**LAPORAN AKHIR PROJEK**

**SISTEM DETEKSI NOMINAL UANG RUPIAH REAL-TIME MENGGUNAKAN ALGORITMA YOLO PADA JETSON ORIN NANO BERBASIS STREAMING KAMERA**

**(24STEE03X309)**

**SEMESTER GANJIL 2025**



|  |  |
| --- | --- |
| **Disusun Oleh:** |  |
| **Made Vero Putra Vrasta Purnawan** | **(2305541004)** |
| **Naufal Mochtar** | **(2305541055)** |
| **Rafif Wafa Saputra** | **(2305541076)** |

**LABORATORIUM TEKNIK DIGITAL DAN  
MIKROPROSESOR PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2025**

**PERCOBAAN I**

**PERCOBAAN INSTALASI NVIDIA JETPACK PADA JETSON ORIN NANO**

* 1. **Tujuan Percobaan**

1. Melakukan instalasi NVIDIA JetPack pada perangkat Jetson Orin Nano secara sistematis dan terstruktur.
2. Memastikan Jetson Orin Nano berfungsi dengan baik setelah proses instalasi, meliputi sistem operasi, GPU, dan akselerasi komputasi.
   1. **Alat dan Bahan**

1. Jetson Orin Nano

2. Keyboard

3. Webcam

4. Kabel Jumper Female to Female

5. Layar LCD

6. SD Card

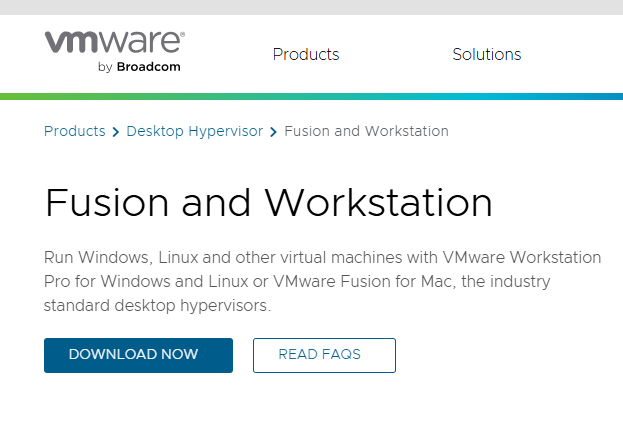
* 1. **Pendahulauan**

Jetson Orin Nano merupakan salah satu platform komputasi embedded dari NVIDIA yang dirancang untuk mendukung aplikasi kecerdasan buatan dengan kebutuhan komputasi tinggi pada perangkat berdaya rendah. Perangkat ini dilengkapi dengan GPU NVIDIA yang mendukung akselerasi CUDA, sehingga sangat cocok digunakan untuk pengembangan aplikasi seperti computer vision, object detection, dan pemrosesan data berbasis deep learning.

Untuk dapat memanfaatkan seluruh kemampuan perangkat Jetson Orin Nano, diperlukan lingkungan perangkat lunak yang terintegrasi dan optimal. NVIDIA menyediakan JetPack sebagai paket perangkat lunak resmi yang mencakup sistem operasi berbasis Linux, driver perangkat keras, serta berbagai pustaka pendukung seperti CUDA, cuDNN, dan TensorRT. JetPack berperan penting dalam memastikan kompatibilitas antara perangkat keras dan perangkat lunak yang digunakan.

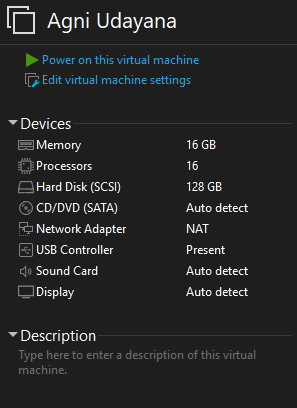
Percobaan ini difokuskan pada proses instalasi JetPack ke Jetson Orin Nano sebagai tahap awal dalam penggunaan perangkat tersebut. Melalui percobaan ini, diharapkan sistem dapat terpasang dengan baik dan siap digunakan untuk pengembangan aplikasi kecerdasan buatan pada tahap selanjutnya.

* 1. **Langkah Percobaan**
     1. **Instal Virtual Machine (VMware)**

****

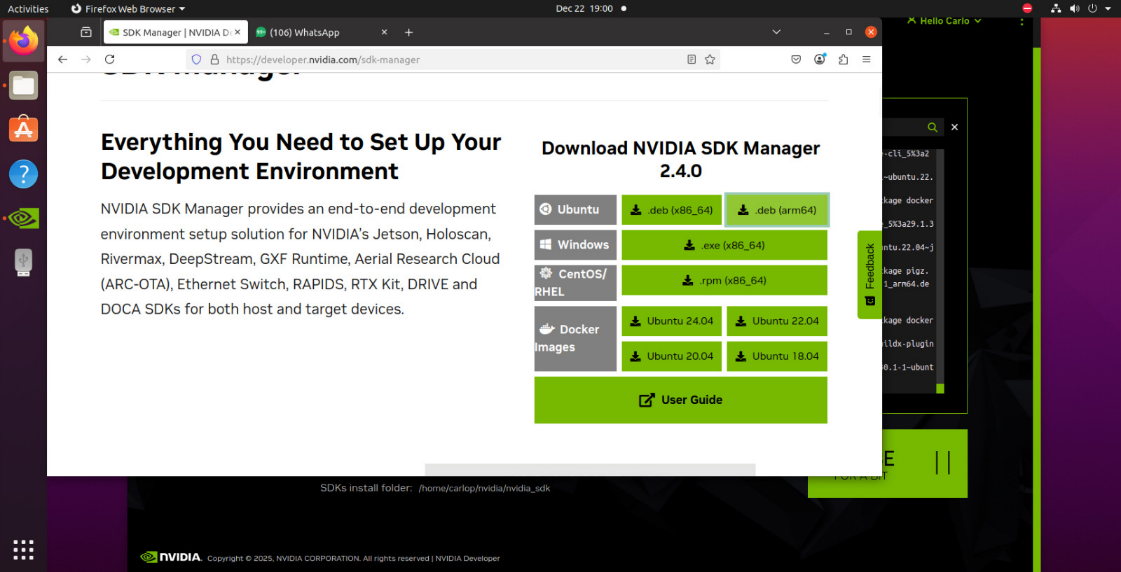
VMware dipilih sebagai platform virtualisasi karena memiliki kompatibilitas yang baik terhadap perangkat USB, sehingga meminimalkan terjadinya *USB port issue* saat digunakan dalam proses flashing JetPack ke perangkat Jetson. Stabilitas koneksi USB pada VMware sangat penting untuk memastikan proses flashing berjalan tanpa gangguan.

* + 1. **Instal Ubuntu 20.04 pada VMware**

****

Sistem operasi Ubuntu digunakan karena proses flashing JetPack menggunakan NVIDIA SDK Manager hanya dapat dilakukan pada sistem operasi berbasis Linux, khususnya Ubuntu. Untuk menginstal JetPack 6 yang kompatibel dengan Jetson Orin Nano, diperlukan Ubuntu versi 20.04. Pada tahap instalasi Ubuntu di dalam virtual machine, kapasitas penyimpanan dikonfigurasikan lebih dari 80 GB guna memastikan seluruh komponen JetPack dapat diunduh.

