkan pada Gambar 2.1 warna dari lava ini menunjukkan komposisi batuan yang tercampur atau sedang mengalami proses pendinginan dan kembali menjadi batuan.



Gambar 2.1 Aliran lava

Sumber: Griffiths (2003)

2.1.2 Awan Vulkanik

Awan Vulkanik merupakan bagian dari aktivitas erupsi gunung berapi yang dicirikan dengan gulungan awan dengan komponen penyusunnya berupa batuan pijar yang panas dan material hasil aktivitas vulkanik padat seperti batu, kerikil, pasir, dan debu vulkanik yang tersembur keluar bersamaan dengan gas yang memiliki suhu 600℃ atau lebih dari itu (Sudarsono, Yuwono, & Ramadhan, 2019).

Semburan awan vulkanik yang muncul ke permukaan seperti gulungan awan maka sering disebut juga sebagai awan panas. Seperti pada Gambar 2.2, awan vulkanik bergerak setelah muncul ke permukaan dari atas puncak gunung. Dalam beberapa kondisi, komponen awan vulkanik tertarik gravitasi atau muatan materialnya terlalu berat dan padat sehingga jatuh menuju ke permukaan. Awan vulkanik memiliki kekuatan yang merusak dikarenakan kuatnya daya luncur yang disebabkan daya dorong gas, jarak jangkauan dari awan panas ini juga ditentukan dari daya dorong yang mempengaruhi kecepatan luncurnya.



Gambar 2.2 Awan Vulkanik

Sumber: MAGMA Indonesia (Multiplatform Application for Geohazard Mitigation and Assessment in Indonesia)

2.2 Visi Komputer

Visi komputer telah menjadi salah satu cabang dari kecerdasan buatan yang berkembang pesat dan pemanfaatannya telah meluas, mulai dari ranah rekaman data gambar hingga ekstraksi pola pada gambar sampai mampu menginterpretasikan informasi pada gambar. Visi komputer memiliki kombinasi konsep, teknik, dan ide dari *digital image processing*, *pattern recognition*, *artificial intelligence*, dan *computer graphics* (Wiley & Lucas, 2018). Konsep awal visi komputer adalah untuk membuat komputer memiliki kemampuan mendapatkan informasi dari citra atau objek visual seperti pengamatan mata manusia. Output dari visi komputer adalah "image understanding" atau proses menginterpretasikan area/objek untuk memahami apa yang terjadi pada suatu data citra (Wiley & Lucas, 2018).

2.3 Image Segmentation

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (region) di mana setiap daerah memiliki kemiripan atribut. Segmentasi membagi citra ke dalam daerah intensitasnya masing-masing sehingga bisa membedakan antara objek dengan objek lain dan background-nya. Segmentasi citra bertujuan untuk membagi area-area yang homogen (Kusuma & Ellyana, 2018). Dalam konteks citra digital daerah hasil segmentasi

tersebut merupakan kelompok piksel yang bertetangga atau berhubungan. Segmentasi citra dapat dilakukan melalui beberapa pendekatan antara lain:

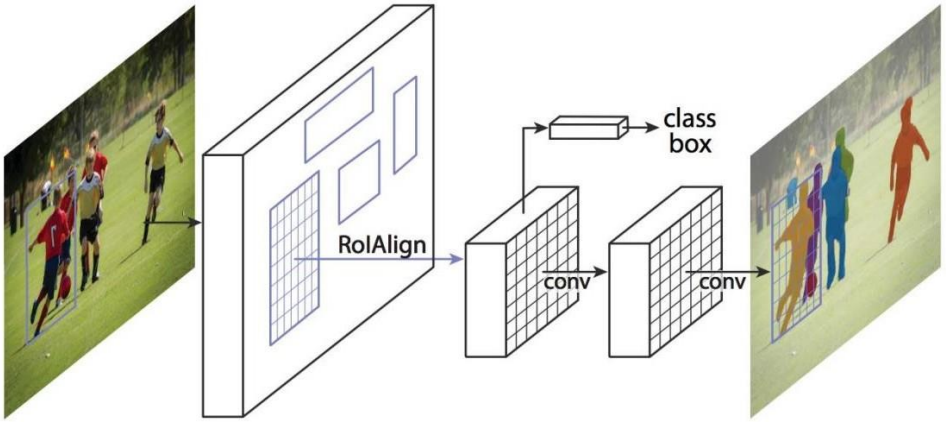
1. Pendekatan batas (boundary approach), pendekatan ini dilakukan untuk mendapatkan batas yang ada antar objek yang berbeda dalam citra. Cara kerja algoritma ini dengan mencari perubahan intensitas atau gradien warna yang tajam dalam citra, yang menandakan adanya batas antara objek yang berbeda.
2. Pendekatan tepi (edge approach), hampir mirip dengan pendekatan batas, pendekatan tepi juga mencari perubahan intensitas atau gradien warna yang tajam pada citra. Namun, pendekatan ini lebih berfokus pada identifikasi tepi atau kontur objek yang berbeda dalam citra dibandingkan batas antar objek. pendekatan tepi dilakukan untuk mengidentifikasi piksel tepi dan menghubungkan piksel-piksel tersebut menjadi suatu batas yang diinginkan.
3. Pendekatan daerah (region approach), pendekatan daerah bertujuan untuk membagi citra dalam daerah-daerah sehingga didapatkan suatu daerah sesuai kriteria yang sama. Daerah ini dikelompokkan berdasarkan piksel yang serupa atau memiliki atribut yang sama seperti intensitas atau warna. Algoritma pada pendekatan ini menggabungkan piksel yang serupa menjadi daerah atau segmen yang koheren.

2.4 *Mask* R-CNN

Mask Region-Based Convolutional Neural Network atau *Mask* R-CNN merupakan sebuah metode yang bekerja untuk menyelesaikan tugas *instance segmentation* pada bidang visi komputer. *Instance segmentation* berfokus pada penyelesaian tugas *object detection* dan *image segmentation*, di mana *object detection* melakukan proses untuk menemukan dan mengklasifikasi objek dalam sebuah citra atau gambar. Sedangkan *instance segmentation* merupakan proses pengklasifikasian piksel-piksel pada suatu citra dengan melakukan *Masking* atau arsiran pada piksel yang merupakan bagian dari kelas objek yang teridentifikasi (Ma, et al., 2021).

Mask R-CNN merupakan kelanjutan dari model *Faster* R-CNN dengan menggunakan *feature extraction* dan menambahkan fitur region proposal network. Kemudian *Mask*-RCNN dilengkapi dengan operasi yang dikenal sebagai ROI-Pooling yang terdiri dari modifikasi penting untuk menghasilkan output sebagai input ke classifier. Perbedaan *Mask*-RCNN dan Faster-RCNN yaitu pertama *Mask*-RCNN menggantikan operasi ROI-Pooling yang kurang akurat pada Faster-RCNN dengan operasi yang disebut ROI-Align yang memungkinkan

menghasilkan *instance segmentation* yang lebih akurat. Kedua, *Mask*-RCNN menambahkan kepala jaringan atau network head (jaringan saraf kecil fully convolutional network) untuk menghasilkan instance segmentation yang diinginkan (Anantharaman, Velazquez, & Lee, 2018). Sebagaimana diilustrasikan pada Gambar 2.3.



Gambar 2.3 *Mask* R-CNN Framework

Sumber: Anantharama et al., (2018)

Kemudian, prediksi *Masking* dan kelas dipisahkan; Kepala jaringan *Mask* memprediksi area arsiran secara independen dari kepala jaringan yang memprediksi kelas. Proses ini memerlukan penggunaan multi-task loss function (Anantharaman, Velazquez, & Lee, 2018). *Mask* R-CNN mendefinisikan fungsi loss dapat dilihat pada Persamaan (2.1):

$$L = L_{cls} + L_{box} + L_{Mask} \qquad (2.1)$$

Di mana L_{cls} menunjukkan kesalahan klasifikasi; L_{box} menunjukkan kesalahan regresi kotak pembatas dan L_{Mask} menunjukkan kesalahan *Mask*.

Proses Segmentasi Pada *Mask* R-CNN

Dengan penyiapan dataset dan konfigurasi yang sudah dipilih maka model *Mask* R-CNN selanjutnya akan melakukan proses mengidentifikasi objek kelas pada citra gunung berapi. Proses ini termasuk untuk mendapatkan hasil deteksi dan segmentasi oleh model yang dijelaskan melalui tahap berikut:

1. Tahap Backbone Network:

Citra gunung berapi yang sudah dipersiapkan sebagai input akan masuk melalui sebuah jaringan konvolusi yang merupakan bagian dari *backbone network*. Jaringan konvolusi ini akan mengekstraksi fitur-fitur visual tingkat tinggi dari citra, seperti tekstur, kontur, dan

bentuk yang terkait dengan objek kelas yang ingin diidentifikasi, seperti gunung, lava, awan vulkanik, dan vegetasi.

2. Tahap Region Proposal Network (RPN):

Setelah melewati backbone network, tahap RPN akan menghasilkan proposal wilayah yang kemungkinan mengandung objek dalam citra gunung berapi. Misalnya, RPN dapat menghasilkan proposal wilayah yang mencakup bagian gunung, area dengan pola yang menyerupai lava, awan tebal di sekitar gunung, dan area dengan tumbuhan hijau yang merupakan vegetasi.

3. Tahap Region of Interest (RoI) Allign:

Setelah mendapatkan proposal wilayah dari tahap RPN, tahap RoI Allign dilakukan untuk mengekstraksi fitur-fitur dari setiap proposal wilayah yang akan digunakan untuk segmentasi objek, dan juga menggunakan informasi hasil ekstraksi fitur dari tahap backbone untuk meningkatkan keyakinan model ketika akan memprediksi objek yang akan disegmentasi. RoI Allign akan menggabungkan fitur-fitur dari proposal wilayah hasil RPN dan backbone yang mencakup bagian gunung, seperti kontur, bentuk, batas, dan warna yang berdasarkan hasil pengetahuan proses pelatihan termasuk ke dalam objek gunung.

4. Tahap Head Network (Output):

Setelah tahap RoI Allign, tahap head network akan menghasilkan *output* dari model, termasuk kelas prediksi objek dan *Masking* (segmentasi) untuk setiap proposal wilayah. Tahap ini dapat menghasilkan kelas "gunung" dengan tingkat kepercayaan tertentu, serta masking atau segmentasi yang menunjukkan wilayah di dalam proposal yang merupakan bagian dari gunung tersebut. Selain itu, tahap ini juga akan menghasilkan kelas dan segmentasi untuk objek-objek lain seperti lava, awan vulkanik, dan vegetasi dalam citra.

2.5 YOLO

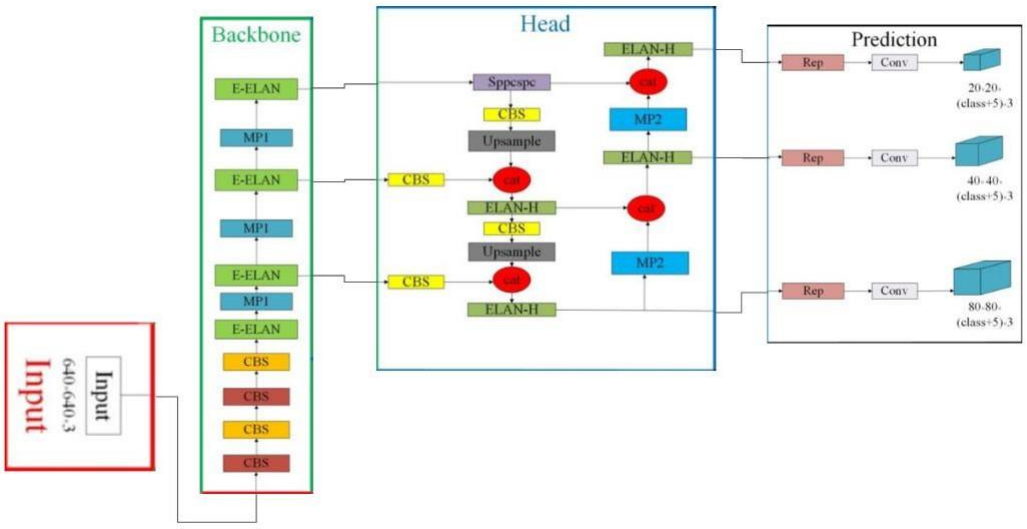
You Only Look Once (YOLO) merupakan *object detection network* yang dibuat oleh Joseph Redmon di tahun 2016, di mana metode ini termasuk salah satu metode yang sering digunakan untuk melakukan deteksi objek pada sebuah gambar. Proses deteksi objek akan dilakukan dengan menentukan terlebih dahulu lokasi dari setiap objek yang ingin dideteksi pada gambar. Di mana YOLO merupakan sebuah metode yang menggunakan penerapan jaringan saraf tunggal (single neural network) pada keseluruhan gambar (Mailoa & Santoso, 2022). Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi

kotak pembatas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasian sebagai objek atau bukan.

YOLOv7

Model YOLOv7 adalah model terbaru dari keluarga model YOLO. Model YOLOv7 merupakan detektor objek satu tahap. Pada model YOLO, frame citra diekstrak fitur-fiturnya melalui sebuah *backbone*. Fitur-fitur ini dicampurkan lalu diteruskan ke kepala jaringan. YOLO memprediksi lokasi dan kelas objek di sekitar bounding box yang harus digambarkan (Sholahuddin, et al., 2023).

Model YOLOv7 dikembangkan oleh Chien-Yao Wang dan Alexey Bochovsjiy pada tahun 2022. YOLOv7 adalah model deteksi objek real-time yang dirancang untuk mencapai keseimbangan antara efisiensi deteksi dan presisi. Arsitektur YOLOv7 terdiri dari empat modul yaitu: Input, Backbone, Head dan Prediction (Liu, et al., 2023).



Gambar 2.4 Arsitektur YOLOv7

Sumber: (Liu, et al., 2023)

Berikut adalah penjelasan arstitektur YOLOv7:

1. Input: Input module YOLOv7 menggunakan metode perhitungan bingkai *anchor* adaptif YOLOv5 untuk menyesuaikan ukuran gambar menjadi 640x640 piksel, sesuai dengan persyaratan ukuran masukan Backbone *network*.
2. Backbone: Jaringan Backbone YOLOv7 terdiri dari tiga modul utama: CBS (Convolution, Batch Normalizaiton, SiLU), E-ELAN (Extended-Eficient Layer Agregation Netwrok), dan MP1 (Module Pooling 1). Modul CBS menggunakan fungsi

konvolusi, normalisasi batch, dan aktivasi SiLU. Modul E-ELAN mempertahankan arsitektur ELAN asli dan meningkatkan kemampuan pembelajaran jaringan. MP1 terdiri dari CBS dan MaxPool dan memiliki cabang atas dan bawah. Cabang atas menggunakan MaxPool dan CBS dengan 128 output channels, sementara cabang bawah mengurangi panjang dan lebar gambar dengan CBS 3 × 3 dan stride 2 × 2, dan menyatukan fitur dari kedua cabang dengan tujuan meningkatkan kemampuan ekstraksi fitur jaringan.

3. Head: Jaringan Head pada YOLOv7 menggunakan arsitektur FPN dan desain PANet, terdiri dari blok CBS, Sppcspc, E-ELAN, dan MP2. Struktur Sppcspc menggabungkan CSP dan SPP, blok MP2 mirip dengan MP1 dengan modifikasi jumlah output channel.
4. Prediction: Jaringan Prediksi YOLOv7 menggunakan struktur Rep untuk menyesuaikan saluran gambar dan aplikasi konvolusi 1x1 untuk prediksi kepercayaan, kategori, dan bingkai *anchor*. Struktur Rep terinspirasi dari RepVGG dan memperkenalkan desain residual khusus untuk membantu tahap pelatihan. Struktur residual unik ini dapat juga disebut sebagai konvolusi sederhana pada prediksi, yang menghasilkan tahapan mengurangi kompleksitas jaringan tanpa mengorbankan performa prediktifnya.

Proses Segmentasi Pada YOLOv7

Setelah mempersiapkan kongifurasi dan dataset sebagai input kepada model YOLOv7, maka tahap selanjutnya adalah melakukan proses segmentasi untuk mengidentifikasi objek kelas yang sudah ditentukan. Pada model YOLOv7 tahapan segmentasi melalui proses berikut:

1. Tahap Input dan Preprocessing:

Citra gunung berapi akan dijadikan sebagai input untuk model YOLOv7. Citra kemudian akan melalui tahap preprocessing, di mana ukurannya dapat diubah ke ukuran yang sesuai dengan kebutuhan model.

2. Tahap Backbone Network:

Setelah preprocessing, citra akan melewati sebuah backbone network dalam model YOLOv7, di mana pada model yang digunakan penelitian ini arsitektur backbone terdiri dari komponen CBS, E-ELAN, dan MP1. Backbone network bertujuan untuk mengekstraksi fitur-fitur penting dalam citra yang berhubungan dengan objek yang ada dalam citra gunung berapi, seperti kontur, tekstur, dan pola.

3. Tahap Deteksi Objek (Head):

Setelah melalui backbone network, citra akan dibagi menjadi grid atau bagian. Setiap sel grid bertanggung jawab untuk memprediksi lokasi dan kelas objek dalam sel tersebut. Pada tahap

deteksi objek ini YOLOv7 melakukan prediksi objek secara bersamaan pada seluruh grid yang membagi citra, detail tahapan ini dapat dijelaskan sebagai berikut:

- Setiap grid dalam citra input memiliki tugas untuk memprediksi objek yang ada di dalamnya. Grid-grid ini didefinisikan dalam tingkat resolusi dan ukuran tertentu, yang menentukan seberapa detail objek yang dapat diprediksi. Sebagai contoh dalam sebuah gambar gunung berukuran 640x640 akan dibagi menjadi 4 (empat) grid/area prediksi.
- Pada setiap grid, detection layer menghasilkan beberapa bounding box yang disebut "anchor boxes" atau "prior boxes". Anchor boxes memiliki ukuran dan rasio aspek yang telah ditentukan sebelumnya dan digunakan untuk merepresentasikan objek dalam berbagai skala dan aspek rasio. Ketika gambar sudah dibagi menjadi empat area, disetiap areanya detection layer akan membaca informasi fitur, dan memprediksi kemungkinan batas dari sebuah objek pada gambar berdasarkan informasi fitur dari layer backbone. Dalam sebuah grid dimungkinkan model hanya memprediksi sebagain area yang dianggap, misal, kelas gunung karena terpotong oleh batas grid. Meskipun hanya separuh area gunung, namun anchor box tetap akan terbentuk dan melakukan prediksi kemungkinan posisi objek.
- Detection layer juga menghasilkan prediksi untuk setiap anchor box yang mencakup:
    - Koordinat kotak pembatas (bounding box) yang mengelilingi objek.
    - Probabilitas kelas objek yang didapatkan oleh bounding box tersebut.

Untuk setiap grid, prediksi objek yang dihasilkan oleh detection layer diterjemahkan menjadi probabilitas kelas objek dan kotak pembatas yang berlaku dalam koordinat citra ukuran asli. Model akan menghasilkan segmentasi kelas objek dan kotak pembatas yang mengandung informasi tentang lokasi dan ukuran objek yang ada dalam citra.

4. Tahap Non-Maximum Suppression (NMS):

Setelah tahap deteksi objek, hasil dari prediksi setiap grid akan digabungkan oleh NMS. Tahap NMS dilakukan untuk menghilangkan deteksi yang tumpang tindih dan memilih deteksi dengan tingkat kepercayaan tertinggi. NMS memungkinkan untuk menjaga satu deteksi yang paling akurat dan relevan untuk setiap objek dalam citra gunung berapi. Dengan menghapus deteksi yang tumpang tindih, YOLOv7 dapat menghasilkan prediksi objek yang lebih bersih dan tidak terjadi duplikasi.

5. Tahap Prediction (Output):

Tahap prediction dilakukan untuk menginterpretasikan dan memvisualisasikan hasil final pada tahap NMS. Tahap ini melibatkan menggambar kotak pembatas (bounding box) yang mengelilingi objek yang terdeteksi dan memberikan label kelas yang sesuai. Hasil segmentasi citra gunung berapi dengan YOLOv7 juga akan menampilkan objek-objek yang terdeteksi, seperti gunung, lava, awan vulkanik, dan vegetasi.

2.6 Phyton

Python merupakan bahasa pemrograman yang berorientasi obyek dinamis, mudah dipahami, dan dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. Python hadir dengan *libraries* yang dapat diperluas (Ngantung & Pakereng, 2021). Python sebagai bahasa komputasi untuk ilmuwan dan insinyur memiliki sintaks yang jelas. Sintaks yang jelas ini dapat membuat kode mudah dimengerti dan mudah untuk di-maintenance (Winata, 2022).

2.7 Mean Average Precision (mAP)

Mean Average Precision (mAP) merupakan parameter yang digunakan untuk mengevaluasi kinerja model secara keseluruhan, yang merupakan nilai rata - rata dari Average Precision (AP) keseluruhan kelas (Yuliyanto, Amani, Akhyar, & Usman, 2022). Mean Average Precision digunakan pada algoritma deteksi objek seperti R-CNN, Faster R-CNN, YOLO, dll. Mean average precission bisa digunakan untuk membandingkan model yang berbeda untuk tugas yang sama atau tugas berbeda untuk satu model yang akan dievaluasi (Sun et al., 2022). Mean average precision merata-ratakan nilai AP dari setiap kelas yang ada pada model yang telah diuji (Rahman, Agustin, Ibrahim, & Kumalasari, 2022).

a. Precision

Merupakan perhitungan seberapa akurat sebuah model melakukan prediksi positif atau prediksi yang tepat (True Positive) dan diklasifikasikan sebagai kelas objek yang benar (Pitriyani, 2019). Perhitungan precision menggunakan Persamaan (2.2)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.2)$$

Keterangan:

TP = objek yang benar terdeteksi oleh model sebagai suatu kelas

FP = objek yang salah terdeteksi oleh model sebagai suatu kelas

b. Recall

Adalah metrik yang mengukur seberapa baik sebuah model mampu mengidentifikasi objek yang sebenarnya relevan dari kesuluruhan objek yang ada atau bagaimana sebuah model mengukur kemampuannya untuk menemukan objek yang benar dalam suatu citra. Perhitungan recall menggunakan Persamaan (2.3)

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.3)$$

Keterangan:

FN = objek yang benar tidak terdeteksi oleh model sebagai suatu kelas

c. Average Precision

Average Precision (AP) adalah metrik yang menggabungkan *precision* dan *recall* untuk memberikan gambaran menyeluruh tentang kinerja model dalam mengidentifikasi objek dalam satu kelas. AP dihitung dengan mengambil rata-rata precision pada setiap titik di mana recall berubah atau meningkat (Pitriyani & Ardiansyah, 2019). Untuk menghitng nilai AP dapat dirumuskan dalam Persamaan (2.4).

$$AP = \int_0^1 p(r)dr \quad (2.4)$$

Keterangan:

AP = average precision

p = nilai precision

r = nilai recall

d. Mean Average Precision

Hasil average precision selanjutnya akan digunakan untuk mencari mean average precision (MAP) yaitu metrik yang digunakan untuk mengukur kinerja model secara keseluruhan untuk semua kelas objek yang ada dalam dataset. Menghitung mean average precision menggunakan Persamaan (2.5)

$$mAP = \frac{1}{N} \sum AP_i \qquad (2.5)$$

keterangan:

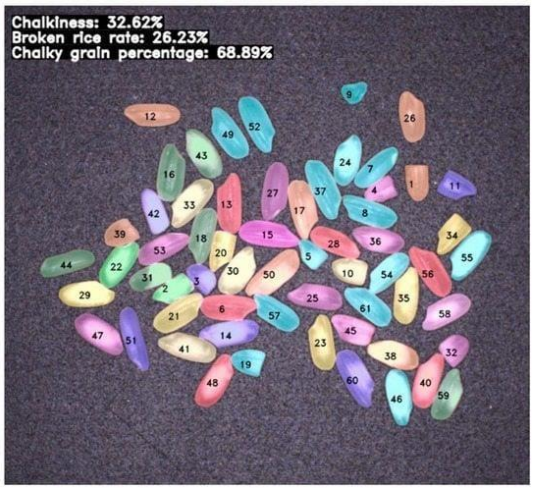
N = Jumlah kelas objek

AP_i = Average precission untuk setiap objek kelas ke-i

2.8 Penelitian Terdahulu

Dengan kajian pustaka ini akan dilakukan kajian penelitian terdahulu yang telah dilakukan sebagai pendukung penyusunan tugas akhir. Pemilihan pustaka yang akan dikaji dilihat dari relevansinya dengan topik penelitian terkait yaitu mengenai Metode YOLO dan *Mask* R-CNN.

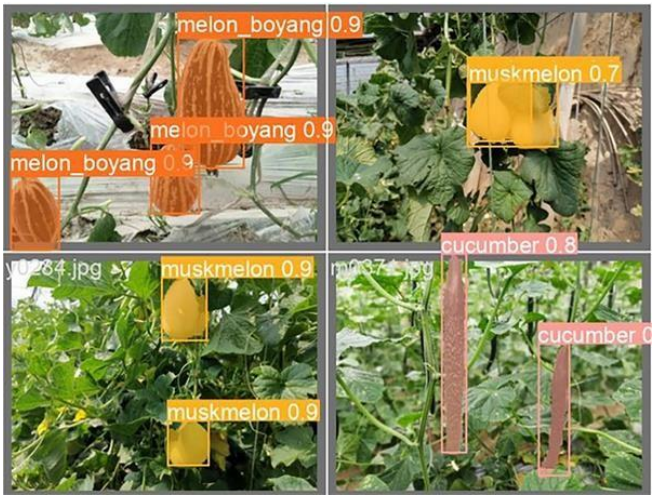
Pada penelitian yang dilakukan oleh (Zhou et al., 2023), membahas tentang penggunaan model YOLOv5 dan YOLACT dalam segmentasi citra beras. Dalam penelitian ini, digunakan dataset buatan sendiri yang terdiri dari 1.200 citra beras yang dibagi menjadi data train (960 citra), val (120 citra), dan test (120 citra). Anotasi dilakukan dengan menggunakan LabelImg. Proses segmentasi dilakukan dengan menggunakan YOLOv5 untuk mendeteksi lokasi ROI dan mengurangi kompleksitas citra, kemudian dilanjutkan dengan YOLACT untuk mendapatkan mask dari ROI. Hasil segmentasi diukur dengan nilai mAP, dimana pada tahap pertama mAP YOLOv5 mencapai 92,1% dan pada tahap kedua mAP YOLACT mencapai 77,5%. Penelitian ini menunjukkan bahwa penggunaan model YOLOv5 dapat meningkatkan akurasi *instance* segmentasi citra beras. Selain itu, dataset buatan sendiri yang digunakan dalam penelitian ini juga dapat menjadi referensi bagi penelitian selanjutnya dalam bidang segmentasi citra beras. Berdasarkan Gambar 2.5 didapatkan hasil citra segmentasi dari bulir-bulir beras untuk melihat kualitas beras, dengan melihat ada atau tidaknya “chalkiness” atau pengapuran, dan mendeteksi “broken rice rate” untuk melihat apakah ada bulir beras yang rusak.



Gambar 2.5 Hasil Segmentasi Multiple Rice Grain

Sumber: (Zhou et al., 2023)

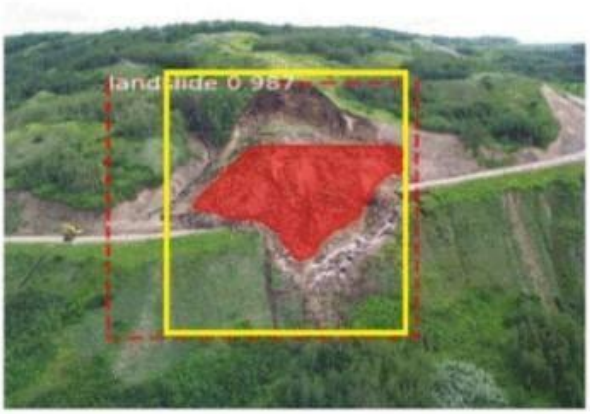
Pada penelitian yang dilakukan oleh (Lawal, 2023), membahas tentang model YOLOv5-LiNet untuk *instance* segmentasi buah. Penelitian ini menggunakan dataset gambar buah yang diambil dari kebun buah di Wanghaizhuang Greenhouses, Jinzhong, Shanxi, China. Dataset ini terdiri dari empat jenis buah yaitu bitter-melon, cucumber, muskmelon, dan melon-boyang dengan total 2469 gambar. Dataset ini diambil dengan mempertimbangkan faktor lingkungan seperti bayangan, cahaya rendah, dan cahaya tinggi. Dataset ini dibagi menjadi 80% untuk train-set, 15% untuk valid-set, dan 5% untuk test-set. Proses anotasi dilakukan dengan menggunakan tools COCO Annotator untuk melakukan anotasi *bounding box* dan *polygon*. Proses segmentasi dilakukan dengan menggunakan *backbone network* yang terdiri dari Stem, Shuffle_Block, ResNet, dan SPPF serta neck network yang menggunakan PANet. Hasil evaluasi menunjukkan bahwa model YOLOv5-LiNet memiliki nilai mAP box sebesar 0.893 dan mAP instance segmentation sebesar 0.885, yang menunjukkan performa yang lebih baik dibandingkan dengan model lightweight lainnya. Pada Gambar 2.6 menunjukkan hasil segmentasi buah pada penelitian oleh (Lawal, 2023) untuk segmentasi empat kelas buah.



Gambar 2.6 Hasil Segmentasi Buah

Sumber: (Lawal, 2023)

Pada penelitian yang dilakukan oleh (Ullo, et al., 2021) metode *Mask* R-CNN digunakan untuk melakukan deteksi tanah longsor. Data yang digunakan pada penelitian ini berjumlah 160 dataset, terdiri dari citra longsor dan bukan longsor. Didapatkan dari kombinasi pengambilan gambar oleh UAV/Drone dan dari search engine seperti Google dan Bing. Untuk mempersiapkan dataset, peneliti menggunakan VGG Image Anotator untuk melakukan anotasi pelabelan pada dataset. Arsitektur backbone yang digunakan pada penelitian ini adalah ResNet-101. Hasil eksperimen diklaim cukup memuaskan karena metode yang diusulkan mencapai presisi sama dengan 1.00 dan nilai recall 0.93 dengan kondisi di mana jumlah dataset foto longsor yang digunakan sebagai data pelatihan dianggap relatif sedikit. Untuk contoh hasil penelitian ini dapat dilihat pada Gambar 2.7.



Gambar 2.7 Hasil Segmentasi Longsor

Sumber: (Ullo et al., 2021)

Pada penelitian yang dilakukan oleh (Qiao et al., 2019), membahas tentang pemanfaatan kemampuan segmentasi *Mask* R-CNN untuk melakukan segmentasi pada gambar ternak untuk memberikan informasi kesehatan dan kesejahteraan ternak. Dataset yang digunakan sebanyak 1200 gambar ternak yang ukurannya dikurangi menjadi 960 × 540 dan dari 988 gambar digunakan sebagai dataset pelatihan dan 200 gambar lainnya merupakan dataset pengujian. Menurut hasil eksperimen, pendekatan yang diusulkan dapat membuat kinerja segmentasi ternak yang cukup diinginkan dengan mAP sebesar 0,92. Penelitian ini juga menunjukkan bahwa peningakatan kualitas gambar dataset yang didapat seperti meningkatkan kecerahan dan mengurangi bayangan mempengaruhi hasil segmentasi. Untuk hasil segmentasi hewan ternak pada penelitian ini dapat dilihat pada Gambar 2.8.



Gambar 2.8 Hasil Segmentasi Hewan Ternak
Sumber: (Qiao et al., 2019)

Penelitian yang dilakukan oleh (Cao et al., 2023), membahas tentang penggunaan model YOLOv7 untuk mengusulkan pendekatan segmentasi semantik terhadap gambar pohon. Dalam penelitian ini, digunakan 500 gambar pohon berdiri dari penelitian Lin'an dan Lishui. Dataset dibagi menjadi 70% untuk set pelatihan, 20% untuk set validasi, dan 10% untuk set pengujian. Data diproses dengan memperbesar ukuran gambar menjadi 1280 x 1280 piksel dan ditingkatkan dengan operasi “enhancement” untuk meningkatkan kualtias data. Evaluasi dilakukan dengan menggunakan model YOLOv7 dan menghasilkan akurasi tertinggi pada learning rate 0,0002 dan epoch 200. Hasil segmentasi menunjukkan bahwa model ini efektif dalam memisahkan pohon dari latar belakang dengan IoU sebesar 91,17% dan MPA (Pixel Accuracy) sebesar 94.69%. Untuk hasil segmentasi semantik dari penelitian ini dapat dilihat pada Gambar 2.9.



Gambar 2.9 Hasil Segmentasi Pohon

Sumber: (Cao et al., 2023)

2.9 Pemilihan Model Segmentasi

Dalam penelitian ini berfokus pada pengajuan pemanfaatan metode segmentasi untuk mengawasi aktivitas gunung berapi. Untuk mencapai tujuan tersebut maka dipilih dua buah model *deep learning* yaitu You Look Only Once version 7 (YOLOv7) dan Mask Region-Based Convolutional Neural Network (*Mask* R-CNN).

Model YOLOv7 dan *Mask* R-CNN dipilih sebagai model utama untuk mencapai tujuan penelitian ini karena keduanya telah terbukti efektif dalam melakukan tugas deteksi objek dan memiliki kemampuan segmentasi objek pada data citra.

Pertama, untuk model YOLOv7 berdasarkan artikel dari (Sun et al., 2023), seperti model YOLO pengembangan sebelumnya, memiliki desain *one-stage*, di mana model YOLOv7 untuk melakukan deteksi dan segmentasi hanya memerlukan satu kali pemrosesan pada satu layer untuk bisa menemukan objek pada citra. Dengan desain tersebut YOLOv7 diklaim sebagai model yang memiliki kemampuan deteksi yang cepat, efisien, dan menggunakan sumber daya yang sedikit untuk bisa melakukan segmentasi. Model YOLOv7 juga merupakan versi terbaru dari seri model YOLO yang dirilis pada 2022.

Kedua, untuk model *Mask* R-CNN berdasarkan artikel dari (He et al., 2017), Merupakan pengembangan dari model *Fast* R-CNN, namun ditingkatkan dengan penambahan RoI-Allign untuk meningkatkan keakuratan dalam memprediksi deteksi objek. Berbeda dengan model YOLO, pada model *Mask* R-CNN menggunakan konsep desain *two-stage*, di mana model untuk melakukan deteksi dan segmentasi objek harus melalui dua tahap pada dua layer

arsitektur berbeda. Pada *Mask* R-CNN model mengajukan terlebih dahulu kemungkinan keberadaan objek pada Region Proposal Network (RPN) yang baru kemudian masuk ke layer RoI-Allign untuk menyatukan semua informasi peta fitur hasil konvolusi pada layer backbone dan hasil RPN untuk mendapatkan output akhir dari deteksi atau segmentasi. Model *Mask* R-CNN dikembangkan pada tahun 2017.

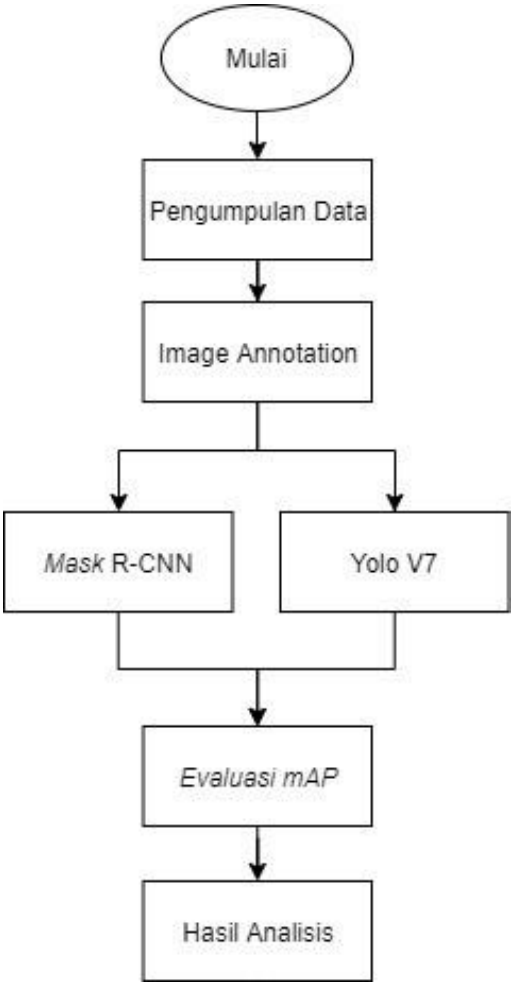
Tabel 2.1 Analisis Penelitian Terdahulu

No	Penulis	Judul	Model	Dataset	Analisis hasil
1.	Zhou et al., (2023)	A Method of Polished Rice Image Segmentation Based on YOLACT for Quality Detection doi: https://doi.org/10.1371/journal.pone.0282297	YOLOv5 dan YOLACT	Dataset buatan sendiri terkumpul 1200 gambar beras. 960 untuk train, 120 untuk validasi, dan 120 untuk test.	Nilai segmentasi citra didapat mAP sebesar 92.1%
2.	Lawal, (2023)	YOLOv5-LiNet: A Lightweight Network for Fruits Instance Segmentation doi: https://doi.org/10.1371/journal.pone.0282297	YOLOv5-LiNet	Data gambar dari kebun buah di China dengan total 2469 gambar. Yang dibagi menjadi 80% untuk train, 15% untuk validasi, dan 5% untuk test.	Didapat nilai instance segmentation pada mAP sebesar 0.885.
3.	Ullo et al., (2021)	A New *Mask* R-CNN-Based Method for Improved Landslide Detection doi: 10.1109/jstars.2021.3064981	*Mask* R-CNN	Jumlah 160 dataset digunakan, terdiri dari citra longsor dan bukan longsor. Didapatkan dari kombinasi pengambilan gambar oleh UAV/Drone dan dari search engine seperti Google dan Bing.	Dengan menggunakan ResNet 101 sebagai backbone, hasil penelitian didapat nilai precision setara dengan 1.00, dan recall 0.93
4.	Qiao et al., (2019)	Cattle segmentation and contour extraction based on *Mask* R-CNN for precision livestock farming doi: 10.1016/j.compag.2019.104958	*Mask* R-CNN	Terdapat 1200 gambar ternak, dari 988 gambar digunakan sebagai dataset pelatihan dan 200 gambar lainnya merupakan dataset pengujian.	Didapat nilai mAP sebesar 0.92

5.	(Cao et al., 2023)	The Semantic Segmentation of Standing Tree Images Based on the YOLOv7 Deep Learning Algorithm doi: https://doi.org/10.3390/electronics12040929	YOLOv7	500 gambar pohon dengan pembagian 70% untuk test, 20% validasi, dan 10% untuk pengujian.	Dengan IoU 91.17% dan MPA (mean pixel accuracy) sebesar 94.69%.

BAB III
METODOLOGI PENELITIAN

Adapun alur yang dilakukan dalam mengerjakan penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.1 Pengumpulan Data

Dalam melakukan pengumpulan data, dibuat sebuah dataset berupa foto-foto aktivitas erupsi gunung berapi yang dikumpulkan dari tiga sumber, yaitu website MAGMA Indonesia (*Multiplatform Application for Geohazard Mitigation and Assessment in Indonesia*) dengan sumber https://magma.esdm.go.id merupakan *website* yang menyediakan data daftar aktivitas kebumian seperti gunung berapi di bawah naungan Pusat Vulkanologi dan Mitigasi Bencana geologi (PVMBG).