* + 1. **Instal NVIDIA SDK Manager pada VMware**

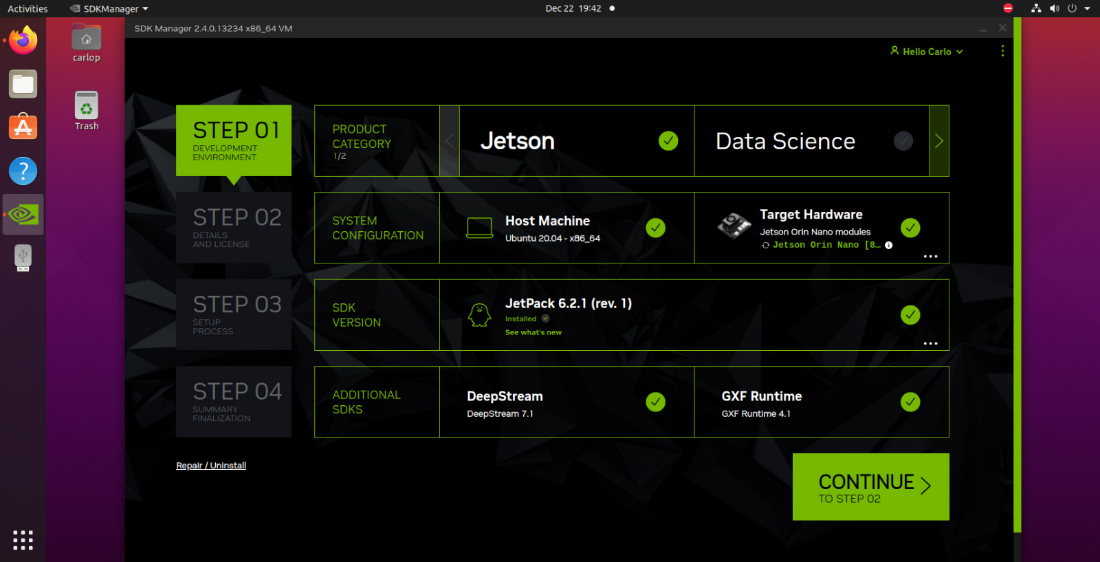
****

NVIDIA SDK Manager digunakan untuk mengunduh dan menginstal JetPack SDK yang mencakup sistem operasi Jetson, driver perangkat keras, CUDA, cuDNN, serta berbagai pustaka kecerdasan buatan lainnya. Seluruh opsi instalasi dicentang agar seluruh komponen terpasang secara otomatis tanpa memerlukan instalasi manual tambahan. Penggunaan SDK Manager bertujuan untuk mempermudah proses konfigurasi Jetson Orin Nano secara terintegrasi dan sistematis.

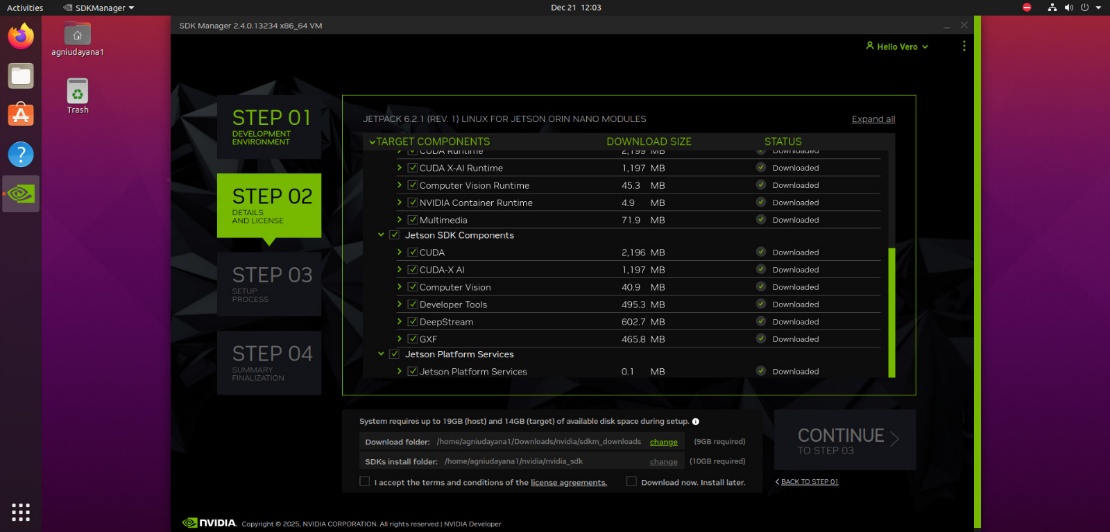
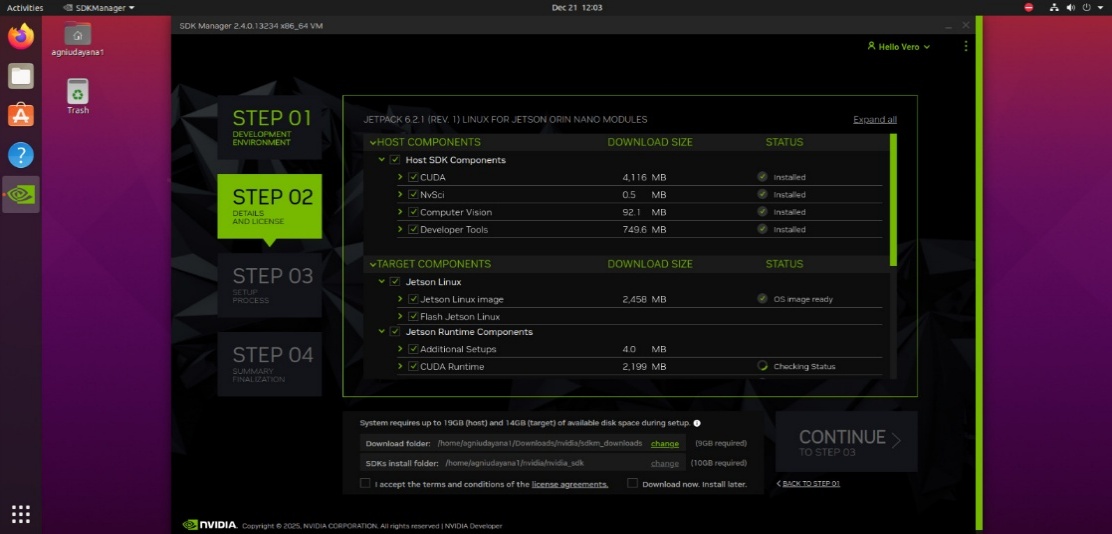
* + 1. **Flash JetPack 6 ke Jatson**

Proses flashing dilakukan dengan menghubungkan kabel USB dari perangkat *host* ke Jetson Orin Nano. Tahapan ini bertujuan untuk menginstal sistem operasi JetPack ke dalam perangkat Jetson agar siap digunakan dengan dukungan penuh terhadap komputasi AI dan akselerasi GPU. Pada tahap ini, diperlukan SD Card dengan kapasitas minimal 64 GB serta kabel USB yang digunakan sebagai media komunikasi selama proses flashing berlangsung. Saat ingin melaukan flashing ke jetson, jumper pin GND dan FC REC agar Jeston bisa masuk ke mode Recovery. Berikut Langkah-langkah untuk melakuakn flashing JetPack ke Jetson:

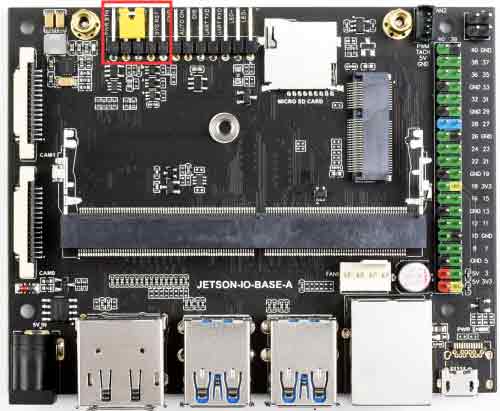
1. Buka aplikasi NVIDIA SDK Manager, lalu login menggunakan akun NVIDIA.
2. Terapkan pengaturan instalasi pada SDK Manager sesuai konfigurasi dibawah



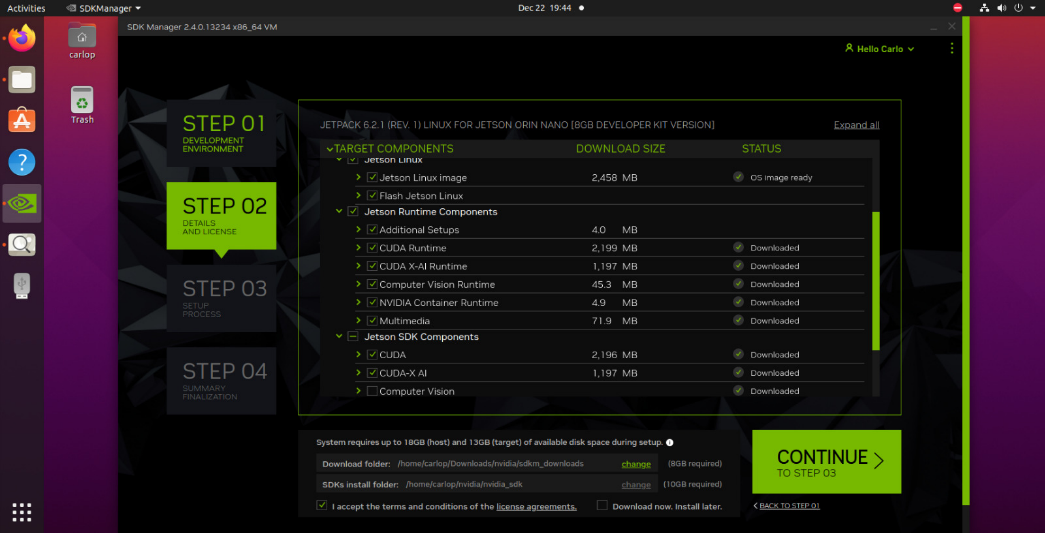
1. Klik tombol Continue, kemudian centang seluruh opsi instalasi agar semua komponen terpasang secara otomatis.