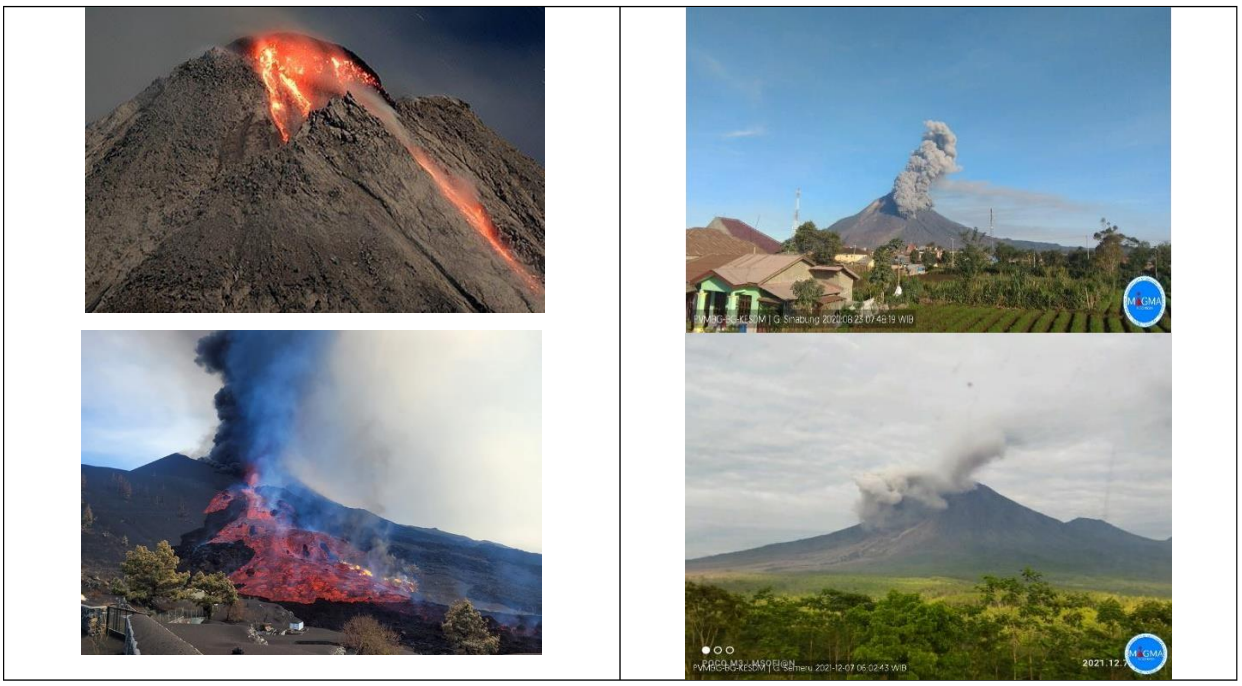
Sumber kedua dengan pencarian di Google Image dengan kata kunci “lava”, “gunung berapi” dan “volcano”, dan pengambilan gambar pada tiap frame dari video Youtube yang menampilkan aktivitas vulkanis pada gunung dengan kata kunci pencarian “lava” dan “volcano”. Dari hasil pengumpulan dataset didapat 140 gambar dengan total 47 gambar diambil dari MAGMA dan 93 gambar diambil dari Google Image yang diambil secara manual. Ke-140 gambar tersebut setelah dihimpun dan dijadikan sebuah dataset akan digunakan untuk melatih model YOLOv7 dan *Mask* R-CNN.

Kriteria Pemilihan Data Citra

Dalam pemilihan data gambar, tidak semua gambar yang ada pada kedua sumber digunakan. Akan tetapi hanya beberapa gambar yang memenuhi kriteria saja yang digunakan. Adapun kriteria dari data citra yang digunakan adalah:

1. Menunjukkan aktivitas vulkanisme pada siang hari.
2. Menampakkan kemunculan lava.
3. Menampakkan kemunculan awan vulkanik.
4. Merupakan gambar yang diambil dari jarak jauh untuk menangkap gambaran visual situasi areal lanskap gunung berapi

Adapun sampel dataset yang diambil dari dua sumber website MAGMA dan pencarian Google Image dapat dilihat pada Gambar 3.2 (a) dan (b).





(a) (b)

Gambar 3.2 Sampel Dataset Gunung Berapi

(a) Sumber Google Image, (b) Sumber Website Magma

Pada gambar 3.2 dapat dilihat meskipun sumber data berbeda, jenis informasi yang bisa dilihat dari kedua sumber tidak berbeda secara signifikan. Baik gambar dari MAGMA maupun Google Image, keduanya menampilkan aktivitas vulkanik atau gunung berapi yang sedang erupsi. Secara kualitas pada gambar yang diambil dapat dianalisis dari metadata yang terdapat dari setiap gambar. Metadata adalah informasi tentang gambar tersebut, ukuran file dalam kilobyte (KB) dan ukuran piksel yang akan menjadi pembanding kualitas.

Tabel 3.2 Metadata Gambar

	Google Image	MAGMA
Ukuran File	608 KB – 69.6 KB	471 KB – 198 KB
Ukuran Piksel	1920x1080 piksel – 512x341 piksel	1000x750 piksel – 1000x562 piksel

Berdasarkan Tabel 3.2 dapat diketahui bahwa ukuran file dengan rentang yang terpaut jauh, dari 608KB hingga 69.9 KB, dan juga piksel dari gambar yang diambil Google Image lebih beragam. Sedangkan untuk gambar dari website MAGMA memiliki ukuran file dengan rentang yang tidak terlalu jauh dan memiliki ukuran piksel atau resolusi yang stabil. Angka

1000 menunjukkan bahwa dataset MAGMA memiliki jumlah piksel horizontal atau lebar yang selalu sama. Hal ini disebabkan karena sumber gambar Google Image berasal dari berbagai website yang dikelola oleh pemilik yang berbeda, sedangkan MAGMA merupakan website yang dikelola oleh institusi negara dan memiliki konsistensi dari segi kualitas gambar yang diunggah.

3.2 Anotasi Dataset

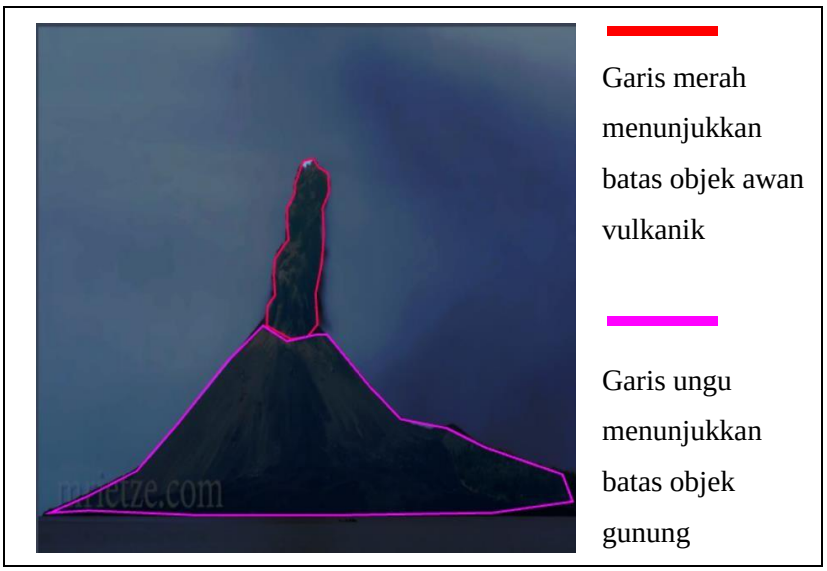
Setelah proses pengumpulan data citra, langkah selanjutnya adalah tahap anotasi pada dataset. Anotasi citra merupakan proses pelabelan gambar yang terdapat pada dataset yang dimiliki. Jumlah dan nama label mengikuti objek kelas yang menjadi batasan penelitian ini, yaitu “lava”, “awan_vulkanik”, “gunung” dan “vegetasi”. Pada tahap ini, proses anotasi dilakukan menggunakan dua *tools*, yaitu Roboflow dan Makesense.ai yang masing-masing digunakan untuk model YOLOv7 dan *Mask* R-CNN. Penggunaan kedua *tools* yang berbeda ini mengikuti kebutuhan dari kode program sesuai dengan referensi yang menjadi rujukan model yang digunakan pada penelitian ini. *Mask* R-CNN menggunakan sumber https://pysource.com/ sebagai rujukan pengerjaan bagaimana dataset *Mask* R-CNN dipersiapkan, dan untuk YOLOv7 menggunakan sumber https://github.com/WongKinYiu/YOLOv7, kedua sumber memberikan referensi terhadap kedua tools beserta penjelasan tahapan atau proses yang dilakukan untuk mempersiapkan dataset.

Setelah proses anotasi dilakukan, hasil ekspor Makesense.ai didapatkan file berekstensi JSON untuk *Mask* R-CNN yang berisikan informasi label kelas dan label segmentasi. Untuk YOLOv7 didapatkan file berekstensi txt yang berisikan label kelas dan titik koordinat segmentasi, yang didapatkan setelah mengekspor hasil anotasi dari Roboflow. Dari kedua *tools* tersebut terdapat fitur *polygon labelling* untuk membuat batas-batas objek dari sebuah citra yang sesuai dengan target kelas objek yang akan disegmentasi.

Polygon Labelling

Dalam *polygon labelling*, anotasi yang digunakan untuk menunjukkan area objek berupa segmen, bukan hanya kotak yang menandakan lokasi objek. Penandaan dilakukan dengan membuat batas atau garis yang mengelilingi objek dengan lebih presisi daripada bounding box. Dengan menggunakan *polygon* labeling, model dapat mempelajari fitur dan bentuk setiap objek, dan menghasilkan segmentasi yang lebih akurat dan detail pada objek

yang diidentifikasi sehingga menciptakan batas antara suatu objek dengan objek lain, contoh hasil sebuah anotasi *polygon labelling* dapat dilihat pada Gambar 3.3.

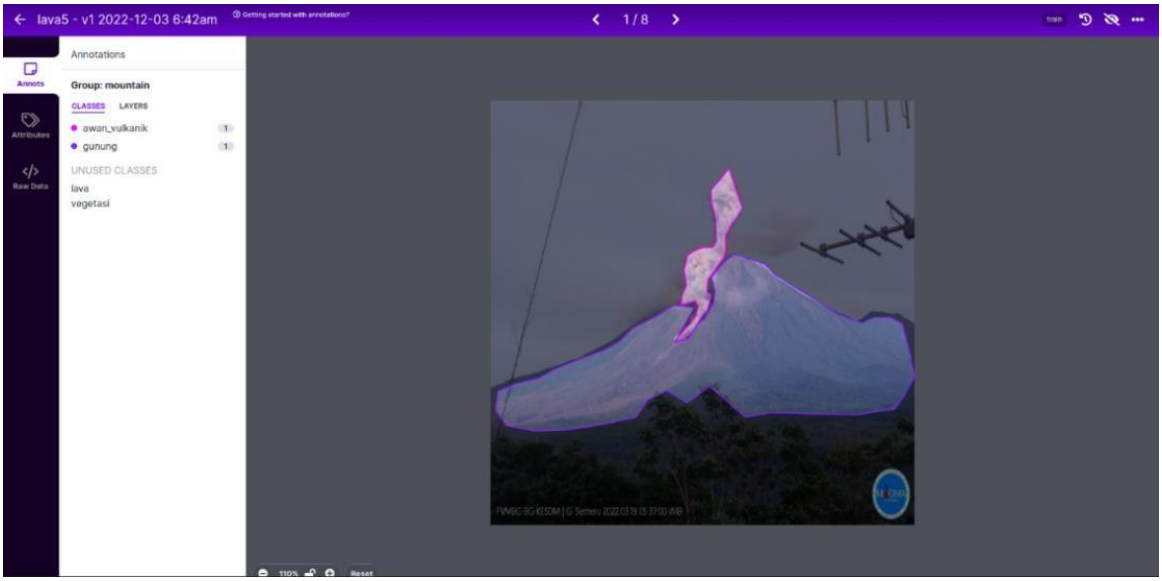


Gambar 3.3 Hasil *Polygon Labelling*

Dari Gambar 3.3 dapat dilihat bahwa terdapat batas yang jelas antara awan vulkanik dengan gunung, secara intuisi manusia dapat membedakan keduanya. Namun, pada proses komputasi untuk melakukan segmentasi memerlukan kejelasan informasi mengenai keduanya.

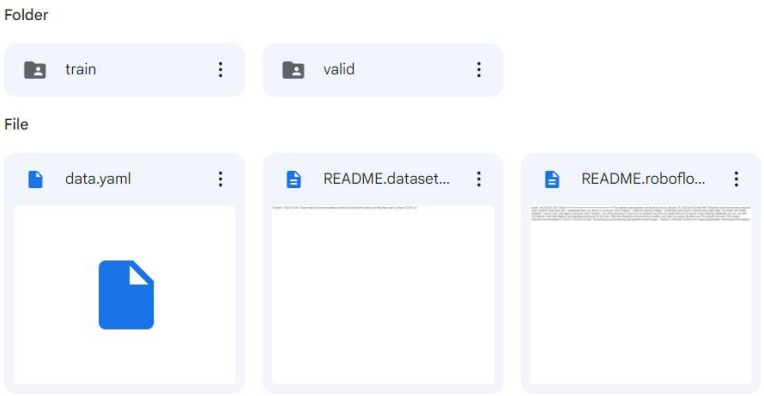
3.2.1 Anotasi Dataset Model YOLOv7

Proses anotasi dataset untuk model YOLOv7 menggunakan *tools* Roboflow yang dapat diakses melalui https://app.roboflow.com/. Fitur dari Roboflow yang digunakan pada penelitian ini diantaranya adalah melakukan *annotate* atau menandai objek yang akan disegmentasi menggunakan *polygon labelling*. Penandaan dilakukan dengan membuat batas atau garis yang mengelilingi area awan vulkanik, sehingga hasil anotasi yang didapatkan adalah area berupa segmen yang terususun atas titik-titik koordinat yang mengelilingi awan vulkanik, setelah dilakukan anotasi segmentasi kemudian diberikan label sesuai kelas objeknya.



Gambar 3.4 Proses Anotasi Citra pada Roboflow

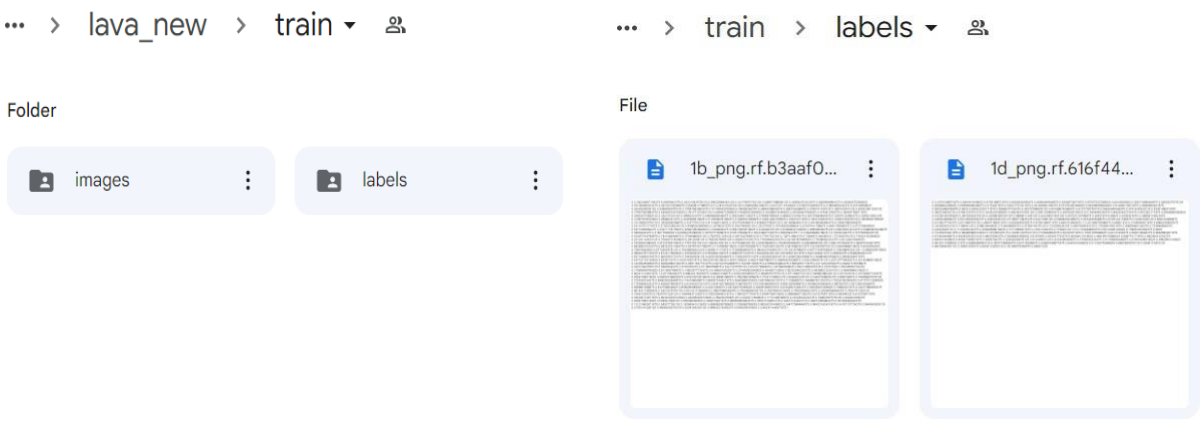
Gambar 3.4 merupakan contoh dilakukannya anotasi terhadap sebuah citra yang akan dianalisis. Pada contoh pelabelan gambar tersebut, terdapat dua objek kelas yang disegmentasi, yaitu objek kelas awan_vulkanik dan gunung. Hasil dari dataset yang telah dianotasi dari Roboflow akan diekspor untuk digunakan pada kode program. Hasil ekspor akan berupa file zip yang diekstrak secara manual kemudian disimpan di Google Drive, folder hasil unzip berisikan dua buah folder dan tiga buah file. Kedua folder terdiri atas folder train dan folder valid, file data.yaml dan dua buah file README. Untuk gambaran penyimpanan hasil anotasi pada YOLOv7 dapat dilihat pada Gambar 3.5.



Gambar 3.5 Folder Hasil Anotasi Roboflow

Gambar 3.5 menunjukkan folder hasil dari proses anotasi Roboflow yang didapat sebuah folder train dan valid yang keduanya berisikan dataset dan hasil label anotasinya, dan

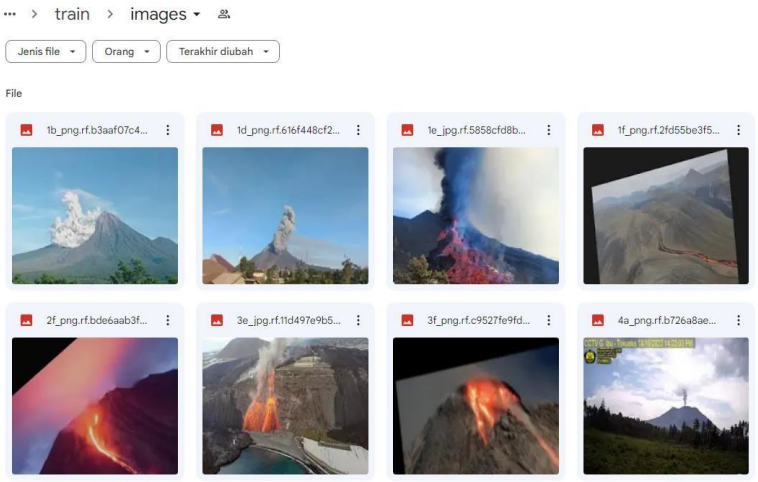
sebuah file berekstensi YAML yang merupakan format penyimpanan struktur data yang akan dibaca oleh model YOLOv7. Jumlah gambar dan label txt mengikuti jumlah dataset yang dimasukkan ke tahap train dan validasi, setiap gambar akan memiliki satu file txt yang menunjukkan informasi area segmen dan label kelasnya, pembagian dataset YOLOv7 dapat dilihat di Gambar 3.15. Untuk detail dari isi folder train dan valid dapat dilihat pada Gambar 3.6.



(a) (b)

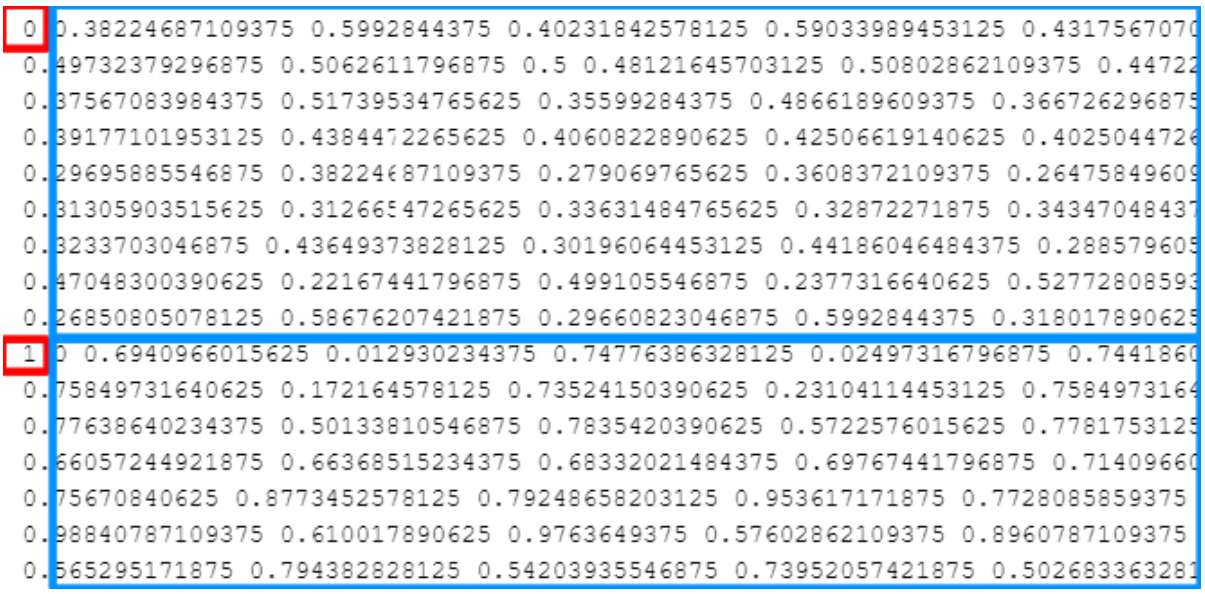
Gambar 3.6 (a) Struktur Folder Train Hasil Anotasi. (b) Isi Folder Labels

Pada Gambar 3.6 (a) dan (b) menunjukkan struktur dan isi folder yang didapatkan ketika mengekspor hasil anotasi dari *tools* Roboflow. Baik folder train dan valid keduanya berisikan folder images dan labels. Folder images berisikan dataset kumpulan citra yang digunakan, untuk folder train berarti berisikan citra yang digunakan saat proses pelatihan dan validasi untuk folder images di folder valid. Contoh isi dari folder images dapat dilihat pada Gambar 3.7.



Gambar 3.7 Isi Folder Images

Folder labels terdiri dari kumpulan file berekstensi .txt yang berisikan titik-titik koordinat dari segmentasi objek yang teridentifikasi sebagai objek kelas dan angka yang mengidentifikasikan kelas objeknya. Untuk contoh isi file txt yang terdapat pada folder labels di folder train dan valid dapat dilihat pada Gambar 3.8.



Gambar 3.8 Isi File .txt pada Folder Labels

Struktur informasi dari file txt terdiri dari dua informasi, yaitu identifikasi kelas objek yang ditunjukkan pada kotak merah, dan titik-titik koordinat yang membentuk area segmentasi ditunjukkan dengan kotak biru.

```
train: ../train/images
val: ../valid/images

nc: 4
names: ['awan_vulkanik', 'gunung', 'lava', 'vegetasi']

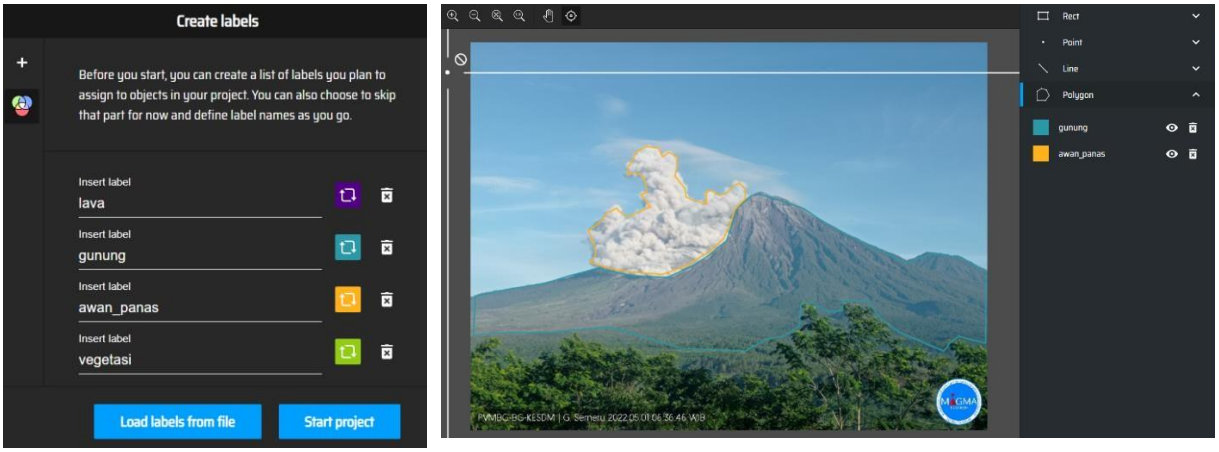
roboflow:
  workspace: universitas-xoefw
  project: xyz-l8a54
  version: 5
  license: CC BY 4.0
  url: https://universe.roboflow.com/universitas-xoefw/xyz-l8a54/dataset/5
```

Gambar 3.9 File YAML Hasil Anotasi Roboflow

Pada Gambar 3.9 merupakan isi dari file berekstensi YAML. File ini digunakan oleh kode program YOLOv7 untuk mengakses informasi mengenai dataset dan konfigurasi pelatihan model. Informasi yang disampaikan berupa lokasi folder yang berisikan data citra *train* dan *val*, nc bernilai empat menunjukkan jumlah kelas objek yang digunakan dan dijelaskan daftar objek kelasnya pada list names.

3.2.2 Anotasi Dataset Model *Mask* R-CNN

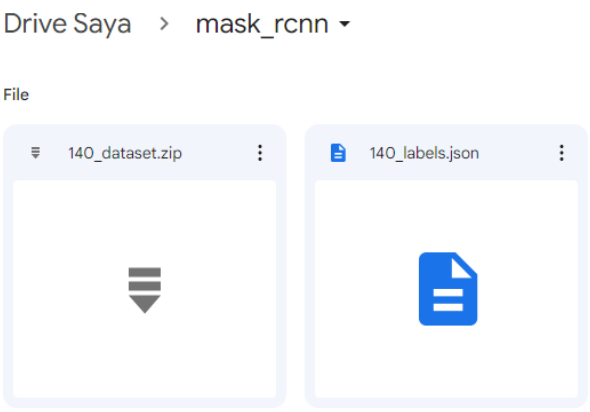
Pada proses persiapan anotasi dataset untuk model *Mask* R-CNN, dilakukan dengan menggunakan makesense.ai yang dapat diakses di https://www.makesense.ai/. Pada tahap ini menggunakan metode yang sama, yaitu *polygon labelling*. Secara umum proses pelabelan dataset menggunakan makesense.ai mirip dengan Roboflow di mana dilakukan pemberian label dan melakukan anotasi dengan *polygon labelling* ke setiap objek kelas yang teridentifikasi.



(a) (b)

Gambar 3.10 (a) Pembuatan Label Kelas. (b) Proses *Labelling*

Pada Gambar 3.10 (a) menunjukkan pembuatan label kelas dengan makesense.ai, proses ini dilakukan menyesuaikan dengan jumlah objek kelas yang ingin diidentifikasi, sedangkan Gambar 3.10 (b) menunjukkan proses pelabelan dimana menggunakan *tools* polygon. Setelah memberikan label dan anotasi segmen ke seluruh gambar, tahap selanjutnya adalah melakukan ekspor hasil anotasi yang akan didapatkan file berekstensi JSON yang berisikan keterangan objek kelas dan titik-titik koordinat area label poligon, file bernama "140_labels.json".



Gambar 3.11 Isi Folder Hasil Anotasi Makesense.ai

Selanjutnya menyimpan seluruh kebutuhan model *Mask* R-CNN berupa file label JSON dan 140 gambar dataset yang disimpan dalam bentuk zip ke Google Drive, struktur penyimpanan file dan folder dataset dapat dilihat pada Gambar 3.11. Pada penelitian ini folder drive diberi nama "mask_rcnn" penamaan folder ini akan digunakan untuk dipanggil di Google Colab. Sedangkan contoh isi dari file JSON hasil eksportasi dari Makesense.ai dilihat pada Gambar 3.12.


```
{"id":184,"iscrowd":0,"image_id":98,"category_id":1,"segmentation":[[295.2830188679
245,270.8679245283019,301.8867924528302,274.64150943396226,308.49056603773585,275.5
849056603774,. . .
,325.47169811320754,273.6981132075472,313.20754716981133,267.09433962264154,302.830
1886792453,264.26415094339626,300,267.09433962264154]],"bbox":[295.2830188679245,26
4.26415094339626,100,29.245283018867894],"area":744.0370238519046,"area":13672.0541
11783647}],"categories":[{"id":1,"name":"lava"},{"id":2,"name":"awan_vulkanik"},{"i
d":3,"name":"gunung"},{"id":4,"name":"vegetasi"}]
```


Gambar 3.12 File JSON Hasil Anotasi Makesense.ai

Gambar 3.12 menunjukkan isi dari file JSON hasil dari proses anotasi makesense.ai yang akan digunakan pada model *Mask* R-CNN. File ini menyimpan informasi penting mengenai anotasi yang telah dilakukan, ID objek ditandai oleh "`id:184`" yang menandakan bahwa objek tersebut merupakan objek ke-184, informasi apakah objek tersegmentasi sebagai kelompok atau objek individu ditandai oleh "`iscrowd`" bernilai boolean 0 berarti objek terdeteksi tunggal, ID gambar yang ditunjukkan oleh "`image_id:98`" menunjukkan bahwa gambar terbaca sebagai gambar ke-98 dari keseluruhan dataset, kategori objek yang ditunjukkan oleh nilai "`category_id:1`" bermakna bahwa objek dengan ID 184 terdeteksi sebagai objek kelas satu yang dapat dilihat pada list kategori "`categories":[{"id":1,"name":"lava"},{"id":2,"name":`

"awan_vulkanik"},{"id":3,"name":"gunung"},{"id":4,"name":"vegetasi"}
]" yang menandakan objek ID 184 adalah lava. Sementara nilai dari "segmentation" merupakan koordinat titik-titik yang membentuk polygon area objek.

3.3 Data Latih dan Validasi

Setelah melakukan proses anotasi untuk kebutuhan kedua model, tahap selanjutnya adalah menyesuaikan kebutuhan model untuk proses pelatihan atau *training*. Pada tahap ini pembagian dataset dilakukan menyesuaikan dengan kondisi dan kebutuhan sistem dari kedua model yang digunakan. Pembagian dataset kedua model memiliki perbedaan dari segi jumlah untuk tahap pelatihan dan validasi, untuk Mask R-CNN 125 gambar digunakan untuk train dan 15 gambar untuk validasi, sedangkan YOLOv7 110 gambar untuk train dan 30 gambar untuk validasi.

Pembagian Data *Mask* R-CNN

Untuk model *Mask* R-CNN pembagian dataset mengikuti kebutuhan dari model yang digunakan. Di mana data yang masuk secara default dari program akan dibagi menjadi 90% data *training* dan 10% data validasi. Detail kode program yang membagi dataset dapat dilihat pada Gambar 3.13.

```
dataset_train, dataset_val, class_number = load_image_dataset(PROJECT_NAME)
```

```
Annotation path /content/gdrive/MyDrive/pysource_mrcnn_pro/mask_rcnn
Found 1 annotation files
125 train images
15 validation images
4 classes
```

Gambar 3.13 Pembagian Dataset Pada *Mask* R-CNN

Dari hasil menjalankan kode program pada Gambar 3.13 mendapatkan hasil pembagian dataset yang sudah ditentukan dari kode program *Mask* R-CNN yang digunakan pada penelitian ini. Detail bagaimana pembagian dataset dilakukan dapat dijelaskan pada Gambar 3.14.

```
# Split the dataset, if train, get 90%, else 10%
        len_images  =  len(coco_json['images'])
        if dataset_type == "train":
            img_range = [int(len_images / 9), len_images]
        else:
            img_range = [0, int(len_images / 9)]
```

```
        for i in range(img_range[0], img_range[1]):
            image  =  coco_json['images'][i]
            image_id = image['id']
            if image_id in seen_images:
                print("Warning: Skipping duplicate image id:
{}".format(image))
            else:
                seen_images[image_id] = image
                try:
                    image_file_name = image['file_name']
                    image_width = image['width']
                    image_height = image['height']
                except KeyError as key:
                    print("Warning: Skipping image (id: {}) with missing
key: {}".format(image_id, key))

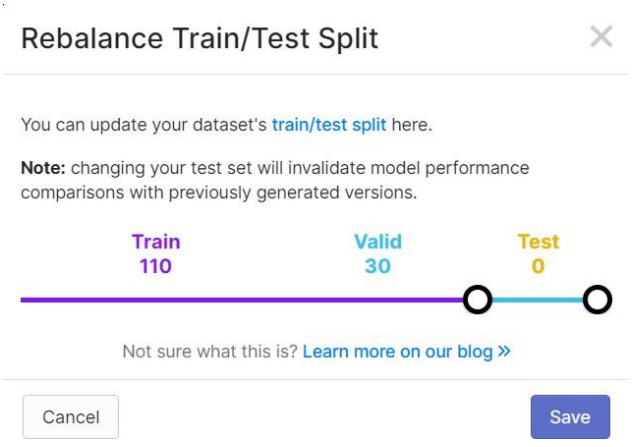
                image_path = os.path.abspath(os.path.join(images_dir,
image_file_name))
                image_annotations = annotations[image_id]
```

Gambar 3.14 Setting Default Pembagian Dataset Mask R-CNN

Secara garis besar, kode program pada Gambar 3.14 akan melakukan pembagian keseluruhan dataset yang terbaca ke dalam tipe dataset "`train`" atau dijadikan dataset pelatihan dan sisanya akan digunakan untuk proses validasi, di mana nilai train ditentukan sebanyak 90%.

Pembagian Data YOLOv7

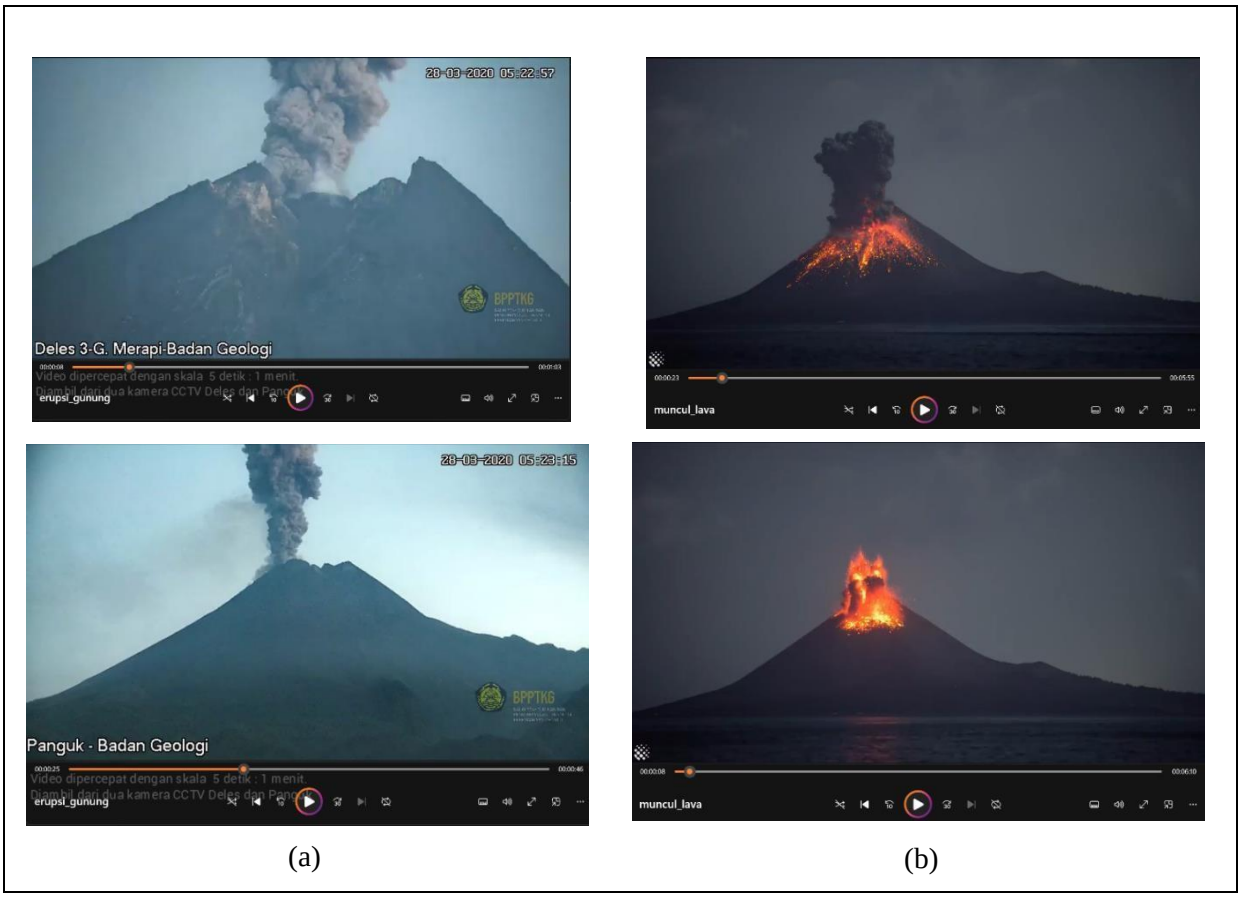
Untuk model YOLO, pembagian dataset memiliki kebebasan dalam menentukan konfigurasi berapa jumlah data training atau validation karena menggunakan bantuan *tools* Roboflow yang bisa membantu manajemen dataset. Untuk penelitian ini dipillih konfigurasi 110 data citra untuk pelatihan dan 30 untuk validasi. Detail pembagian dataset dapat dilihat pada Gambar 3.15.



Gambar 3.15 Pembagian Dataset pada Roboflow

3.4 Data Uji

Tujuan pengujian untuk melihat kemampuan model yang telah dilatih terhadap situasi yang belum pernah dilihat sebelumnya atau melihat kemampuan model dalam kondisi *real time*. Dalam pengujian (*test*) model digunakan data dua buah video yang menampilkan aktivitas vulkanisme sebuah gunung berapi. Pengujian data video untuk melihat kemampuan model dalam melakukan segmentasi dan deteksi *real time* kondisi lapangan pengamatan gunung berapi. Data video diambil dari dua sumber; Video 1 didapat dari akun YouTube BPPTKG (https://www.youtube.com/@BPPTKG/videos) yang menampilkan pengamatan Gunung Merapi secara *live stream*, untuk Video 2 merupakan Gunung Krakatau yang erupsi dan direkam oleh fotografer Martin Rietze dari kanal YouTube-nya (https://www.youtube.com/watch?v=9Plj4b79cwg). Untuk data video dapat dilihat dalam cuplikan frame keduanya pada Gambar 3.16.



(a) (b)

Gambar 3.16 Tangkapan Frame Data Uji Video

(a) Video 1 – video_merapi.mp4, (b) Video 2 – Krakatoa_lava.mp4

Pada Gambar 3.16 menunjukkan tangkapan dua buah video yang menampakkan aktivitas vulkanisme. Pada Gambar 3.16 (a) menampilkan sebuah gunung yang mengeluarkan

awan vulkanik yang merupakan video gunung Merapi dari kanal BPPTKG yang menampakan awan vulkanik, dan pada Gambar 3.16 (b) merupakan video kedua dari gunung Krakatau yang memunculkan lava.

Tabel 3.3 Metadata Video Data Uji

	Video 1 - video_merapi.mp4	Video 2 – krakatoa_lava.mp4
Durasi	00:01:11	00:06:18
Ukuran File	2,98 MB	12,7 MB
Ukuran Piksel	640x480	1280x720
Frame per Second	30 fps	29,97 fps

Tabel 3.3 menunjukkan karakteristik dari kedua data video yang digunakan sebagai data uji untuk melihat kemampuan dari hasil mode Mask R-CNN dan YOLOv7.

3.5 Persiapan Model Segmentasi

Pada tahap persiapan model untuk melakukan segmentasi, dilakukan konfigurasi terhadap kode program untuk menyesuaikan kondisi dan keperluan model ketika akan melakukan pelatihan. Pada proses ini dilakukan dua tahap, yaitu konfigurasi pada kode program *Mask* R-CNN dan YOLOv7.