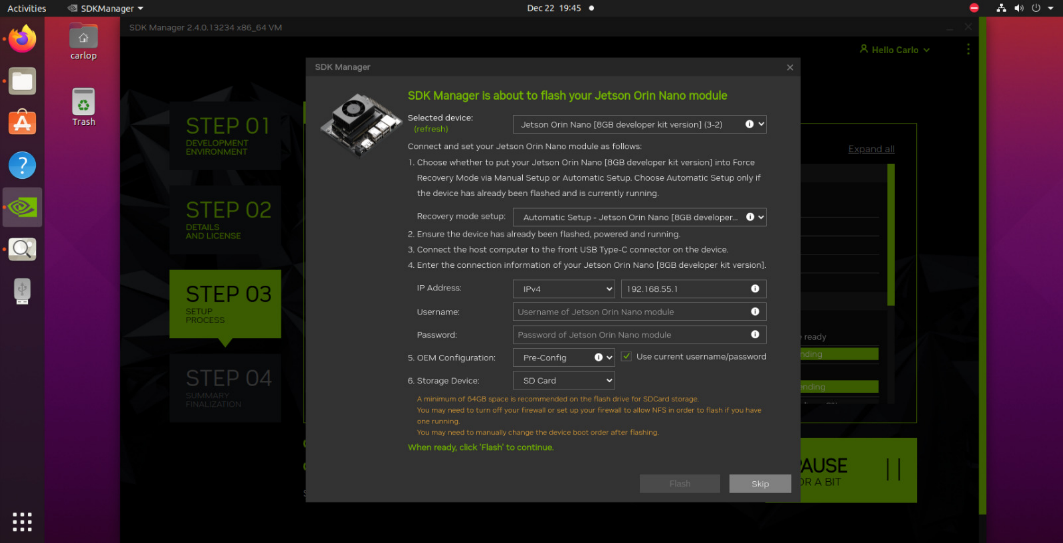
1. Pastikan perangkat Jetson Orin Nano masuk ke Recovery Mode dengan cara menghubungkan (short) pin GND dan FC REC menggunakan jumper.



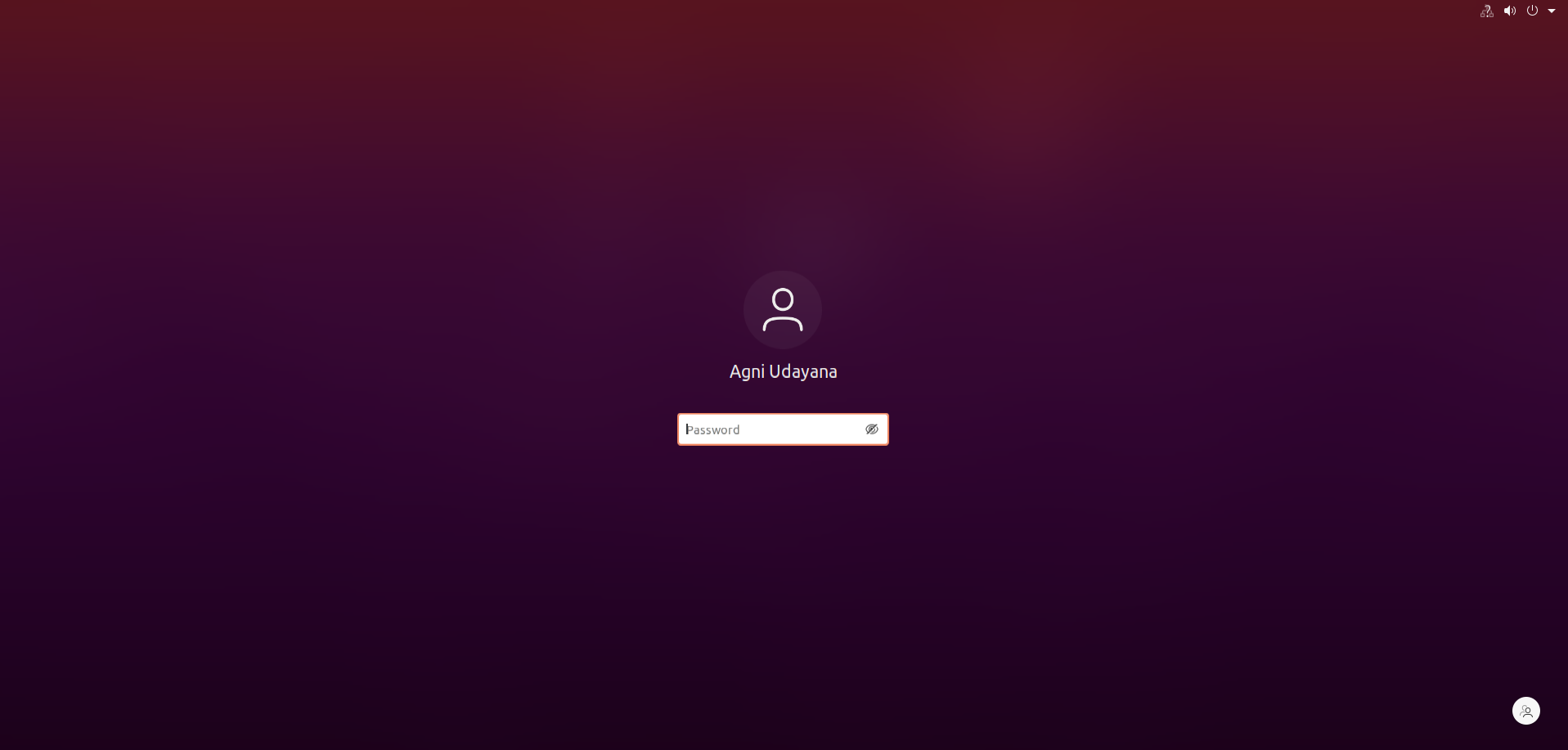
1. Hubungkan Jetson Orin Nano ke perangkat host menggunakan kabel USB.
2. Klik tombol Continue setelah seluruh berkas berhasil diunduh pada perangkat host (VM).



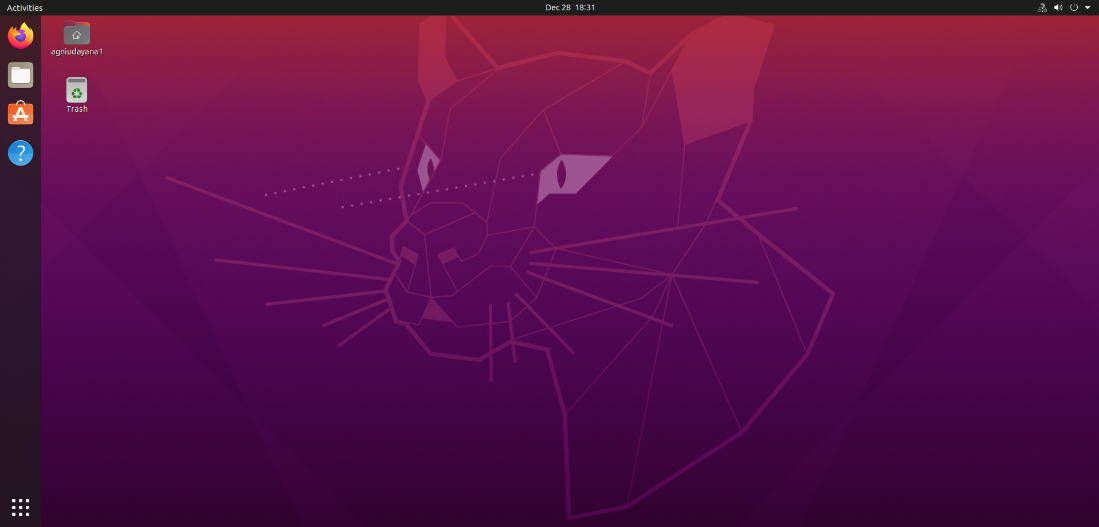
1. Masukkan username dan password untuk sistem Jetson Orin Nano, kemudian tekan tombol Flash dan tunggu hingga proses flashing selesai sepenuhnya.