Dalam menentukan konfigurasi model atau dalam model deep learning dikenal juga sebagai *finetuning hyperparameter*, merupakan proses penting yang dilakukan untuk mendapatkan konfigurasi terbaik yang menghasilkan model deep learning paling optimal. Dalam menentukan konfigurasi kedua model dilakukan beberapa kali percobaan untuk bisa mendapatkan nilai terbaik.

3.5.1 Konfigurasi YOLOv7

Tahap pengaturan pada YOLOv7 melibatkan konfigurasi pada *batch size*, jumlah *epochs*, dan *image size*. Pemilihan parameter ini mempertimbangkan sumber daya yang tersedia dan hasil pelatihan paling optimal. Untuk detail konfigurasi yang digunakan dapat dilihat pada Tabel 3.4.

Tabel 3.4 Finetuning Model YOLOv7

Parameter yang Disesuaikan	Nilai yang Diuji	Jumlah Percobaan
Batch	8, 16	4 Kali
Epoch	80, 90, 100	

Pada Tabel 3.4 menunjukkan informasi mengenai berapa kali model YOLOv7 diujicoba untuk mendapatkan nilai Batch dan Epoch terbaik. Jumlah percobaan untuk YOLOv7 dilakukan lebih sedikit dikarenakan setiap percobaan model YOLOv7 akan mengambil model terbaik secara otomatis dan setelah pengujian pada nilai epoch 100 dan batch 16 tidak terjadi perubahan signifikan, maka pada parameter tersebut dianggap sebagai nilai terbaik. Untuk angka konfigurasi paling optimal didapatkan nilai pada Tabel 3.5.

Tabel 3.5 Parameter Settings YOLOv7

Parameter	**Nilai Setting**
Batch	16
Epochs	100
Image Size	640

Tabel 3.5 menunjukkan beberapa parameter terbaik yang digunakan dalam pelatihan model. Ukuran gambar (Image Size) adalah ukuran gambar yang digunakan dalam pelatihan model. Pada tabel ini, ukuran gambar yang digunakan adalah 640 piksel dan tidak dilakukan penyesuaian dikarenakan merupakan rekomendasi dari setting default https://github.com/WongKinYiu/yolov7. Ukuran gambar dapat mempengaruhi kinerja model, di mana gambar yang lebih besar akan menghasilkan waktu pelatihan yang lebih lama dan memerlukan lebih banyak sumber daya komputasi. *Batch size* (ukuran batch) adalah jumlah gambar yang akan diproses dalam satu iterasi. Pada tabel ini, pengaturan Batch size adalah 16, yang artinya 16 gambar akan diproses dalam satu iterasi. Ukuran batch yang lebih besar dapat mempercepat proses pelatihan karena memanfaatkan sumber daya komputasi yang lebih efisien, namun memerlukan memori yang lebih besar. Sebaliknya, ukuran batch yang terlalu kecil memerlukan banyak iterasi untuk menyelesaikan satu epoch, tetapi menggunakan memori yang lebih sedikti. *Epochs* adalah jumlah berapa kali model akan melatih seluruh dataset. Pada tabel ini, pengaturan Epochs adalah 100, artinya model akan melatih seluruh dataset sebanyak 100 kali. Jumlah epoch yang besar dapat meningkatkan akurasi, tetapi juga memerlukan waktu pelatihan yang lama.

```
!python train.py --data ../Dataset/lava_new/data.yaml --batch 16 --
weights ../YOLOv7-seg.pt --epochs 100 --name ../YOLOv7-seg --img 640 --
hyp ../hyp.scratch-high.yaml
```

Gambar 3.17 Baris Kode Setting pada YOLOv7

Seperti ditunjukkan pada Gambar 3.17 yang menampilkan konfigurasi *epoch*, *batch size*, dan ukuran gambar yang telah di-*setting* untuk menjalankan proses pelatihan pada YOLOv7. Dengan konfigurasi ini, model YOLOv7 akan dilatih dengan parameter yang telah disesuaikan dengan kebutuhan sumber daya yang tersedia.

Pre-trained weight

Pada pelatihan model YOLOv7 menggunakan *weight* atau bobot *pre-trained* yang sudah dilatih menggunakan dataset MS COCO (cocodataset.org). Sebagai hasil dari penggunaan *transfer learning* dari pelatihan sebelumnya, dihasilkan file YOLOv7-seg.pt yang digunakan untuk melatih data yang digunakan pada penelitian ini.

```
!wget https://github.com/WongKinYiu/YOLOv7/releases/download/v0.1/YOLOv7-seg.pt
```

Gambar 3.18 Sumber Pre-Trained Weight YOLOv7

Kode pada Gambar 3.18 tersebut melakukan *download* file dengan nama "YOLOv7-seg.pt" dari alamat "https://github.com/WongKinYiu". File YOLOv7-seg.pt merupakan format file yang digunakan oleh PyTorch, salah satu *framework* populer untuk deep learning. File ini menyimpan informasi yang memetakan setiap lapisan bobot hasil pengetahuan yang diperoleh dari pelatihan sebelumnya.

Arsitektur

Model YOLOv7 menggunakan tiga komponen pada arsitektur *backbone*, yaitu CBS, E-ELAN dan MP1 seperti dijelaskan pada sub bab 2.5 YOLO mengenai arsitektur YOLOv7.

3.5.2 Konfigurasi *Mask* R-CNN

Tahap pengaturan pada *Mask* R-CNN melakukan konfigurasi pada *step per epoch*, *epoch* dan *image size*. Pemilihan jumlah *epochs*, *image size*, dan *steps per epoch* mempertimbangkan sumber daya komputasi yang tersedia.

Tabel 3.6 Finetuning Model *Mask* R-CNN

Parameter yang Disesuaikan	Nilai yang Diuji	Jumlah Percobaan
Step per Epoch	20, 30, 55, 65	20 Kali
Epoch	70, 75, 80, 85, 90, 100	
Image Size	256, 512, 768, 832, 1024	

Pada Tabel 3.6 merupakan informasi proses *finetuning* yang dilakukan. Model *Mask* R-CNN dilakukan lebih banyak uji coba untuk mendapatkan hasil pelatihan paling optimal,

dikarenakan model pada epoch terbaik tidak diambil secara otomatis oleh program melainkan harus diamati satu persatu pada tiap epochnya. Setelah diujicoba pada nilai epoch 100 dan step per epoch 65, tidak terdapat perbaikan signifikan dari nilai val_loss dan loss maka dianggap sebagai parameter paling optimal. Untuk detail nilai parameter paling optimal model Mask R-CNN dapat dilihat pada Tabel 3.7.

Tabel 3.7 Parameter Settings *Mask*-RCNN

Parameter	Nilai Setting
Steps_Per_Epoch	65
Epochs	100
Image_Size	512

Tabel 3.7 menyajikan beberapa parameter dan pengaturannya pada suatu model. *Steps Per Epoch* menunjukkan jumlah langkah atau batch yang akan diproses dalam satu epoch, dengan pengaturan parameter 65 pada Tabel 3.7. *Epochs* adalah jumlah iterasi pelatihan keseluruhan dataset, dengan pengaturan parameter nilai 100, yang berarti model akan dilatih selama 100 *epoch* di mana setiap *epoch* akan terjadi *steps* sebanyak 65 kali. Image_Size adalah ukuran gambar yang akan di-*resize* dalam pelatihan model, dengan ukuran 512 piksel pada seluruh dataset, pemilihan ukuran ini untuk tetap menjaga kualitas gambar namun tidak membuat proses komputasi berjalan dengan berat.

```
# SETTINGS
PROJECT_NAME = “mask_rcnn”
STEPS_PER_EPOCH = 65
EPOCHS = 100
IMAGE_SIZE = 512 # Image sizes (256, 512, 768, 832, 1024)
AUGMENTATION = true
```

Gambar 3.19 Baris Kode Settings pada *Mask* R-CNN

Pada Gambar 3.19 menunjukkan parameter/konfigurasi dari *model* Mask R-CNN didefinisikan di kode program pada bagian SETTINGS.

Pre-trained model

Pada model *Mask* R-CNN yang digunakan, menggunakan *pre-trained model* yang sudah ada sebelumnya, yaitu model mask_rcnn_coco.h5 sebagai titik awal konfigurasi yang diperolah dari sumber PySource (psfiles.github). Pre-trained model merupakan model yang telah dilatih sebelumnya pada dataset yang berbeda, dalam hal ini dataset MS COCO (cocodataset.org) digunakan yang mencakup berbagai kategori objek.

Manfaat menggunakan pre-trained model adalah memanfaatkan pengetahuan yang telah diperoleh model tersebut dari proses pelatihan sebelumnya dan mempercepat proses

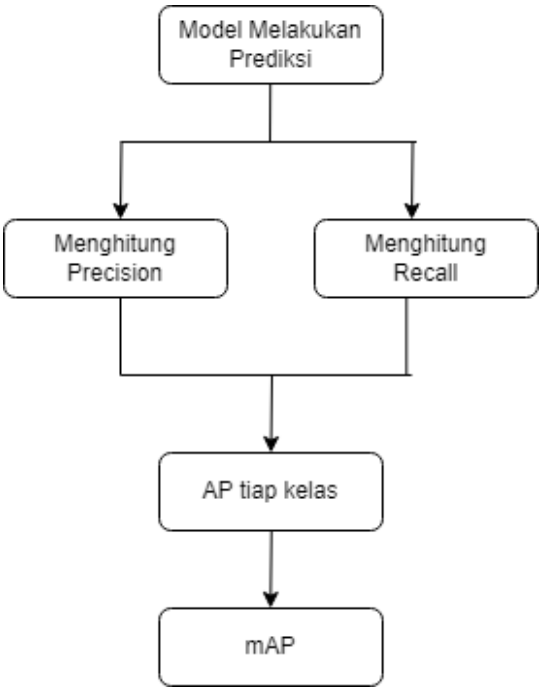
pelatihan pada dataset baru. Karena model telah dilatih pada dataset yang luas dan beragam, model tersebut telah mengembangkan pemaham yang baik tentang membaca fitur dan struktur yang ada dalam citra. Hal ini memungkinkan model untuk lebih mudah mengadaptasi dan mempelajari fitur-fitur yang spesifik pada dataset yang digunakan di penelitian ini.

Arsitektur

Model *Mask* R-CNN yang digunakan pada penelitian ini memanfaatkan *backbone* ResNet-101 sebagai dasar ekstraksi fitur. ResNet101 didapatkan dari sumber referensi repositori *Mask* R-CNN pada penelitian ini yaitu PySource (psfiles.github). *Backbone* merupakan bagian penting dalam arsitektur yang bertanggung jawab untuk memahami fitur penting dari citra yang akan digunakan untuk proses segmentasi.

3.6 Evaluasi Hasil

Dalam melakukan evaluasi hasil, penelitian ini melakukan perhitungan performa model *Mask*-RCNN dan YOLOv7 dengan menggunakan evaluasi dari *mean average precision* (mAP). Nilai dari mAP merupakan salah satu metrik umum yang digunakan dalam evaluasi model segmentasi objek. Metrik ini menggabungkan nilai presisi dan *recall* untuk menghasilakan skor tunggal yang mencerminkan kinerja model. Flowchart pada Gambar 3.16 berikut menunjukkan tahapan dalam melakukan perhitungan Mean Average Precision (mAP).



Gambar 3.20 Flowchart penghitungan nilai mAP

Diagram flowchart pada Gambar 3.20 menunjukkan proses bagaimana nilai *mean average precision* dihitung. Diawali dengan model melakukan prediksi terlibih dahulu terhadap dataset kemudian hasil prediksi akan dihitung nilai *precision* dan *recall*-nya berdasarkan *ground truth* atau label yang sesungguhnya. Hasil setiap *precision* dan *recall* akan digabungkan untuk memberikan gambaran menyeluruh tentang kinerja model dalam mengidentifikasi dan segmentasi objek dalam suatu kelas, proses ini menghasilkan nilai *average precision* (AP) untuk setiap kelas objek. Setelah mendapatkan nilai AP dari setiap kelas objek, untuk mendapatkan hasil metrik evaluasi menyeluruh kinerja model, nilai AP di rata-ratakan untuk mendapat nilai mAP.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Konfigurasi Tensorflow dan CUDA

Agar bisa menjalankan kode program *Mask* R-CNN dan YOLOv7 saat pelatihan maka diperlukan Tensorflow dan CUDA ketika akan menjalankan program di Google Colab. Tensorflow adalah framework open-source yang digunakan untuk mengembangkan dan melatih model deep learning, sedangkan CUDA (Compute Device Architecture) adalah platform komputasi yang dikembangkan oleh NVIDIA untuk memanfaatkan kemampuan komputasi dari pemrosesan GPU (Graphics Processing Unit). Baik *Mask* R-CNN dan YOLOv7 menggunakan keduanya, namun jenis versi yang digunakan berbeda terutama *Mask* R-CNN yang sudah lama tidak ada pengembangan dari sumber aslinya sehingga menggunakan versi Tensorflow dan CUDA yang lama, sedangkan YOLOv7 menggunakan konfigurasi default dari Google Colab. Detail mengenai konfigurasi yang digunakan kedua model dapat dilihat pada Tabel 4.1.

Tabel 4.1 Konfigurasi Tensorflow dan CUDA

	Versi Tensorflow	Versi CUDA
Mask R-CNN	2.5.0	11.2
YOLOv7	2.12.0	11.8

4.2 Implementasi Model *Mask* R-CNN

Situs https://pysource.com/ merupakan sumber asli dari kode program *Mask* R-CNN yang digunakan di penelitian ini. Sumber tersebut berisi tentang tutorial pengembangan *computer vision*, termasuk tutorial penggunaan *Mask* R-CNN. Pada tahap ini data yang sudah disimpan pada Google Drive seperti file data gambar dan label JSON, seperti ditunjukkan pada Sub bab 3.2.2, akan di-*mount* di Google Colab dengan menggunakan source code pada Gambar 4.1.

```
from google.colab import drive
drive.mount('/content/drive')
```

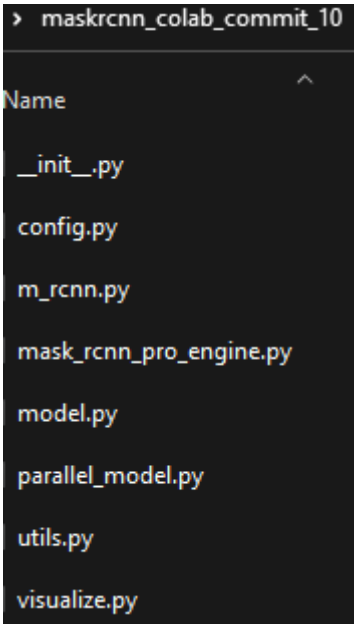
Gambar 4.1 Source Code Koneksi Google Drive

Setelah melakukan import dataset tahapan berikutnya adalah melakukan download dan import library *Mask* R-CNN dengan menggunakan source code pada gambar 4.2.

```
import warnings
warnings.filterwarnings('ignore')
!wget https://pysource.com/extra_files/mask_rcnn_colab_commit_10.zip
!unzip maskrcnn_colab_commit_10.zip
import sys
sys.path.append("/content/Maskrcnn_colab/mrcnn")
from mask_rcnn_pro_engine import *
%matplotlib inline
```

Gambar 4.2 Source Code Download Import Library *Mask* RCNN

Pada *source code* tersebut, dilakukan beberapa tindakan seperti mengimport *library* "warnings" untuk mengabaikan atau menghilangkan warning yang muncul selama proses eksekusi kode. Kemudian mengunduh file zip dengan nama "mask_rcnn_colab_commit_10.zip" menggunakan perintah wget dari situs https://pysource.com/extra_files/mask_rcnn_colab_commit_10.zip. File zip tersebut berisikan berbagai file yang diperlukan untuk menjalankan model *Mask* R-CNN, seperti train.py, model.py,_init_.py, config.py, m_rcnn.py, utils.py, visualize.py, parallel_model.py dan mask_rcnn_pro_engine.py yang berisikan berbagai fungsi yang digunakan di Google Colab. Untuk detail isi dari file zip tersebut dapat dilihat pada Gambar 4.3.



Gambar 4.3 Isi File maskrcnn_colab_commit_10

File zip pada Gambar 4.3 yang diunduh berisi file yang diperlukan untuk menjalankan *Mask* R-CNN pada Google Colaboratory. Kemudian, file yang telah diunduh diekstrak dengan menggunakan perintah "unzip". Perintah sys menambahkan path dari file yang akan diimport ke dalam sistem (sys.path.append). Kemudian semua fungsi yang terdapat pada mask_rcnn_pro_engine.py diimport dari direktori "/content/mask_rcnn_colab/mrcnn" dan

menggunakan matplotlib untuk menampilkan hasil output secara langsung di bawah sel kode yang menghasilkannya.

Tahapan berikutnya adalah melakukan ekstrak terhadap dataset yang telah diimpor, hal ini dilakukan karena dataset yang dimpor masih berekstensi ".zip" sehingga diperlukan ekstraksi terlebih dahulu agar dataset dapat terbaca oleh sistem. Adapun *source code* dari extrask data dapat dilihat pada Gambar 4..

```
images_path = os.path.join(DRIVE_ROOT_DIR, "mask_rcnn/140_dataset.zip")
extract_images(images_path, "/content/dataset")
```

Gambar 4.4 Source Code Ekstrak Dataset

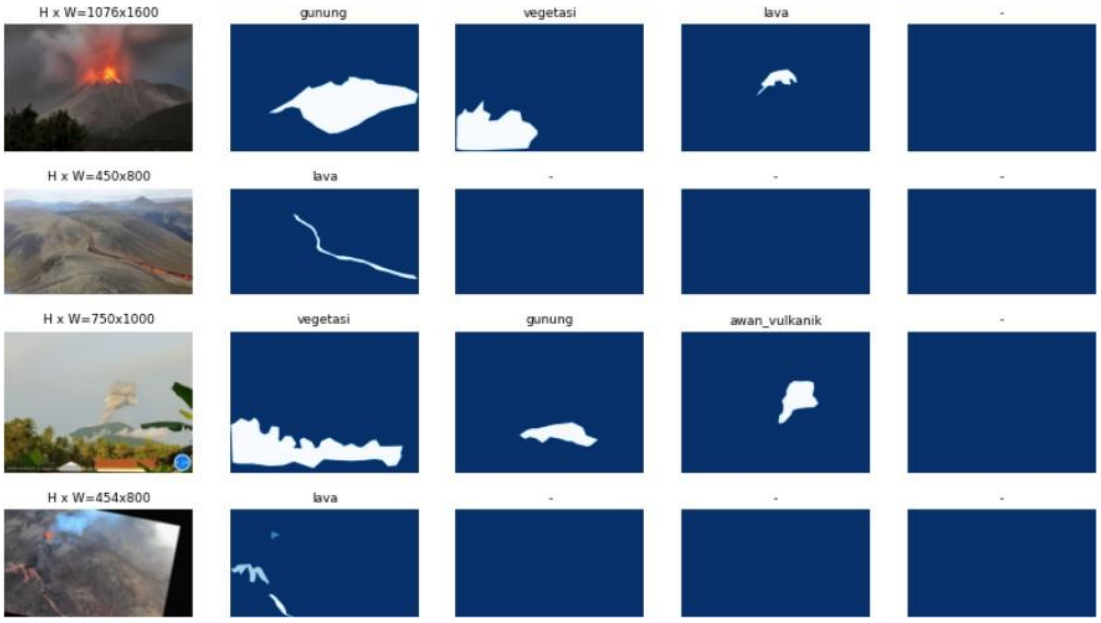
Pada Gambar 4.4 menunjukkan variabel images_path yang menyimpan jalur menuju direktori folder di root drive yang bernama "mask_rcnn", nama folder tersebut sudah di deklarasikan sebagai `PROJECT_NAME` seperti dijelaskan pada Gambar 3.19. File zip berisi data gambar akan diekstrak dengan perintah kode atau diekstrak secara otomatis, ini berbeda dengan YOLO di mana folder dataset harus diekstrak manual sebagaimana penjelasan pada Sub bab 3.2 Anotasi Dataset.