* 1. **Hasil Percobaan**
     1. **Tampilan Booting**

****

* + 1. **Tampilan Dekstop setelah Booting**



* 1. **Analisa**

Berdasarkan hasil percobaan yang telah dilakukan, proses instalasi NVIDIA JetPack pada perangkat Jetson Orin Nano berhasil dilaksanakan tanpa mengalami kendala yang signifikan. Keberhasilan instalasi ditunjukkan dengan sistem operasi yang dapat berjalan secara normal, perangkat dapat melakukan proses booting dengan baik, serta lingkungan desktop dapat diakses sesuai dengan spesifikasi yang diharapkan.

Selain itu, komponen utama yang disertakan dalam JetPack, seperti driver GPU, CUDA, dan pustaka pendukung lainnya, terpasang dengan benar sehingga perangkat siap digunakan untuk pengembangan aplikasi berbasis kecerdasan buatan. Hal ini menunjukkan bahwa prosedur instalasi yang diterapkan telah sesuai dengan panduan resmi dan kompatibel dengan perangkat keras Jetson Orin Nano yang digunakan.

Keberhasilan percobaan ini juga menandakan bahwa Jetson Orin Nano telah berada dalam kondisi siap pakai sebagai platform komputasi embedded. Dengan lingkungan JetPack yang telah terpasang, perangkat mampu mendukung pengembangan aplikasi lanjutan seperti computer vision dan deep learning secara optimal.

* 1. **Kesimpulan**

Berdasarkan percobaan yang telah dilakukan, dapat disimpulkan bahwa instalasi NVIDIA JetPack pada Jetson Orin Nano berhasil dilakukan dengan baik. Seluruh tahapan instalasi dapat dijalankan sesuai prosedur, dan sistem mampu beroperasi secara normal setelah proses instalasi selesai. Dengan demikian, Jetson Orin Nano telah siap digunakan sebagai lingkungan pengembangan aplikasi kecerdasan buatan dan pemrosesan data berbasis GPU.

* 1. **Saran**

Berdasarkan hasil percobaan, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Percobaan selanjutnya dapat dikembangkan dengan mengimplementasikan aplikasi sederhana, seperti deteksi objek atau pengolahan citra, guna menguji performa Jetson Orin Nano secara lebih mendalam.
2. Disarankan untuk selalu menggunakan versi JetPack yang sesuai dan stabil agar kompatibilitas perangkat keras dan perangkat lunak tetap terjaga.

**PERCOBAAN II**

**PERCOBAAN DETEKSI OBJEK NOMINAL UANG RUPIAH MENGGUNAKAN ALGORITMA YOLO**

* 1. **Tujuan Percobaan**

Tujuan yang ingin dicapai dalam percobaan ini adalah sebagai berikut:

1. Menerapkan algoritma YOLO untuk membangun sistem deteksi nominal uang rupiah berbasis computer vision.
2. Mengetahui tingkat akurasi sistem dalam mengklasifikasikan nominal uang rupiah berdasarkan hasil pengujian sederhana.
   1. **Alat dan Bahan**

1. Jetson Orin Nano

2. Keyboard

3. Mouse

4. Webcam

5. Layar LCD

6. Uang Kertas Rupiah

* 1. **Pendahulauan**

Perkembangan teknologi computer vision dan deep learning menunjukkan kemajuan yang pesat dalam dekade terakhir, terutama dalam kemampuan sistem untuk secara otomatis memahami dan mengenali objek visual. Salah satu kontribusi signifikan dalam bidang object detection adalah algoritma You Only Look Once (YOLO), yang terkenal karena kemampuannya melakukan deteksi objek secara real-time dengan kecepatan dan akurasi yang tinggi dibandingkan metode deteksi tradisional dua tahap (two-stage detector) seperti R-CNN. Pendekatan YOLO mendefinisikan deteksi objek sebagai masalah regresi yang langsung memprediksi lokasi dan kelas objek dalam satu langkah inferensi, sehingga efisien untuk aplikasi real-time pada citra atau video streaming.

Kebutuhan akan sistem deteksi nominal uang rupiah secara otomatis semakin meningkat seiring dengan upaya digitalisasi layanan dan otomatisasi berbagai proses bisnis. Identifikasi nominal uang secara manual memiliki keterbatasan, seperti potensi kesalahan manusia, ketergantungan pada kondisi pencahayaan dan latar belakang, serta kurangnya efisiensi saat bekerja dalam skala besar. Selain itu, sistem otomatis ini memiliki peran penting dalam mendukung aksesibilitas, misalnya sebagai alat bantu bagi penyandang disabilitas penglihatan yang memerlukan bantuan pengenalan nominal uang secara mandiri. Oleh karena itu, pengembangan sistem berbasis computer vision yang mampu melakukan deteksi nominal uang rupiah secara akurat dan responsif menjadi sangat relevan dalam berbagai aplikasi praktis.

Dalam konteks implementasi, object detection berbasis YOLO dinilai sebagai pendekatan yang efektif karena mampu mengenali dan melokalisasi objek dalam citra secara simultan dengan performa yang mendukung pemrosesan real-time. Berbagai studi menunjukkan bahwa varian Varian YOLO terus dioptimalkan untuk memenuhi tuntutan kinerja model pada perangkat edge seperti Jetson, termasuk kemampuan mempertahankan FPS tinggi dan akurasi deteksi yang memadai pada perangkat komputasi rendah.

Untuk mendukung inferensi model YOLO secara langsung tanpa ketergantungan penuh pada cloud computing, diperlukan perangkat edge AI yang memiliki kapabilitas pemrosesan deep learning di sisi perangkat. Jetson Orin Nano merupakan salah satu perangkat edge AI yang dirancang untuk mempercepat penerapan aplikasi visi komputer berbasis AI secara lokal. Perangkat ini dilengkapi dengan akselerator AI dan unit pemrosesan GPU yang mampu menjalankan model YOLO dengan throughput dan efisiensi energi yang baik pada data citra secara real-time.

Berdasarkan uraian tersebut, penelitian ini difokuskan pada pengembangan sistem deteksi nominal uang rupiah berbasis computer vision dan algoritma YOLO yang diimplementasikan pada Jetson Orin Nano sebagai edge AI device. Penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan teknologi deteksi uang yang lebih akurat, efisien, dan aplikatif dalam berbagai skenario praktis seperti otomatisasi layanan, sistem pembayaran cerdas, dan alat bantu tunanetra.

* 1. **Dasar Teori**
     1. **Sistem Embedded**

Sistem *embedded* merupakan perangkat komputer yang didesain khusus untuk melakukan tugas yang lebih spesifik. Sistem *embedded* terdiri dari perangkat keras atau *hardware* dan perangkat lunak atau *software*. Perangkat keras pada sistem *embedded* meliputi *Single Board Microcomputer* dan juga komponen I/O yang dipasang di dalamnya seperti, *LED*, *LCD*, *keypad*, dan komponen aktuator lainnya. Sedangkan perangkat lunak sistem *embedded* merupakan *listing* program untuk penggerak sistem *embedded* itu sendiri.

* + 1. **Deep Learning**

Deep learning adalah salah satu cabang dari machine learning yang menggunakan struktur jaringan saraf tiruan (artificial neural network) dengan banyak lapisan untuk memproses dan menganalisis data. Tujuan utamanya adalah untuk meniru cara kerja otak manusia dalam mempelajari pola, mengenali objek, memahami bahasa, dan membuat keputusan.

* + 1. ***Object Detection***

Object detection merupakan teknik dalam bidang computer vision yang digunakan untuk mengidentifikasi dan menentukan lokasi keberadaan objek pada citra maupun video. Algoritma object detection umumnya memanfaatkan metode machine learning atau deep learning untuk menghasilkan hasil deteksi yang akurat dan bermakna.

Ketika manusia mengamati citra atau video, manusia dapat mengenali dan menentukan posisi objek yang menjadi perhatian dalam waktu yang sangat singkat. Tujuan dari object detection adalah mereplikasi kemampuan kognitif tersebut ke dalam sistem komputer agar mampu mengenali dan melokalisasi objek secara otomatis.

* + 1. **YOLO**

Algoritma YOLO (You Only Look Once) adalah algoritma deep learning canggih untuk deteksi objek secara real-time, yang bekerja dengan memproses seluruh gambar dalam satu lintasan jaringan saraf konvolusional (CNN) tunggal, memprediksi kotak pembatas (bounding box) dan probabilitas kelas objek secara bersamaan dan cepat, menjadikannya sangat efektif untuk aplikasi seperti mobil otonom, pengawasan keamanan, dan analisis video.

* + 1. **Jetson Orin Nano**

Jetson Orin Nano adalah komputer mini bertenaga dari NVIDIA untuk pengembangan kecerdasan buatan (AI) di perangkat edge, menawarkan performa tinggi dalam ukuran kecil, ideal untuk robotika, sistem penglihatan cerdas, dan AI generatif, dengan varian Super yang meningkatkan kinerja secara signifikan hanya dengan pembaruan perangkat lunak. Perangkat ini menggunakan CPU Arm Cortex-A78AE dan GPU arsitektur Ampere, serta mendukung berbagai model AI mutakhir melalui ekosistem NVIDIA.

* + 1. **Web Camera**

Web camera atau yang biasa dikenal dengan webcam, adalah kamera yang gambarnya bisa di akses menggunakan world wide web (www), program instant messaging, atau aplikasi komunikasi dengan tampilan video pada PC. Webcam juga digambarkan sebagai kamera video digital yang sengaja didesain sebagai kamera dengan resolusi rendah. Pada beberapa webcam, ada yang di lengkapi dengan software yang mampu mendeteksi pergerakan dan suara. Dengan software tersebut, memungkinkan PC yang terhubung ke kamera untuk mengamati pergerakan dan suara, serta merekamnya ketika terdeteksi. Hasil rekaman ini bisa disimpan pada komputer, e mail atau di upload ke internet

* 1. **Langkah Percobaan**
     1. **Instal Ultralytics YOLO**

Ultralytics YOLO digunakan sebagai pustaka utama dalam implementasi algoritma YOLO untuk sistem deteksi objek. Instalasi Ultralytics dilakukan dengan menyertakan dependensi tambahan (optional dependencies) agar model PyTorch dapat diekspor ke berbagai format, khususnya NVIDIA TensorRT. Penggunaan TensorRT bertujuan untuk memperoleh performa inferensi maksimum pada perangkat Jetson Orin Nano. Buka Terminal lalu jalankan baris kode dibawah:

sudo apt update

sudo apt install python3-pip -y

pip install -U pip

pip install ultralytics[export]

sudo reboot

* + 1. **Instal Python**

Python digunakan sebagai bahasa pemrograman utama dalam pengembangan sistem deteksi objek. Instalasi Python bertujuan untuk mendukung proses pengolahan data, pemodelan deep learning, serta eksekusi algoritma deteksi objek berbasis YOLO pada perangkat Jetson Orin Nano. Buka Terminal lalu jalankan baris kode dibawah:

sudo apt install python3-pip -y

* + 1. **Instalasi Framework PyTorch**

PyTorch merupakan framework deep learning yang digunakan sebagai dasar pengembangan dan pelatihan model YOLO. Instalasi PyTorch dilakukan dengan menyesuaikan arsitektur perangkat Jetson Orin Nano yang berbasis ARM64 agar mampu memanfaatkan akselerasi GPU NVIDIA secara optimal. Oleh karena itu, PyTorch dipasang menggunakan pre-built wheel yang kompatibel dengan JetPack 6 dan CUDA pada Jetson. Buka Terminal lalu jalankan baris kode dibawah:

pip install <https://github.com/ultralytics/assets/releases/download/v0.0.0/torch-2.5.0a0+872d972e41.nv24.08-cp310-cp310-linux_aarch64.whl>

pip install <https://github.com/ultralytics/assets/releases/download/v0.0.0/torchvision-0.20.0a0+afc54f7-cp310-cp310-linux_aarch64.whl>

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/arm64/cuda-keyring\_1.1-1\_all.deb

sudo dpkg -i cuda-keyring\_1.1-1\_all.deb

sudo apt-get update

sudo apt-get -y install libcusparselt0 libcusparselt-dev

* + 1. **Instal Visual Studio Code**

****

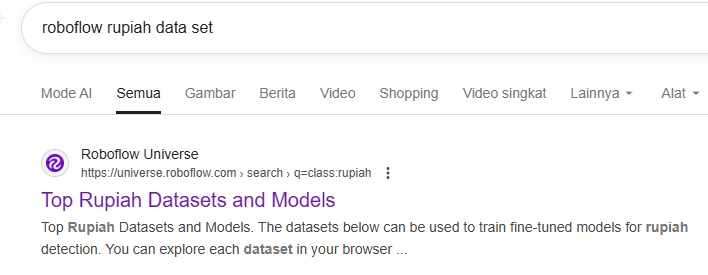
Visual Studio Code digunakan sebagai Integrated Development Environment (IDE) untuk menulis, mengelola, dan menjalankan kode program. Penggunaan Visual Studio Code bertujuan untuk meningkatkan efisiensi pengembangan perangkat lunak serta mempermudah proses debugging dan pengelolaan proyek.

* + 1. **Pemodelan Sistem Menggunakan Roboflow**

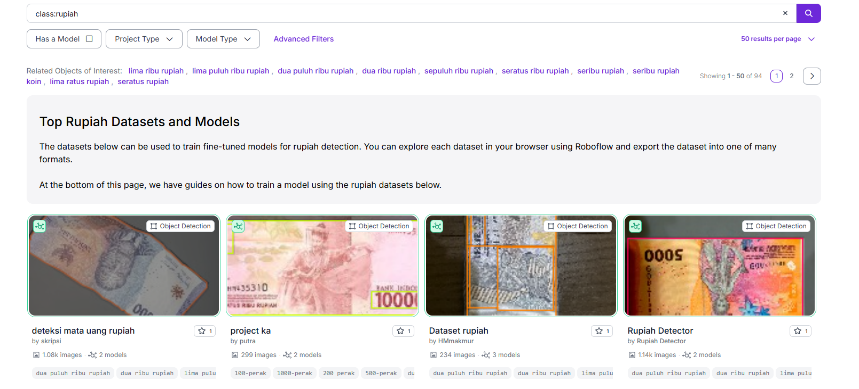
Pemodelan sistem deteksi objek nominal uang rupiah dilakukan menggunakan platform Roboflow. Pada tahap ini, dataset uang rupiah dikumpulkan dan dikelola dengan melakukan proses pelabelan pada setiap citra berdasarkan kelas nominal yang telah ditentukan. Selain itu, dilakukan pengaturan struktur dataset, seperti pembagian data latih, data validasi, dan data uji, serta penyesuaian format dataset agar sesuai dengan kebutuhan algoritma YOLO. Penggunaan Roboflow bertujuan untuk mempermudah manajemen dataset serta memastikan data yang digunakan telah terstruktur dengan baik sebelum masuk ke tahap pelatihan model.