Kemudian dilakukan penyimpanan dataset untuk pelatihan yang disimpan pada variabel `dataset_train`, dataset validasi disimpan pada variabel `dataset_val`, dan penyimpanan informasi jumlah kelas pada `class_number`, di mana ketiga nilai tersebut didapatkan dengan menjalankan fungsi `load_image_dataset( )` dengan menerima `PROJECT_NAME` sebagai parameter. Adapun source code yang digunakan dapat dilihat pada Gambar 4.5.

```
# Load Dataset
dataset_train, dataset_val, class_number = load_image_dataset(PROJECT_NAME)
display_image_samples(dataset_train)
```

Gambar 4.5 Source Code Display Image

Sedangkan baris kode yang menampilkan fungsi `display_image_samples (dataset_train)` digunakan untuk menampilkan sampel citra dan label yang telah di-annotasi pada dataset *training*. Tujuan dari fungsi ini adalah untuk memeriksa apakah proses anotasi citra dan label sudah terbaca dan sesuai dengan ekspektasi. Berdasarkan Gambar 4.6 hasil yang ditampilkan adalah cuplikan dari dataset validasi berupa citra aslinya dan informasi label segmentasinya.



Gambar 4.6 Gambar Data Citra dan Labelnya

Setelah membaca dataset dengan membaca kode program pada Gambar 4.5, selanjutnya adalah memuat konfigurasi default *Mask* R-CNN dan juga memuat model yang akan digunakan pada tahap pelatihan atau training, model yang digunakan merupakan model mask_rcnn_coco.h5 seperti yang telah dijelaskan pada Sub bab 3.5.2 Konfigurasi *Mask* R-CNN. Kode program yang akan dijalankan untuk memuat konfigurasi dan model dapat dilihat pada Gambar 4.7.

```
# Load Model & Configuration
config = CustomConfig(class_number, STEPS_PER_EPOCH, IMAGE_SIZE)
model = load_training_model(config, PROJECT_NAME)
```

Gambar 4.7 Source Code Memuat Konfigurasi dan Model

Pada kode progam di Gambar 4.7, objek `config` akan menyimpan konfigurasi *Mask* R-CNN dengan mewarisi kelas `CustomConfig`, yang menerima argumen `class_number` yang berasal dari Gambar 4.5, serta `STEP_PER_EPOCH` dan `IMAGE_SIZE` yang telah didefinisikan pada Gambar 3.19 Sub bab 3.5.2 Konfigurasi *Mask* R-CNN. Dan model akan dimuat pada variabel `model` dengan memanggil fungsi `load_training_model` yang menerima parameter `config` dan `PROJECT_NAME`, di mana `PROJECT_NAME` telah didefinisikan pada Gambar 3.19 Sub bab 3.5.2 Konfigurasi *Mask* R-CNN.

Setelah menyiapkan dataset, konfigurasi, dan memuat model yang diperlukan, langkah selanjutnya adalah melatih model *Mask* R-CNN. Proses pelatihan dilakukan dalam dua tahap:

pertama melatih kepala jaringan (*head*) dan kemudian melatih seluruh lapisan jaringan. Berikut Gambar 4.8 berisikan baris kode yang digunakan pada proses pelatihan:

```
train_head(model, dataset_train, dataset_val, config, EPOCHS, AUGMENTATION)
model = load_training_model(config, PROJECT_NAME, init_with="last")
train_all_layers(model, dataset_train, dataset_val, config, EPOCHS,
AUGMENTATION)
```

Gambar 4.8 Source Code *Train*ing Citra Model *Mask*-RCNN

Berdasarkan Gambar 4.8, fungsi `train_head( )` bertujuan untuk melatih head layers pada awal pelatihan model, fungsi tersebut memanggil parameter `model`, `dataset_train`, `dataset_val`, dan `config` yang telah dibuat seperti ditmpilkan pada Gambar 4.7 dan Gambar 4.5. Sedangkan `EPOCHS` dan `AUGMENTATION` telah didefinisikan seperti yang dapat dilihat pada Gambar 3.19 Sub bab 3.5.2 Konfigurasi *Mask* R-CNN.

Variabel `model` memanggil fungsi "`load_training_model`" untuk memuat model pelatihan yang digunakan, di mana fungsi akan diinisiasikan dengan parameter `config`, `PROJECT_NAME`, dan "`init_with=last`" yang mengindikasikan pelatihan akan dilanjutkan dari pelatihan terakhir. Baris paling akhir fungsi "`train_all_layer`" dipanggil untuk melatih seluruh lapisan layer dari model.

Berikutnya adalah melakukan testing model epoch ke-70 sebagai model terbaik terhadap dataset uji. Adapun source code dari testing model terbaik dapat dilihat pada Gambar 4.9.

```
selected_model_path = '/content/gdrive/MyDrive/pysource_mrcnn_pro/Mask_rcnn
/trained_models/object20230418T0725/mask_rcnn_object_0070.h5'

def load_test_model_custom(num_classes, project_name, model_path = ''):
    inference_config = InferenceConfig(num_classes)
    model_dir = os.path.join(DRIVE_ROOT_DIR,  project_name, "trained_models
")
    # Recreate the model in inference mode
    model = modellib.MaskRCNN(mode="inference",
                              config=inference_config,
                              model_dir=model_dir)

    # Get path to saved weights
    # Either set a specific path or find last trained weights
    # model_path = os.path.join(ROOT_DIR, ".h5 file name here")
    if model_path == '':
        model_path = model.find_last()

    # Load trained weights
    print("Loading weights from ", model_path)
    model.load_weights(model_path, by_name=True)
```

```
    return model, inference_config


test_model, inference_config = load_test_model_custom(class_number,
PROJECT_NAME, selected_model_path)
```

Gambar 4.9 Source Code Testing Model Terbaik

Pada kode tersebut terdapat sebuah fungsi Python bernama "`load_test_model_custom`" yang berfungsi untuk memuat model *Mask* R-CNN yang sudah dilatih dan di-load dari file dengan ekstensi ".h5". Fungsi ini memiliki tiga parameter yaitu "`num_classes`" yang menentukan jumlah kelas yang ingin dikenali oleh model, "`model_path`" yang menentukan path atau jalur dari file model yang ingin di-load, dan "`project_name`" yang telah didefinisikan sesuai dengan nama folder di drive seperti pada Gambar 3.19 yaitu "mask_rcnn".

Berdasarkan Gambar 4.9 dapat dijelaskan bahwa *source code* tersebut digunakan untuk memuat model terbaik, di mana model terbaik ini adalah model yang memiliki nilai val loss 0.3772 dan train loss 0.3449 yang merupakan nilai terendah, yang didapatkan pada epoch ke 70 dan disimpan pada file berekstensi .h5 bernama "mask_rcnn_object_0070.h5". Untuk hasil proses training yang menampilkan epoch dan nilai loss dan val_loss tiap epochnya dapat dilihat pada Gambar 4.10.

```
Epoch 65/100
65/65 [==============================] - 46s 712ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.4149 - val_loss: 0.6927
Epoch 66/100
65/65 [==============================] - 47s 717ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.4144 - val_loss: 0.7177
Epoch 67/100
65/65 [==============================] - 46s 702ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3771 - val_loss: 0.6787
Epoch 68/100
65/65 [==============================] - 46s 703ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3759 - val_loss: 0.7398
Epoch 69/100
65/65 [==============================] - 47s 723ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.4233 - val_loss: 0.9285
Epoch 70/100
65/65 [==============================] - 46s 708ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3449 - val_loss: 0.3772
Epoch 71/100
65/65 [==============================] - 46s 701ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3502 - val_loss: 0.9602
Epoch 72/100
65/65 [==============================] - 46s 705ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3807 - val_loss: 0.5462
Epoch 73/100
```

```
65/65 [==============================] - 47s 724ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3661 - val_loss: 0.7776
Epoch 74/100
65/65 [==============================] - 46s 700ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.3765 - val_loss: 0.8920
Epoch 75/100
65/65 [==============================] - 46s 706ms/step - batch: 32.0000 -
size: 2.0000 - loss: 0.4306 - val_loss: 0.6736
```

Gambar 4.10 Hasil Riwayat Training

Setelah melakukan *load* terhadap model, tahap berikutnya adalah melakukan evaluasi dengan menggunakan Mean Average Precision (mAP), evaluasi ini digunakan untuk melakukan evaluasi kinerja model secara keseluruhan, yang merupakan nilai rata - rata dari Average Precision (AP) keseluruhan kelas. Adapun *source code* yang digunakan untuk melakukan evaluasi Mean Average Precision (mAP) dapat dilihat pada Gambar 4.10.

```
# The evaluation of the Model is performed on the Validation images
# mAP @ IoU=0.5
APs = model_evaluation(dataset_val, test_model, inference_config)
print("mAP: ", np.mean(APs))
```

Gambar 4.11 Source Code E*val*uasi Mean Average Precision (mAP)

Pada Gambar 4.11, Kode tersebut mengevaluasi performa model pada dataset validasi dengan mAP (mean Average Precision). Pertama, fungsi `model_evaluation()` digunakan untuk menyimpan nilai precision dari setiap data validasi dan menghitung APs untuk setiap kelas pada *threshold* IoU=0.5. Terakhir, nilai rata-rata dari semua APs dihitung dengan np.mean() dan dihasilkan mAP sebagai ukuran performa dari model. Untuk detail fungsi model_evaluation dapat dilihat pada Gambar 4.12.

```
def model_evaluation(dataset_val, test_model, inference_config):
    APs = []
    print("Testing the model on {} validation
images.".format(len(dataset_val.image_ids)))
    for image_id in dataset_val.image_ids:
        # Load image and ground truth data
        image, image_meta, gt_class_id, gt_bbox, gt_mask = \
            modellib.load_image_gt(dataset_val, inference_config,
                                   image_id, use_mini_mask=False)
        molded_images = np.expand_dims(modellib.mold_image(image,
inference_config), 0)
        # Run object detection
        results = test_model.detect([image], verbose=0)
        r = results[0]
        # Compute AP
        AP, precisions, recalls, overlaps = \
            utils.compute_ap(gt_bbox, gt_class_id, gt_mask,
                             r["rois"], r["class_ids"], r["scores"],
r['masks'])
```

```
        APs.append(AP)
    return APs
```

Gambar 4.12 Fungsi model_evaluation

Pada Gambar 4.12 menunjukkan bahwa fungsi `model_evaluation` menerima tiga parameter: dataset_val, test_model, dan inference_config. Secara umum fungsi tersebut menjalankan tahapan berikut:

1. Membuat sebuah list kosong bernama APs untuk menyimpan Average Precision (AP) dari setiap gambar dalam dataset validasi.
2. Mencetak pesan yang menunjukkan jumlah gambar validasi yang akan diuji.
3. Melakukan iterasi pada setiap `image_id` dalam `dataset_val.image.ids`. Ini berarti fungsi akan mengevaluasi setiap gambar dalam dataset validasi secara individual.
4. Pada setiap iterasi:
    - Memuat gambar dan data ground truth (gt) terkait.
    - Melakukan resize gambar dengan menggunakan fungsi mold_image dari modul modellib.
    - Menjalankan deteksi objek pada gambar dengan menggunakan model yang diuji.
    - Mengambil hasil deteksi pada gambar yang pertama (indeks 0).
    - Mendapatkan nilai Average Precision (AP), presisi (precisions), recall (recalls), dan perpotongan area (overlaps) dengan memanggil fungsi utils.compute_ap( ).
    - Menyimpan nilai AP pada list APs.
5. Mengembalikan list APs yang berisi AP dari setiap gambar dalam dataset validasi.

Sedangkan untuk penentuan nilai IoU sebesar 0.5 mengikuti baris kode default yang terdapat pada file model.py seperti yang ada pada Gambar 4.13.

```
# Positive ROIs are those with >= 0.5 IoU with a GT box.
    fg_ids = np.where(rpn_roi_iou_max > 0.5)[0]
```

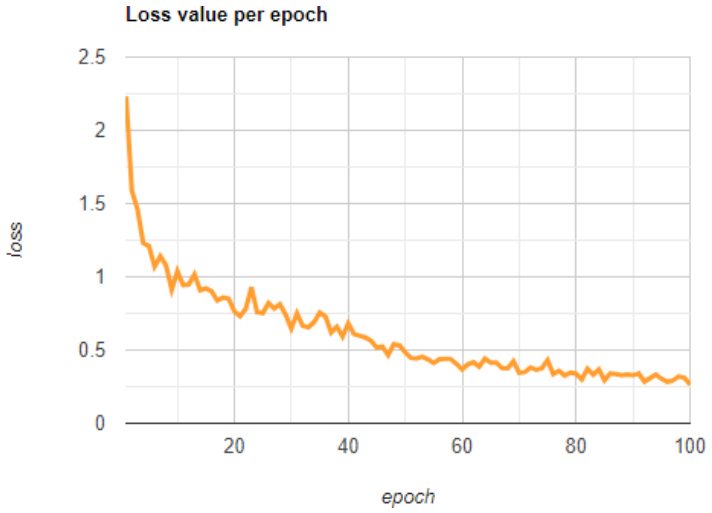
Gambar 4.13 Pendefinisian Nilai IoU

Pada kutipan kode Gambar 4.13 mendefinisikan bahwa nilai ROI (Region of Interest) hanya bisa dipenuhi kondisinya jika nilai IoU di atas 0.5 dan hasilnya akan disimpan dalam "`fg_ids`".

```
Testing the model on 15 validation images.
mAP:  0.5351851867304909
```

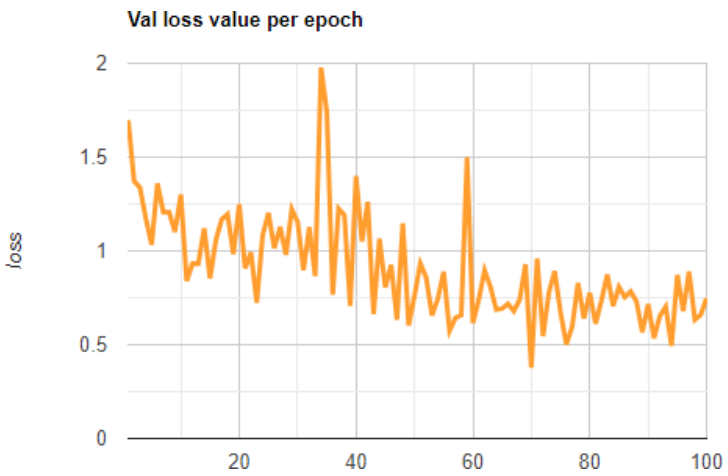
Gambar 4.14 Hasil Evaluasi Mean Average Precision (mAP)

Hasil menjalankan kode pada Gambar 4.11 dapat dilihat pada Gambar 4.14. Dapat disimpulkan bahwa hasil evaluasi terhadap model *Mask* R-CNN dengan menggunakan Mean Average Precision (mAP) dengan nilai *threshold* IoU sebesar 0.5 diperoleh hasil sebesar 0.535. Nilai ini menunjukkan bahwa model memiliki ruang untuk peningkatan performa, namun memiliki sudah cukup baik dalam melakukan segmentasi objek pada dataset. Nilai tersebut berarti model terbaik yang dipilih memiliki tingkat akurasi sebesar 53.5% dalam mengelompokkan objek ke kelas yang benar.



Gambar 4.15 Hasil Loss Value Model

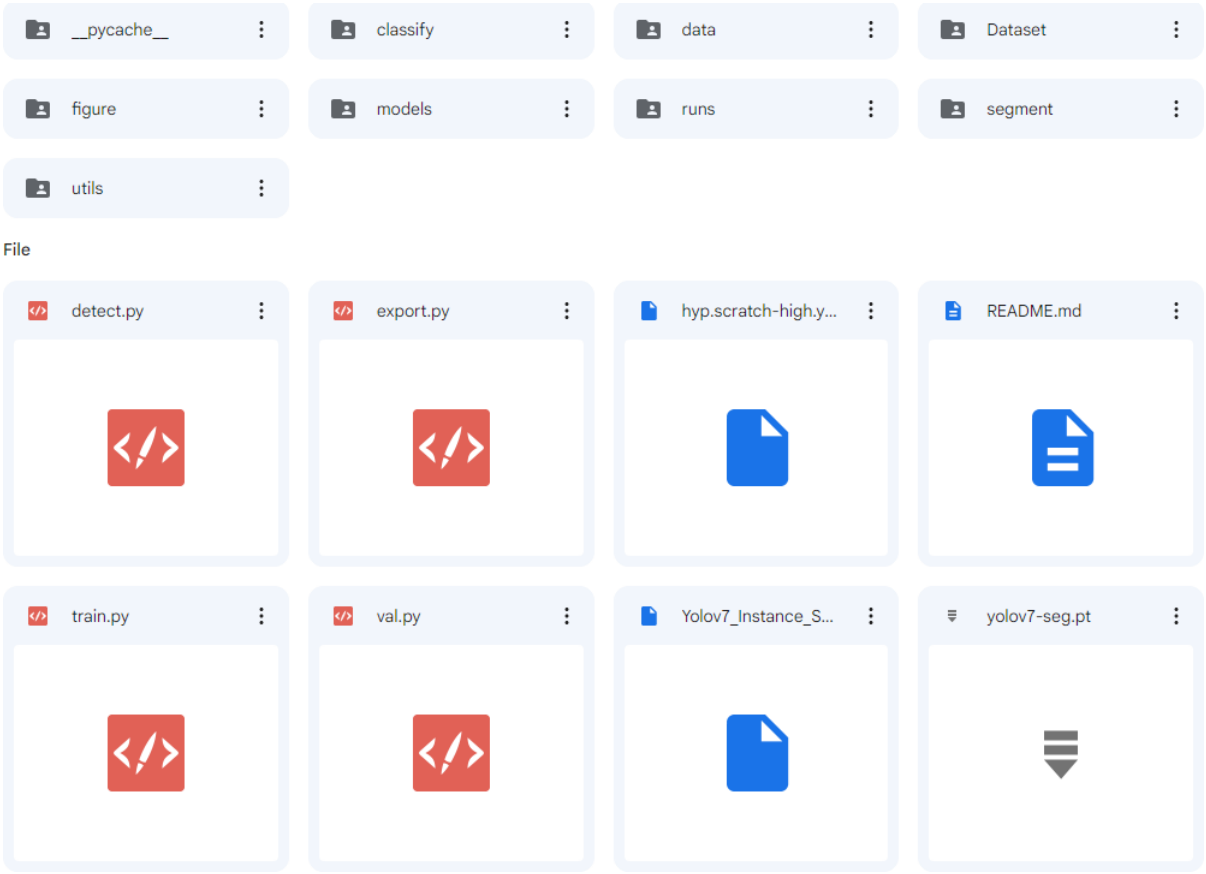
Berdasarkan Gambar 4.15 maka dapat dilihat bahwa model *Mask*-RCNN mengalami penurunan nilai loss pada epoch ke-100 dengan nilai 0,265, tetapi berdasarkan Gambar 4.16 nilai val loss terkecil ada pada epoch ke-70 dengan nilai 0,372 sedangkan paada epoch ke-100 nilai val loss sebesar 0,747. Sehingga model terbaik diambil pada epoch ke-70. Untuk detail grafik dari loss *validation* dapat dilihat pada Gambar 4.16.



Gambar 4.16 Hasil Loss Validation

4.3 Implementasi Model YOLOv7

Selanjutnya adalah menggunakan model YOLOv7. Pada tahap implementasi model YOLOv7, hal yang pertama dilakukan adalah dengan memuat model YOLOv7 dari library atau file yang tersedia dan menyiapkan data *training* dan *validasi* yang telah dianotasi seperti dijelaskan pada Sub bab 3.2.1. Sebelum melakukan pelatihan model, data-data yang diperlukan harus disiapkan ke Google Drive projek YOLOv7. Pada Google Drive sendiri perlu dibuat kumpulan folder yang akan menyimpan kebutuhan kode program YOLOv7 di Google Colab. Di mana struktur folder yang telah dibuat pada Google Drive mengikuti referensi dari https://github.com/WongKinYiu/YOLOv7 seperti yang ditunjukkan pada Gambar 4.17.



Gambar 4.17 Struktur Folder YOLOv7

Pada struktur folder tersebut akan terdapat beberapa dokumen yang digunakan untuk menjalankan program YOLOv7, seperti train.py dan yolov7-seg.pt yang akan digunakan ketika melakukan train pada YOLOv7, file hyperparameter, juga dataset untuk train dan validasi seperti pada Gambar 3.5 Sub bab 3.2.1 akan disimpan di dalam struktur folder tersebut. Tahap berikutnya adalah membuat program YOLOv7 membaca direktori ke file segment. Source code untuk melakukan *mount* terhadap folder Drive YOLOv7 dapat dilihat pada Gambar 4.18.

```
cd /content/drive/MyDrive/YOLOv7/segment
```

Gambar 4.18 Source Code Change Directory

Kemudian dilakukan *download hyperparameter* YOLOv7 yang ditunjukkan pada Gambar 4.18. Kode pada Gambar 4.19 mengunduh file hyp.scratch-high.yaml dari GitHub WongKinYiu. File ini memuat konfigurasi *hyperparameter* untuk YOLOv7 yang digunakan pada penelitian ini.

```
!wget https://raw.githubusercontent.com/WongKinYiu/YOLOv7/u7/seg/dat
a/hyps/hyp.scratch-high.yaml
```

Gambar 4.19 Source Code Download Hyperparameter YOLOv7

Jika pada Sub bab 3.5.1 terdapat konfigurasi mengenai batch, step per epoch, dan image size, hyperparameter yang didownload pada Gambar 4.18 berisikan kumpulan parameter default yang ditentukan berdasarkan sumber GitHub YOLOv7 asli. Untuk detail isi dari hyperparameter yang didownload dapat dilihat pada Gambar 4.20.

```
lr0: 0.01 # initial learning rate (SGD=1E-2, Adam=1E-3)
lrf: 0.1 # final OneCycleLR learning rate (lr0 * lrf)
momentum: 0.937  # SGD momentum/Adam beta1
weight_decay: 0.0005 # optimizer weight decay 5e-4
warmup_epochs: 3.0 # warmup epochs (fractions ok)
warmup_momentum: 0.8 # warmup initial momentum
warmup_bias_lr: 0.1  # warmup initial bias lr
box: 0.05 # box loss gain
cls: 0.3  # cls loss gain
cls_pw: 1.0  # cls BCELoss positive_weight
obj: 0.7 # obj loss gain (scale with pixels)
obj_pw: 1.0 # obj BCELoss positive_weight
iou_t: 0.20  # IoU training threshold
anchor_t: 4.0  # anchor-multiple threshold
# anchors: 3  # anchors per output layer (0 to ignore)
fl_gamma: 0.0 # focal loss gamma (efficientDet default gamma=1.5)
hsv_h: 0.015  # image HSV-Hue augmentation (fraction)
hsv_s: 0.7 # image HSV-Saturation augmentation (fraction)
hsv_v: 0.4 # image HSV-Value augmentation (fraction)
degrees: 0.0  # image rotation (+/- deg)
translate: 0.1 # image translation (+/- fraction)
scale: 0.9  # image scale (+/- gain)
shear: 0.0  # image shear (+/- deg)
perspective: 0.0 # image perspective (+/- fraction), range 0-0.001
flipud: 0.0  # image flip up-down (probability)
fliplr: 0.5 # image flip left-right (probability)
mosaic: 1.0  # image mosaic (probability)
mixup: 0.1  # image mixup (probability)
copy_paste: 0.1  # segment copy-paste (probability)
```

Gambar 4.20 Isi dari Hyperparameter Default YOLOv7

Selanjutnya melakukan instalasi *library* PyYAML. Pada penelitian ini library PyYAML digunakan untuk membaca file berekstensi ".yaml" yang digunakan oleh *source code* asli dari model YOLOv7. PyYAML adalah library Python untuk membaca dan menulis data dalam format YAML. Format YAML umumnya digunakan untuk menyimpan data konfigurasi seperti hasil anotasi dari Roboflow pada Sub bab 3.2.1 atau file hyperparameter seperti ditunjukkan Gambar 4.19. File YAML juga digunakan untuk menyimpan konfigurasi

aplikasi, skema database, atau data yang terstruktur lainnya. Adapun *source code* untuk melakukan instalasi library PyYAML dapat dilihat pada Gambar 4.20.

```
!pip install PyYAML==5.4.1
```

Gambar 4.21 Source Code Instalasi Library PyYAML

Berdasarkan Gambar 4.21 ketika penelitian ini dilakukan, versi PyYAML yang digunakan adalah versi 5.4.1 yang merupakan versi stabil saat itu.

Langkah selanjutnya menjalankan tahapan training pada model. Adapun *source code* untuk melakukan menjalankan training dan konfigurasi dari setting Sub bab 3.5.1 dapat dilihat pada Gambar 4.22.

```
!python train.py --data ../Dataset/lava/data.yaml --batch 16 --
weights ../YOLOv7-seg.pt --epochs 100 --name ../YOLOv7-seg --
img 640 --hyp ../hyp.scratch-high.yaml
```

Gambar 4.22 Source Code Pengaturan Model YOLOV7

Gambar 4.22 dapat dijelaskan bahwa potongan *source code* "`!python train.py`" menunjukkan bahwa pada tahap ini kode program yang akan dijalankan berasal dari file train.py yang terdapat pada lokasi Google Drive seperti ditunjukkan Gambar 4.16. Bagian "`--data ../Dataset/lava/data.yaml`" digunakan untuk melakukan pemilihan lokasi folder dataset dan lokasi file YAML yang telah di upload pada Google Drive, bagian kode "`batch 16`" digunakan untuk menentukan ukuran batch, bagian kode "`weights ../YOLOv7-seg.pt`" digunakan untuk melakukan *load weight pretrained* YOLOv7 yang dijelaskan pada Gambar 3.18 Pre-Trained Weight YOLOv7 di Sub bab 3.5.1, bagian kode "`epochs 100`" digunakan untuk menentukan jumlah epochs yang akan digunakan, bagian kode "`name ../YOLOv7-seg`" untuk memberi nama folder yang akan menjadi tempat penyimpanan hasil *training*, bagian kode "`img 640`" digunakan untuk melakukan pengaturan *resize* citra dan yang terakhir "`hyp ../hyp.scratch-high.yaml`" digunakan untuk melakukan load *hyperparamter* yang telah di download pada Gambar 4.19. Adapun hasil dari source code pada Gambar 4.22 akan menghasilkan riwayat hasil pelatihan dan dapat dilihat pada Gambar 4.23.

Class	Images	Instances
all	30	56
awan_vulkanik	30	11
gunung	30	9
lava	30	34
vegetasi	30	2

(a)

Box(P	R	mAP50	mAP50-95)
0.834	0.818	0.847	0.607
0.861	0.909	0.895	0.718
0.652	0.629	0.69	0.493
0.833	0.735	0.807	0.569
0.988	1	0.995	0.647

(b)

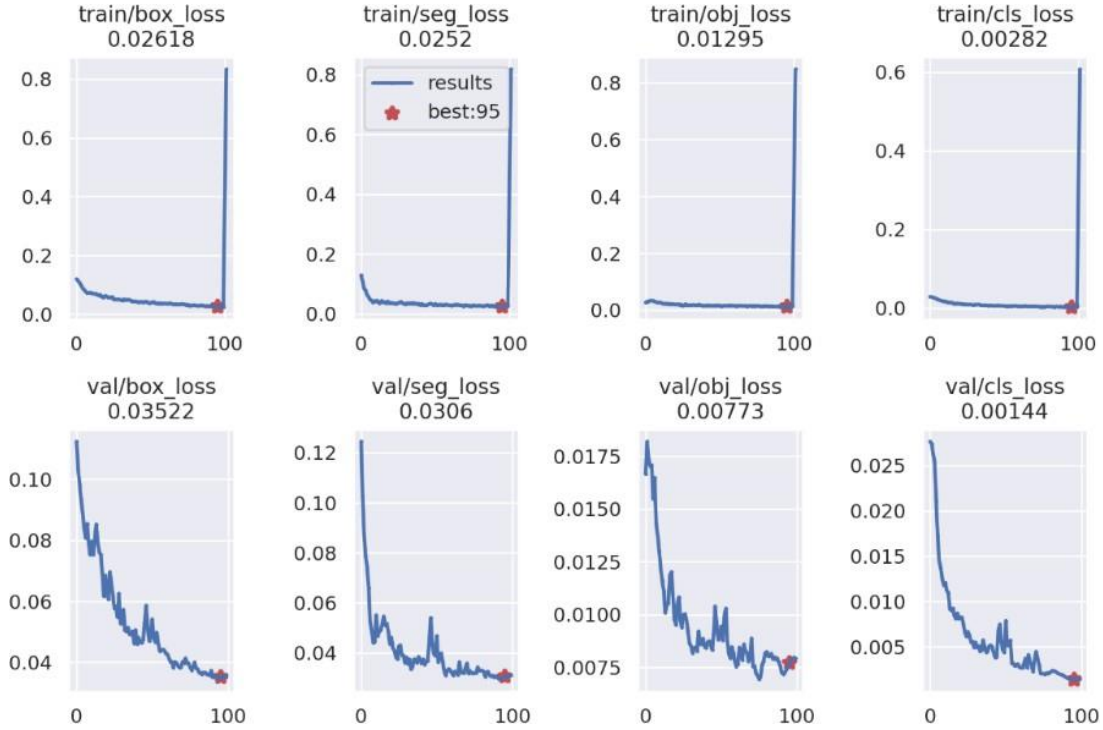
Mask(P	R	mAP50	mAP50-95)
0.85	0.842	0.872	0.552
0.946	1	0.995	0.637
0.653	0.632	0.69	0.457
0.833	0.735	0.809	0.368
0.968	1	0.995	0.746

(c)

Gambar 4.23 Hasil Training YOLOv7

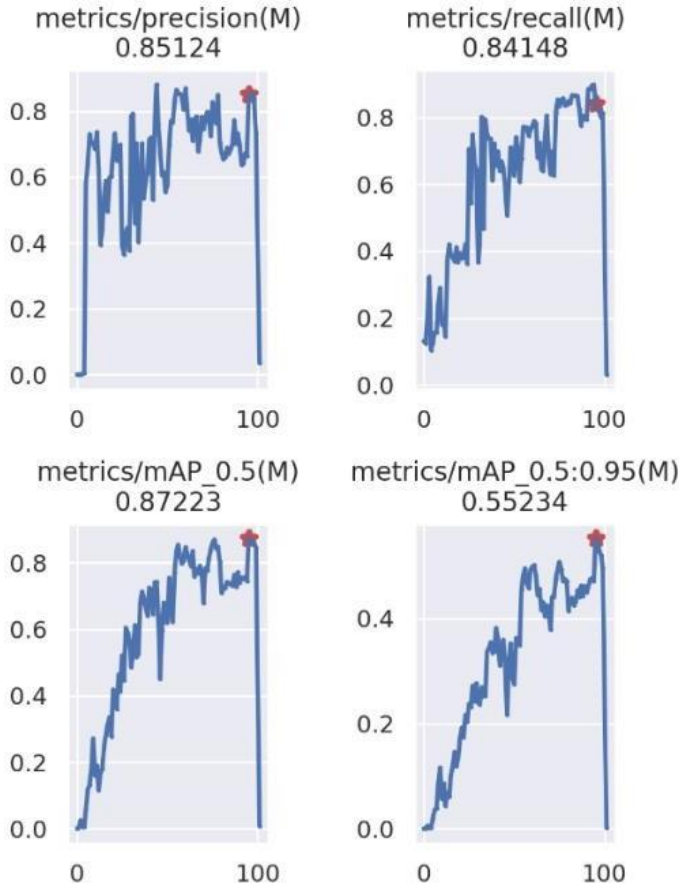
Berdasarkan Gambar 4.23 terdapat analisis hasil evaluasi nilai presisi (P), recall (R), mAP pada IoU 0,5 (mAP50) dan mAP pada IoU 0,5 - 0,95 (mAP50-95), dari pelatihan pada data validasi. Pada Gambar 4.23 (a), “Class” merupakan daftar kelas objek yang yang sudah ditentukan, “Images” menampilkan jumlah data validasi, “Instances” menunjukkan jumlah berapa kali objek dari kelas tertentu ditemukan. Untuk Gambar 4.23 (b) merupakan ringkasan dari kualitas bounding box pada hasil training validasi. Untuk analisis kualitas mask atau segmentasi dapat dilihat pada Gambar 4.23 (c), di mana untuk mAP50 Mask pada seluruh kelas memperoleh nilai sebesar 0,872.

Adapun hasil tampilan grafik hasil proses training pada model YOLOv7 dapat dilihat pada Gambar 4.24.



Gambar 4.24 Hasil Evaluasi Loss Training Model YOLOv7

Berdasarkan Gambar 4.24 dapat dilihat bahwa model YOLOv7 mengalami penurunan nilai *validation loss*, seperti pada gambar tersebut, val/box_loss memiliki nilai 0.03522, val/seg_loss memiliki nilai 0.0306, val/obj_loss memiliki nilai 0.00773, val/cls_loss memiliki nilai 0.00114. Kemudian terdapat juga nilai loss train di kategori lain seperti train/box_loss, train/seg_loss, train/obj_loss, train/cls_loss. Seluruh nilai evaluasi baik di tahap train dan validation terus turun di mana hal tersebut menunjukkan model melakukan proses pembelajaran dan mengurangi nilai loss.



Gambar 4.25 Hasil Evaluasi mAP Validasi Model YOLOv7

Sedangkan pada Gambar 4.25 merupakan perhitungan precission, recall, dan mAP dari proses validasi. Model YOLOv7 menghasilkan mean Average Precision (mAP) dengan nilai 0.87223 untuk *threshold* IoU 0.5, dan nilai 0.55234 untuk *threshold* IoU 0.5 hingga 0.95. Hasil evaluasi menunjukkan bahwa model YOLOv7 memiliki performa yang baik dalam pengenalan objek selama pelatihan pada *threshold* 0.5 IoU, namun menurun pada *threshold* 0.5 hingga 0.95 IoU.

4.4 Penyandingan Kedua Model

Dengan karakteristik dari kedua model tersebut dapat dipahami bahwa keduanya memiliki kelebihan dan kekurangan masing-masing. Penelitian ini memiliki tujuan untuk mengajukan alternatif pengawasan gunung berapi dengan model *deep learning* dan tidak terpaku dari perbandingan secara detail arsitektur dari kedua model. Kemampuan kedua model inheren terbatas dari desain awal model dan tujuan utama kedua model tersebut dikembangkan oleh peneliti aslinya. YOLO menekankan kecepatan dan model yang ringan,

sedangkan *Mask* R-CNN menekankan tingkat pengenalan objek yang lebih tinggi seperti nilai confident level pada deteksi objek, namun lebih lambat. Penggunaan *real-time* dari pengajuan alternatif pengawasan gunung pada penelitian ini tentu akan menyesuakan dengan kondisi sumber daya, seperti kemampuan komputasi. Maka dari itu tujuan utama penelitian ini untuk melihat terlebih dahulu potensi dari model yang tersedia, maka dilakukan penyandingan dari kedua model yang dipilih, yang memiliki potensi kemampuan deteksi dan segmentasi.

4.5 Hasil Evaluasi Validasi

Berdasarkan hasil pengujian dengan menggunakan mean Average Precision (mAP) telah dilakukan terhadap model *Mask* R-CNN dan YOLOv7 dapat dilihat pada Tabel 4.1.

Tabel 4.2 Hasil mAP *Mask* R-CNN dan YOLOv7

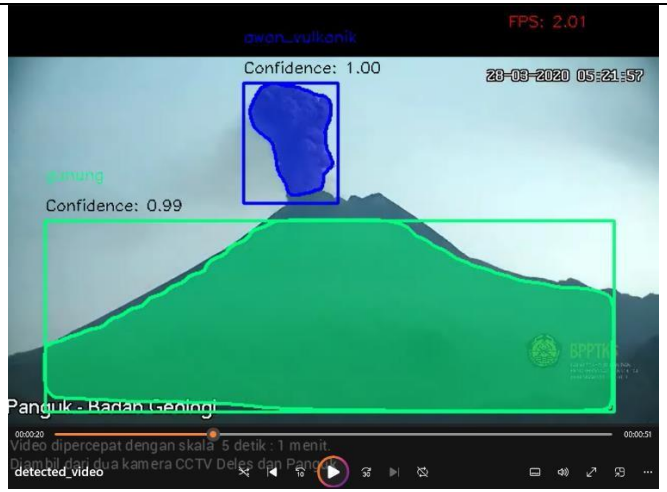
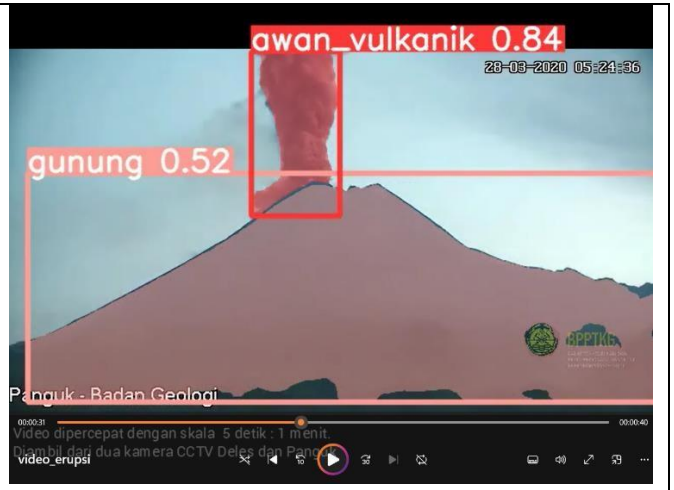
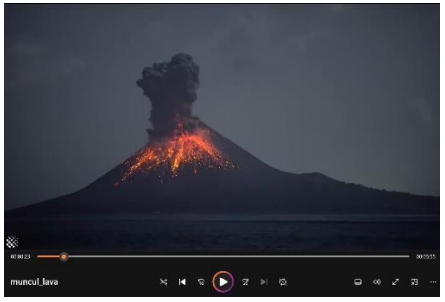
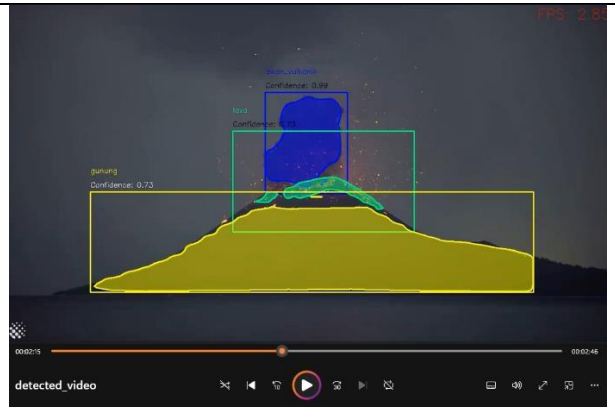
***Mask*-RCNN**	**YOLOV7**
0.5351851867304909	0,87223

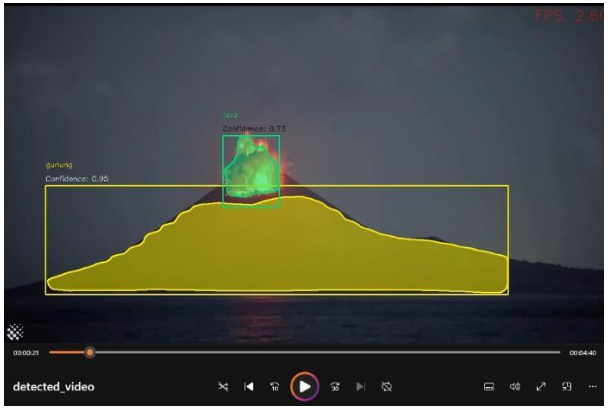
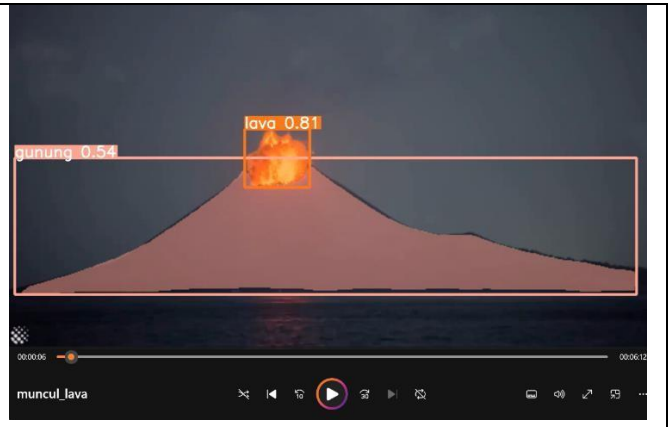
Dari Tabel 4.2 diatas, didapatkan hasil mAP pada *Mask* R-CNN sebesar 0.5351851867304909, sedangkan hasil mAP pada YOLOv7 adalah sebesar 0,872. Dengan menggunakan dua sumber pengujian, yaitu data citra dan video. Maka hasil deteksi dan segmentasi pada data uji dari kedua model dapat dilihat pada Tabel 4.3.