Berikut adalah langkah-langkah untuk mendapatkan dataset dari Roboflow:

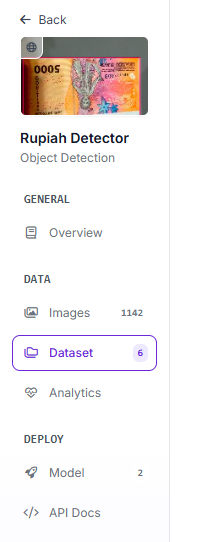
* 1. Cari Dataset di Roboflow Universe menggunakan kata kunci pencarian yang relevan, seperti "roboflow rupiah data set



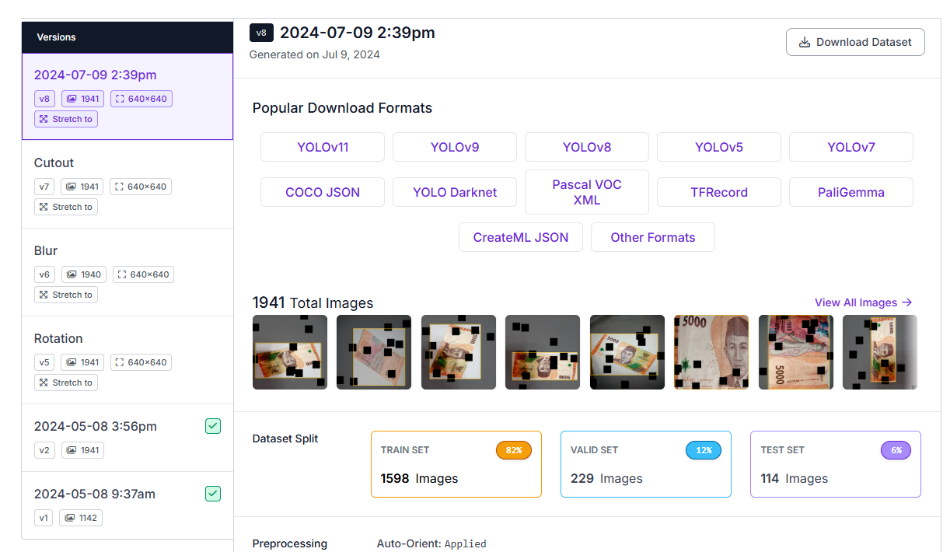
* 1. Pilih dataset yang sesuai dengan kebutuhan sistem deteksi objek.



* 1. Buka halaman dataset yang telah dipilih untuk melihat detail dan struktur data.



* 1. Tentukan format dataset YOLOv5 sebagai format yang digunakan.



* 1. Download dataset dalam bentuk zip

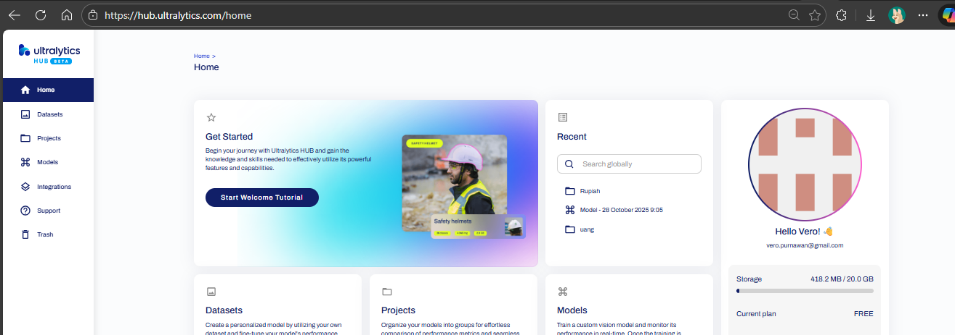


* + 1. **Pelatihan Model Menggunakan Ultralytics YOLO**

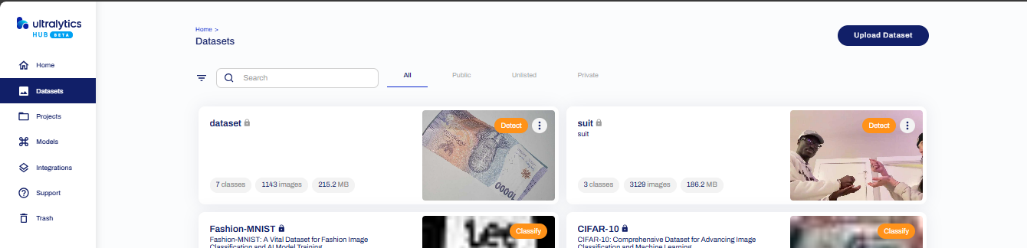
Proses pelatihan model dilakukan menggunakan framework Ultralytics YOLO. Dataset yang telah dipersiapkan pada tahap sebelumnya digunakan untuk melatih model agar mampu mengenali dan mengklasifikasikan nominal uang rupiah berdasarkan karakteristik visualnya. Pelatihan dilakukan dengan menentukan parameter pelatihan seperti jumlah epoch dan ukuran batch. Hasil dari tahap ini berupa model YOLO terlatih yang siap digunakan untuk proses deteksi objek.

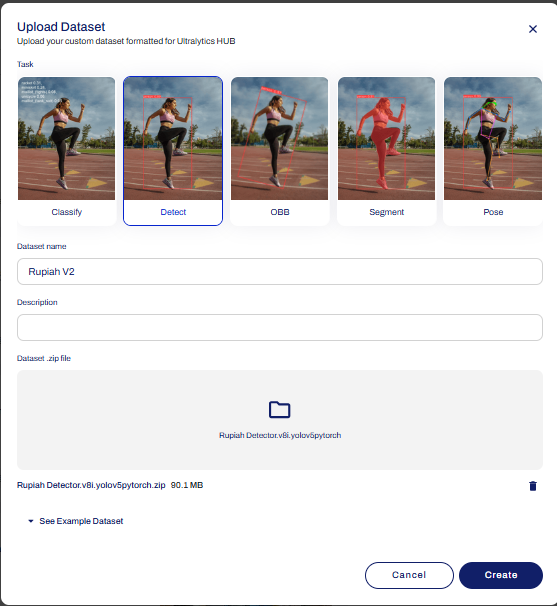
Berikut cara melakukan training model dari dataset yang dimiliki di Ultralytics:

1. Akses situs Ultralytics Hub melalui alamat <https://hub.ultralytics.com/home> dan lakukan login.

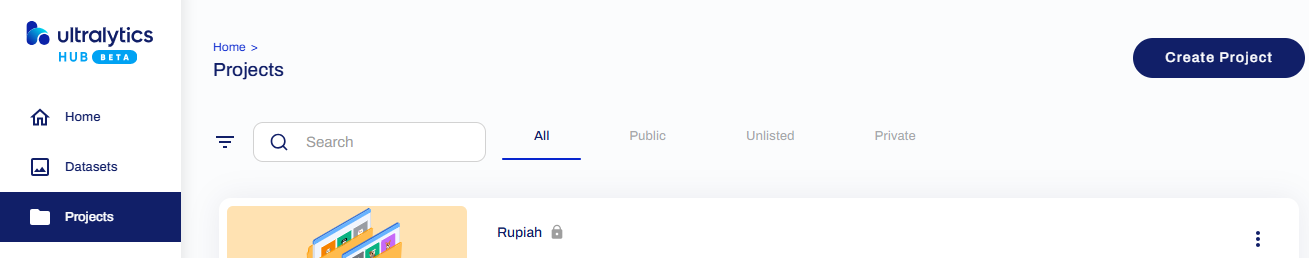


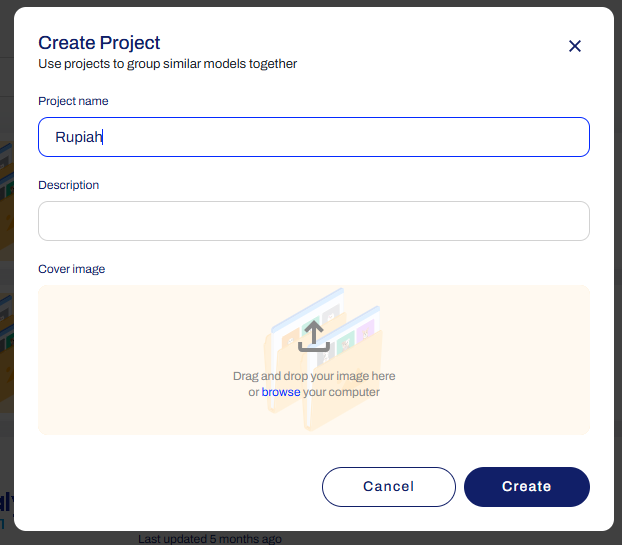
1. Masuk ke menu Dataset, lalu unggah dataset yang telah diunduh dari Roboflow. Pilih jenis *task* Object Detection (Detect) dan beri nama dataset sesuai kebutuhan. Tunggu hingga proses unggah dataset selesai.