4.6 Hasil Pengujian

Tabel 4.3 Hasil Perbandingan Data Uji Video

No.	Cuplikan Frame Data Video Asli	Hasil Deteksi Model	
		***MASK* R-CNN**	**YOLOv7**
1.			

2.	Panguk - Badan Geologi erupsi_gunung	awan_vulkanik FPS: 2.01 Confidence: 1.00 28-03-2020 05:21:57 gunung Confidence: 0.99 Panguk - Badan Geologi Video dipercepat dengan skala 5 detik : 1 menit. detected_video	awan_vulkanik 0.84 28-03-2020 05:24:36 gunung 0.52 Panguk - Badan Geologi Video dipercepat dengan skala 5 detik : 1 menit. video_erupsi
3.	muncul_lava	gunung Confidence: 0.73 detected_video	awan_vulkanik 0.89 gunung 0.52 lava 0.50 muncul_lava

4.	muncul_lava	gunung Confidence: 0.95 lava Confidence: 0.73 detected_video	lava 0.81 gunung 0.54 muncul_lava

Berdasarkan **Error! Reference source not found.**2 terdapat dua sumber video yang merupakan bagian dari data *test* yang telah dilakukan inferensi dengan menggunakan model *Mask* R-CNN dan YOLOv7. Dari hasil deteksi tersebut menunjukan bahwa hasil segmentasi yang dilakukan oleh model *Mask* R-CNN dan YOLOv7 memiliki kualitas yang beragam dan menghasilkan nilai *confident* yang baik.

YOLOv7 dan Mask R-CNN keduanya berhasil mendeteksi dan mensegmentasi objek dalam citra uji yang diberikan. Namun, terdapat beberapa perbedaan yang cukup mencolok antara kedua model tersebut. YOLOv7, meskipun memiliki nilai mAP hasil pelatihan validasi yang lebih tinggi (0.872) dibandingkan dengan Mask R-CNN (0.535). Untuk YOLOv7 menunjukkan variasi nilai *confidence score* yang lebih besar dalam rentang 0.50 hingga 0.90. Sementara itu, Mask R-CNN selalu menghasilkan nilai confidence score yang sangat tinggi, di atas 0.90, untuk semua objek yang terdeteksi.

Selain itu, terdapat beberapa kasus di mana kualitas segmentasi dari Mask R-CNN lebih buruk dibandingkan dengan YOLOv7, terutama dalam segmentasi area gunung, seperti pada Tabel 4.3 baris nomor ke-3 dan ke-4 di mana *Mask* R-CNN bisa mendeteksi dan melakukan segmentasi area gunung, tetapi batas gunung tidak serapih hasil YOLOv7. Sementara hasil segmentasi YOLOv7 tidak dapat mendeteksi dan melakukan segmentasi pada area kawah gunung seperti *Mask* R-CNN yang ditunjukkan pada gambar baris ke-1 pada Tabel 4.3.

Perbedaan antara nilai mAP dan tingkat *confident level* pada Mask R-CNN dan YOLOv7 dapat dijelaskan oleh perbedaan dalam arsitektur dan metode pelatihan kedua model. Mask R-CNN adalah model berbasis dua tahap yang menggunakan Region Proposal Network (RPN) untuk menghasilkan proposal objek sebelum melakukan klasifikasi dan segmentasi, sedangkan YOLOv7 adalah model berbasis satu tahap yang melakukan deteksi dan klasifikasi secara bersamaan. Hal ini dapat menyebabkan Mask R-CNN lebih percaya diri dalam hasil deteksinya, karena telah mempertimbangkan konteks objek sebelum melakukan klasifikasi.

Namun, kualitas segmentasi yang lebih buruk pada Mask R-CNN pada beberapa citra mungkin disebabkan oleh kesulitan dalam menggeneralisasi batas-batas objek yang kompleks, seperti vegetasi. Model YOLOv7, dengan pendekatan yang lebih sederhana dan lebih cepat, mungkin lebih fleksibel dalam menangani variasi ini, sehingga menghasilkan kualitas segmentasi yang lebih baik dalam beberapa kasus.

Meskipun YOLOv7 memiliki nilai mAP yang lebih tinggi, Mask R-CNN mungkin lebih sesuai jika tingkat kepercayaan yang tinggi dalam deteksi objek merupakan prioritas utama. Namun, jika kualitas segmentasi dan kecepatan inferensi lebih penting, YOLOv7 mungkin menjadi pilihan yang lebih baik.

4.7 Analisis Kekurangan dan Kelebihan Model

Dari analisis hasil pengujian kedua model menunjukkan keduanya memiliki hasil yang berbeda baik dari hasil proses pelatihan maupun prediksi pada area *Mask* ataupun deteksi. Perbedaan ini bisa dianalisis lebih lanjut mengenai kelebihan dan kekurangan yang dimiliki dari kedua model yang digunakan untuk melakukan segmentasi citra gunung berapi.

YOLOv7 dan *Mask* R-CNN keduanya merupakan model yang bisa digunakan untuk melakukan segmentasi dan deteksi objek pada sebuah citra. Kedua model memiliki pendekatan berbeda mengenai bagaimana proses segmentasi dilakukan. YOLOv7 membagi cita menjadi beberapa *grid* atau bagian, lalu dalam setiap *grid* dilakukan prediksi untuk menghasilkan segmentasi dan deteksi objek. Pada *Mask* R-CNN untuk mendapatkan segmentasi adalah dengan membuat proposal area yang akan menjadi fokus deteksi dan segmentasi pada *layer* Region Proposal Network (RPN) baru kemudian masuk ke layer CNN untuk memprediksi area segmentasi.

Berdasarkan desain arsitektur, hasil proses pelatihan model, dan analisis hasil pengujian model pada data gambar dan video, maka analisis kelebihan dan kekurangan kedua model dapat disimpulkan pada Tabel 4.3 berikut.

Tabel 4.3 Analisis Kelebihan dan Kekurangan Model

Fitur yang Dianalisis	**YOLOv7**	***Mask* R-CNN**
Kecepatan Pelatihan	Cepat (30 Menit)	Lambat (2 Jam)
Nilai Confident Level	Lebih Rendah (0.63 – 0.90)	Tinggi (Di atas 0.90)
Kualitas Segmentasi	Lebih baik untuk semua kelas objek	Cukup baik, dengan kualitas kurang pada objek kompleks

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka diperoleh kesimpulan sebagai berikut:

a. Penelitian ini berhasil menerapkan metode *instance segmentation* menggunakan model *Mask* Region-Based Convolutional Neural Network (*Mask* R-CNN) dan You Only Look Once Version7 (YOLOv7) dalam melakukan segmentasi objek yang ada di citra gunung berapi dengan memanfaatkan dataset yang telah dianotasi.
b. Hasil segmentasi citra lava dengan menggunakan model *Mask* R-CNN dan YOLOv7 dapat berjalan dengan baik, di mana pelatihan kedua model yang dilakukan dengan menggunakan data latih yang sama memperoleh hasil mAP sebesar 53% untuk *Mask*-RCNN dan 87% untuk YOLOv7. Dan hasil uji pada lima data uji (*test*) yang sama menunjukkan variasi hasil *confidence level* di atas 50% dan mampu melakukan segmentasi pada objek-objek yang teridentifikasi sebagai kelas objek yang menjadi fokus penelitian ini.

5.2 Saran

Adapun saran untuk pengembangan penelitian kedepannya adalah sebagai berikut:

a. Diperlukan penambahan jumlah dataset yang jauh lebih besar untuk memperbaiki hasil akurasi yang diperoleh dari kedua model.
b. Perlu diimplementasikan kedalam sebuah aplikasi berbasis web atau bahkan mobile guna mempermudah pengguna model secara praktis oleh ahli dalam melakukan pengamatan gunung berapi.

DAFTAR PUSTAKA

Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. *International Conference on Advanced Computing & Communication System*, 687-694.

Anantharaman, R., Velazquez, M., & Lee, Y. (2018). Utilizing mask R-CNN for detection and segmentation of oral diseases. *IEEE international conference on bioinformatics and biomedicine (BIBM)*, 2197-2204.

Asni, A., Amin, & Waruni, M. (2021). Penerapan Metode Yolo Object Detection V1 Terhadap Proses Pendeteksian Jenis Kendaraan Di Parkiran. *JTE Uniba, 6(1)*, 194-199.

Baihaqi, K. A., & Zonyfar, C. (2022). Deteksi Lahan Pertanian Yang Terdampak Hama Tikus Menggunakan Yolo v5: Indonesia. *Jurnal Informatika, 11(2)*, 1-9.

Benyang, D., Xiaochun, L., & Miao, Y. (2020). Safety helmet detection method based on YOLO v4. *International Conference on Computational Intelligence and Security*, 155-158.

Deci. (2022, November 3). *How Does YOLOv6 Differ From YOLO v5*. Retrieved from Deci.ai: https://deci.ai/blog/yolov6-vs-yolov5-vs-yolox/

Fadillah, F. N., & Aelani, K. (2022). Sistem Deteksi Penggunaan Masker pada Pengunjung STMIK "AMIKBANDUNG" Menggunakan Algoritma Convolutional Neural Network (CNN). *Jurnal ICT: Information Communication & Technology*, 172-183.

Ghosh, S., Das, N., Das, I., & Maulik, U. (2019). Understanding deep learning techniques for image segmentation. *ACM Computing Surveys (CSUR, 52(4)*, 1-35.

Griffiths, R. W. (2000). The dynamics of lava flows. *Annual review of fluid mechanics, 32(1)*, 477-518.

Hendrik, Anjomshooa, A., & Tjoa, A. M. (2014). Towards Semantic Mashup Tools For Big Data Analysis. *Proceding of the Information & Communication Technology-EurAsia Conference 2014*, (pp. 100-145). Bali.

Hermawan, M. I., Tritoasmoro, I. I., & Ibrahim, N. (2021). Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo. *e-Proceeding of Engineering, 8(1)*.

Jenkins, S. F., Day, S. J., Faria, B. V., & Fonseca, J. F. (2017). Damage from lava flows: insights from the 2014-2015 eruption of Fogo, Cape Verde. *Journal of applied Volcanology, 6(2)*.

Kusuma, I. W., & Ellyana, R. L. (2018). Penerapan Citra Terkompresi Pada Segmentasi Citra Menggunakan Algoritma K-Means. *JUTEI, 2(1)*, 65-74.

Lawal, O. M. (2023). YOLOv5-LiNet: A lightweight network for fruits instance segmentation. *PLOS ONE*, *18*(3), e0282297. https://doi.org/10.1371/JOURNAL.PONE.0282297

Liu, K., Sun, Q., Sun, D., Peng, L., Yang, M., & Wang, N. (2023). Underwater Target Detection Based on Improved YOLOv7. *Journal of Marine Science and Engineering*, 1-21.

Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., & Zhang, Y. (2021). Abdomenct-1k: Is abdominal organ segmentation a solved problem? *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10)*, 6695-6714.

MAGMA Indonesia (Multiplatform Application for Geohazard Mitigation and Assessment in Indonesia). (n.d.). Retrieved November 23, 2022, from https://magma.esdm.go.id/v1/gunung-api/gallery

Mailoa, R. M., & Santoso, W. L. (2022). Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN. *Jurnal Infra, 10(2)*, 56-62.

Miao, Y., Liu, F., Hou, T., Liu, L., & Liu, Y. (2020). A nighttime vehicle detection method based on YOLO v3. *Chinese Automation Congress (CAC)*, 6617-6621.

Negro, C. D., Cappello, A., & Bilotta, G. (2020). Living at the edge of an active volcano: Risk from lava flows on Mt. Etna. *Geological Society of America Bulletin*, 132.

Pitriyani, E., & Ardiansyah. (2019). Rekomendasi Pembimbing dan Penguji Tugas Akhir Menggunakan Vector Space Model. *Prosiding Nasional Rekayasa Teknologi Industri dan Informasi*, 317-323.

Qiao, Y., Truman, M., & Sukkarieh, S. (2019). Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Computers and Electronics in Agriculture*, 165.

Rahman, A. A., Agustin, S. D., Ibrahim, N., & Kumalasari, C. N. (2022). Perbandingan Algoritma YOLOv4 dan Scaled YOLOv4 untuk Deteksi Objek pada Citra Termal. *Multimedia Artificial Intelligent Networking Database, 7(1)*, 61-71.

Setiawan, A. M. (2013). *Integrated Framework For Business Process Complexity Analysis*. Retrieved from ECIS 2013 Completed Research: http://aisel.aisnet.org/ecis2013_cr/49

Sholahuddin, M. R., Atqiya, F., Wulan, S. R., Harika, M., Fitriani, S., & Sofyan, Y. (2023). Implementasi Sistem Identifikasi Senjata Rea Time Menggunakan YOLOv7 dan Notifikasi Chat Telegram. *Journal of Information System Research (JOSH)*, 598-606.

Sudarsono, B., Yuwono, B. D., & Ramadhan, F. (2019). Analisis Sebaran Aliran Lava Untuk Pembuatan Peta Mitigasi Bencana Gunung Slamet. *Jurnal Geodesi dan Geomatika, 2(1)*, 13-20.

Sun, Q., Magee, & Jackson, C. (2020). How do deep-water volcanoes grow? *Earth and Planetary Science Letter*, 542.

Taufiq, H. (2015). *Argumentasi dan Validitas.* Yogyakarta: Darqin.

Ullo, S. L., Mohan, A., Sebastianelli, A., Ahamed, S. E., Kumar, B., Dwivedi, R., & Sinha, G. R. (2021). A new mask R-CNN-based method for improved landslide detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14*, 3799-3810.

Virkhansa, C. F., Setiawan, B. D., & Dewi, C. (2019). Klasifikasi Status Gunung Berapi dengan Metode Learning Vector Quantization (LVQ). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 3(7)*, 7119-7126.

Wahid, F. (2014). The Antecedents And Impacts of a Green Eprocurement Infrastructure: Evidence From The Indonesian Public Sector. *International Journal of internet Protocol Technology*, *7*(4), 210-218.

Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence, 2(1)*, 29-36.

Winata, P. A. (2022). Klasifikasi Naive Bayes Keparahan Trauma Pasien Menggunakan Data Neuro Cognitive Dan Data Physiologic Dengan Python. *UNEJ e-Proceeding*, 99-108.

Wu, T.-H., Wang, T.-W., & Liu, Y.-Q. (2021). Real-time vehicle and distance detection based on improved yolo v5 network. *3rd World Symposium on Artificial Intelligence*, 24-28.

Yuliyanto, S., Amani, N. F., Akhyar, F., & Usman, K. (2022). SISTEM INSPEKSI PERMUKAAN BAJA BERBASIS DEEP LEARNING MENGGUNAKAN METODE ANCHOR-FREE. *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer, 2(3)*, 184-190.

Zukhri, Z. (2014). *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Andi Publisher.

Anantharama, R., Velazquez, M., & Lee, Y. (2018). Utilizing Mask R-CNN for Detection and Segmentation of Oral Diseases. *EEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2197–2204.

Cao, L., Zheng, X., & Fang, L. (2023). The Semantic Segmentation of Standing Tree Images Based on the Yolo V7 Deep Learning Algorithm. *Electronics 2023, Vol. 12, Page 929*, *12*(4), 929. https://doi.org/10.3390/ELECTRONICS12040929

Griffiths, R. W. (2003). The Dynamics of Lava Flows. *Https://Doi.Org/10.1146/Annurev.Fluid.32.1.477*, *32*, 477–518. https://doi.org/10.1146/ANNUREV.FLUID.32.1.477

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

Lawal, O. M. (2023). YOLOv5-LiNet: A lightweight network for fruits instance segmentation. *PLOS ONE*, *18*(3), e0282297. https://doi.org/10.1371/JOURNAL.PONE.0282297

MAGMA Indonesia (Multiplatform Application for Geohazard Mitigation and Assessment in Indonesia). (n.d.). Retrieved November 23, 2022, from https://magma.esdm.go.id/v1/gunung-api/gallery

Ngantung, R. K., & Pakereng, M. A. I. (2021). Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *5*(3), 1052. https://doi.org/10.30865/mib.v5i3.3054

Pitriyani, E. (2019). Rekomendasi Pembimbing dan Penguji Tugas Akhir Menggunakan Vector Space Model. *Prosiding Nasional Rekayasa Teknologi Industridan Informasi XIV*, 317–323. http://journal.itny.ac.id/index.php/ReTII

Qiao, Y., Truman, M., & Sukkarieh, S. (2019a). Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Computers and Electronics in Agriculture*, *165*, 104958. https://doi.org/10.1016/J.COMPAG.2019.104958

Qiao, Y., Truman, M., & Sukkarieh, S. (2019b). Cattle segmentation and contour extraction based on Mask R-CNN for precision livestock farming. *Computers and Electronics in Agriculture*, *165*. https://doi.org/10.1016/j.compag.2019.104958

Sun, T., Chen, H., Liu, H., Lou, H., & Duan, X. (2023). *HPS-YOLOv7: A High Precision Small Object Detection Algorithm*. https://doi.org/10.21203/rs.3.rs-2813484/v1

Sun, T., Xing, H., Cao, S., Zhang, Y., Fan, S., & Liu, P. (2022). A novel detection method for hot spots of photovoltaic (PV) panels using improved anchors and prediction heads of YOLOv5 network. *Energy Reports*, *8*, 1219–1229. https://doi.org/10.1016/J.EGYR.2022.08.130

Ullo, S., Mohan, A., Sebastianelli, A., Ahamed, S., Kumar, B., Dwivedi, R., & Sinha, G. (2021a). A New Mask R-CNN-Based Method for Improved Landslide Detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 3799–3810. https://doi.org/10.1109/JSTARS.2021.3064981

Ullo, S., Mohan, A., Sebastianelli, A., Ahamed, S., Kumar, B., Dwivedi, R., & Sinha, G. (2021b). A New Mask R-CNN-Based Method for Improved Landslide Detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, 3799–3810. https://doi.org/10.1109/JSTARS.2021.3064981

Zhou, J., Zeng, S., Chen, Y., Kang, Z., Li, H., & Sheng, Z. (2023). A Method of Polished Rice Image Segmentation Based on YO-LACTS for Quality Detection. *Agriculture 2023, Vol. 13, Page 182*, *13*(1), 182. https://doi.org/10.3390/AGRICULTURE13010182