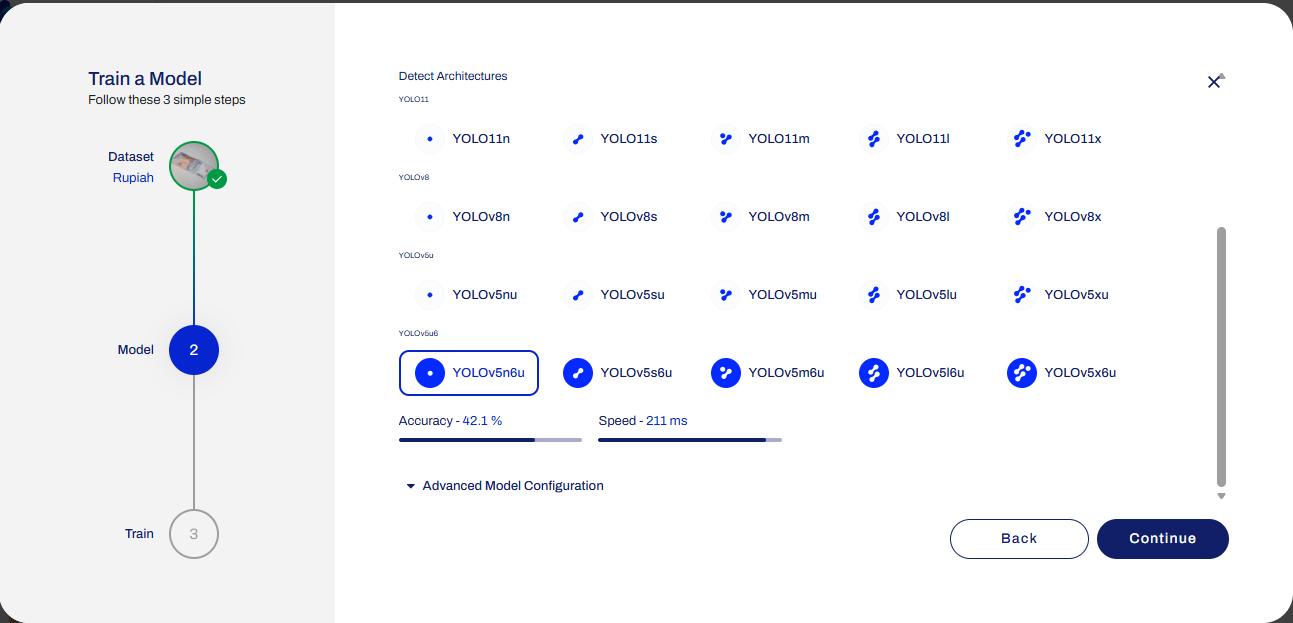


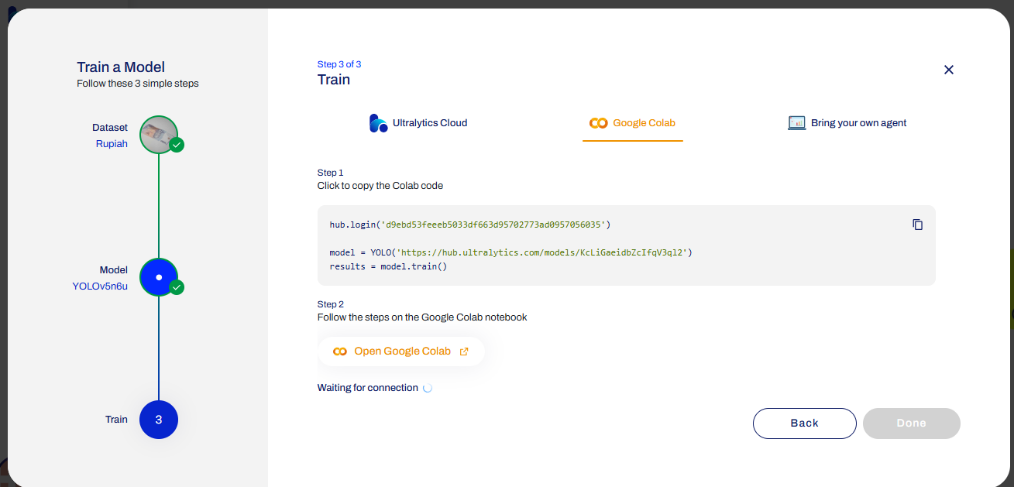


1. Bila datset sudah selesai diupload, pergi ke menu projek dan buat projek baru

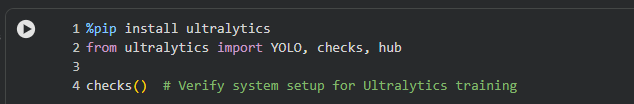




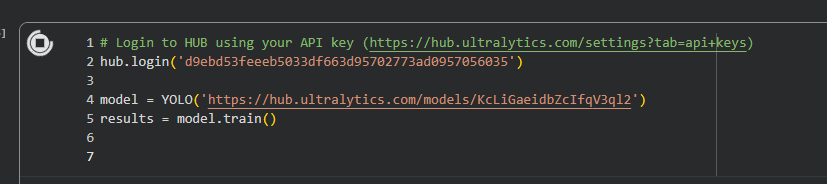
1. Pilih dataset yang telah diunggah dan tentukan model YOLO yang akan digunakan. 
2. Lakukan proses pelatihan (training) dataset menggunakan Google Colab. Sebelum membuka Google Colab copy baris kode step 1



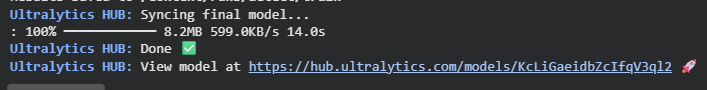
1. Jalankan baris kode untuk menginstal library Ultralytics pada proyek.



1. Paste baris kode step 1 lalu jalankan baris kode tersebut untuk melakukan proses training model.



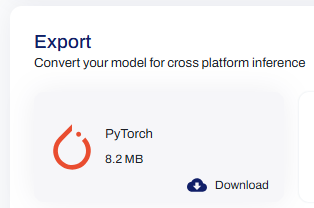
1. Buka hasil model setelah proses training selesai.



1. Masuk ke menu Deploy, lalu pilih opsi Export PyTorch.



1. Unduh model hasil pelatihan untuk digunakan pada tahap implementasi.



* + 1. Implementasi dan Menjalankan Program

Model YOLO yang telah dilatih selanjutnya diimplementasikan pada perangkat Jetson Orin Nano. Program deteksi dijalankan dengan memanfaatkan kamera sebagai input citra. Sistem akan memproses citra secara real-time dan menampilkan hasil deteksi berupa bounding box dan label nominal uang rupiah. Tahap ini bertujuan untuk memastikan bahwa sistem dapat berjalan dengan baik pada perangkat Jetson Orin Nano serta mampu melakukan deteksi nominal uang rupiah secara real-time sesuai dengan perancangan sistem. Berikut adalah baris kode yang kami gunakan

from ultralytics import YOLO

import cv2

model = YOLO("uang.pt")

cap = cv2.VideoCapture(0)  # USB camera

while True:

    ret, frame = cap.read()

    if not ret:

        break

    results = model(frame)

    annotated\_frame = results[0].plot()

    cv2.imshow("YOLO - Jetson USB Camera", annotated\_frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

cap.release()

cv2.destroyAllWindows()

* 1. **Hasil Percobaan**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nominal Uang** | **Hasil Deteksi** | **Skor Confidence** | **Status** | **Foto Hasil Deteksi** | **FPS** | **Bukti** |
| 1 | Rp1.000 | Rp1.000 | 83% | Berhasil |  | 32,051 |  |
| 2 | Rp2.000 | Rp10.000 | 80% | Gagal |  | 28,409 |  |
| 3 | Rp5.000 | Rp5.000 | 84% | Berhasil |  | 33,333 |  |
| 4 | Rp10.000 | Rp10.000 | 77% | Berhasil |  | 32,894 |  |
| 5 | Rp20.000 | Rp20.000 | 39% | Berhasil |  | 32,051 |  |
| 6 | Rp50.000 | Rp50.000 | 95% | Berhasil |  | 29,850 |  |
| 7 | Rp100.000 | Rp100.000 | 76% | Berhasil |  | 31,545 |  |

* 1. **Analisa**

Berdasarkan hasil pengujian sistem deteksi objek menggunakan algoritma YOLO, sebagian besar nominal uang rupiah berhasil terdeteksi dengan benar. Nominal Rp1.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, dan Rp100.000 menunjukkan hasil deteksi yang sesuai antara nominal aktual dan hasil klasifikasi sistem. Nilai confidence pada nominal-nominal tersebut berada pada rentang 77% hingga 95%, yang menandakan bahwa model mampu mengenali ciri visual utama uang kertas dengan tingkat keyakinan yang cukup tinggi.

Keberhasilan deteksi ini menunjukkan bahwa model YOLO telah mampu mempelajari fitur-fitur visual penting, seperti warna dominan, pola desain, serta karakteristik visual tokoh dan ornamen pada uang kertas. Nominal Rp50.000 memperoleh nilai confidence tertinggi sebesar 95%, yang mengindikasikan bahwa fitur visual pada nominal tersebut memiliki tingkat keunikan yang tinggi dan mudah dibedakan oleh model. Sementara itu, nominal Rp20.000 berhasil terdeteksi meskipun memiliki nilai confidence yang relatif rendah, yaitu 39%, yang menunjukkan bahwa meskipun klasifikasi benar, tingkat keyakinan model terhadap objek tersebut masih terbatas.

Di sisi lain, hasil deteksi pada uang nominal Rp2.000 mengalami kesalahan klasifikasi, di mana sistem mendeteksi uang tersebut sebagai Rp10.000 dengan nilai confidence sebesar 80%, sehingga status deteksi dinyatakan gagal. Kesalahan ini diasumsikan terjadi akibat keterbatasan pada tahap pembuatan dan pelatihan model, khususnya pada distribusi dan kualitas dataset. Jumlah data latih uang Rp2.000 yang lebih sedikit dibandingkan nominal lain dapat menyebabkan model kurang optimal dalam mempelajari karakteristik visual spesifik dari nominal tersebut.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem deteksi objek berbasis YOLO telah bekerja dengan baik dalam mengenali sebagian besar nominal uang rupiah. Kegagalan pada nominal Rp2.000 menjadi indikator bahwa peningkatan kualitas dataset, penyeimbangan jumlah data antar kelas, serta penyempurnaan proses pelatihan model masih diperlukan untuk meningkatkan akurasi dan konsistensi deteksi pada seluruh nominal.

Selain akurasi deteksi, performa sistem juga dievaluasi berdasarkan nilai *Frame Per Second* (FPS) untuk mengetahui kemampuan sistem dalam memproses citra secara real-time. FPS menunjukkan jumlah frame yang dapat diproses sistem dalam satu detik, sehingga semakin tinggi nilai FPS, semakin responsif sistem deteksi objek yang dihasilkan.

Perhitungan FPS didasarkan pada waktu pemrosesan satu frame. Secara umum, FPS dapat dihitung menggunakan persamaan sebagai berikut:

Karena satu detik setara dengan 1000 milidetik (ms), maka persamaan tersebut dapat dituliskan menjadi:

Sebagai contoh, apabila waktu pemrosesan rata-rata satu frame adalah 31 ms/frame, maka FPS yang dihasilkan adalah:

Berdasarkan hasil pengujian, sistem menghasilkan nilai FPS rata-rata pada rentang 28 FPS hingga 33 FPS untuk berbagai nominal uang rupiah. Nilai FPS tertinggi tercatat pada pengujian nominal Rp5.000 sebesar 33,333 FPS, sedangkan nilai FPS terendah tercatat pada nominal Rp2.000 sebesar 28,409 FPS. Variasi nilai FPS ini dipengaruhi oleh kompleksitas visual objek, proses inferensi model YOLO, serta beban komputasi yang dijalankan oleh Jetson Orin Nano pada saat pengujian.

Secara umum, nilai FPS yang berada di atas 25 FPS menunjukkan bahwa sistem telah memenuhi kriteria *real-time processing*. Hal ini membuktikan bahwa Jetson Orin Nano mampu menjalankan inferensi model YOLO dengan baik menggunakan input kamera USB tanpa mengalami penurunan performa yang signifikan. Dengan demikian, sistem deteksi nominal uang rupiah yang dikembangkan tidak hanya akurat, tetapi juga responsif dan layak digunakan untuk aplikasi deteksi objek secara langsung.

* 1. **Kesimpulan**

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem deteksi objek nominal uang rupiah menggunakan algoritma YOLO pada perangkat Jetson Orin Nano, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma YOLO berhasil diterapkan untuk membangun sistem deteksi dan klasifikasi nominal uang rupiah berbasis computer vision yang mampu berjalan secara real-time pada perangkat edge AI Jetson Orin Nano.
2. Sistem yang dikembangkan mampu mendeteksi sebagian besar nominal uang kertas rupiah dengan tingkat keberhasilan yang baik. Nominal Rp1.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, dan Rp100.000 berhasil terdeteksi sesuai dengan kelas yang benar, dengan nilai confidence yang umumnya berada di atas 70%.
3. Kesalahan deteksi terjadi pada nominal Rp2.000, yang terklasifikasi sebagai Rp10.000. Hal ini menunjukkan bahwa performa sistem masih dipengaruhi oleh kualitas dan distribusi dataset, khususnya ketidakseimbangan jumlah data latih antar kelas serta kemiripan visual antar nominal tertentu.
4. Implementasi sistem menggunakan kamera USB sebagai input citra menunjukkan bahwa model YOLO terlatih dapat dijalankan secara langsung pada Jetson Orin Nano tanpa ketergantungan pada cloud computing, sehingga sistem bersifat mandiri dan efisien.
5. Secara keseluruhan, percobaan ini membuktikan bahwa pendekatan deteksi objek berbasis YOLO pada perangkat edge AI memiliki potensi yang baik untuk diaplikasikan pada sistem pengenalan nominal uang rupiah, meskipun masih diperlukan penyempurnaan untuk meningkatkan konsistensi dan keakuratan deteksi.
   1. **Saran**

Berdasarkan hasil percobaan dan keterbatasan yang ditemukan, beberapa saran yang dapat diberikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Dataset yang digunakan sebaiknya diperbanyak dan diseimbangkan jumlahnya pada setiap kelas nominal uang agar model dapat mempelajari karakteristik visual masing-masing nominal secara lebih optimal.
2. Perlu dilakukan proses data augmentation yang lebih variatif, seperti perubahan pencahayaan, rotasi, dan sudut pengambilan gambar, guna meningkatkan kemampuan generalisasi model terhadap kondisi lingkungan yang berbeda.
3. Evaluasi performa sistem dapat dikembangkan lebih lanjut dengan menggunakan metrik kuantitatif yang lebih lengkap, seperti precision, recall, dan mean Average Precision (mAP), untuk memperoleh gambaran performa model secara lebih objektif.
4. Sistem dapat dikembangkan untuk mendukung input video secara kontinu atau integrasi dengan antarmuka audio, sehingga dapat dimanfaatkan sebagai alat bantu bagi penyandang tunanetra dalam mengenali nominal uang rupiah.
5. Untuk peningkatan performa inferensi, model YOLO yang telah dilatih dapat diekspor ke format TensorRT agar pemanfaatan akselerasi GPU pada Jetson Orin Nano menjadi lebih optimal.

**DAFTAR PUSTAKA**

Ali, J. (2016) Sistem security webcam dengan menggunakan Microsoft Visual Basic 6.0. Jurnal Teknologi dan Sistem Informasi UNIVRAB, 1(2), Juli. ISSN 2477-2062.

MathWorks (2023) Apa itu deteksi objek?. Tersedia di: <https://www.mathworks.com/discovery/object-detection.html> (Diakses pada: 23 Desember 2025).

Roboflow (2023) Panduan model YOLO: algoritma dan arsitektur. Tersedia di: <https://blog.roboflow.com/guide-to-yolo-models/> (Diakses pada: 23 Desember 2025).

Seeed Studio (2024) Pembaruan Jetson Orin Nano Developer Kit menjadi Super Kit. Tersedia di: <https://wiki.seeedstudio.com/update_orin_nano_developer_kit_to_super_kit/> (Diakses pada: 23 Desember 2025).

Telkom University (2022) Deep learning: pengertian, cara kerja, dan penerapannya. Tersedia di: <https://bds.telkomuniversity.ac.id/deep-learning-pengertian-cara-kerja-dan-penerapannya/> (Diakses pada: 23 Desember 2025).