

BAB III

METODE PENELITIAN

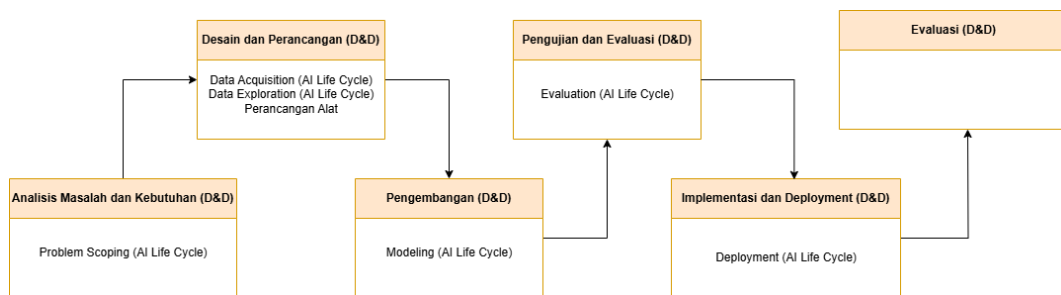
3.1. Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode *Design and Development* (D&D) dengan pendekatan *AI Project Cycle*. Metode Penelitian *Design and Development* (D&D) merupakan pendekatan sistematis yang digunakan untuk merancang, mengembangkan, dan mengevaluasi produk atau sistem baru. Metode ini sering diterapkan dalam penelitian yang berfokus pada inovasi teknologi, pengembangan perangkat lunak (Richey & Klein, 2014). Sementara *AI Project Cycle* merupakan pendekatan yang melibatkan berbagai tahap untuk mengembangkan dan mengimplementasikan sistem *artificial intelligence* (Ng dkk., 2022). Kedua metode ini dipilih karena mencakup serangkaian tahapan yang sistematis dan berulang untuk merancang, mengembangkan, menguji, dan mengimplementasikan teknologi *artificial intelligence*. *Design and Development* (D&D) lebih berfokus pada pengembangan perangkat keras dan perangkat lunak tertanam (*embedded systems*) dan *AI Project Cycle* menekankan pada pengembangan model kecerdasan buatan (AI), mulai dari pengumpulan data, pengembangan model, evaluasi, hingga implementasi (Aurelia C & Prasetya D., 2023)

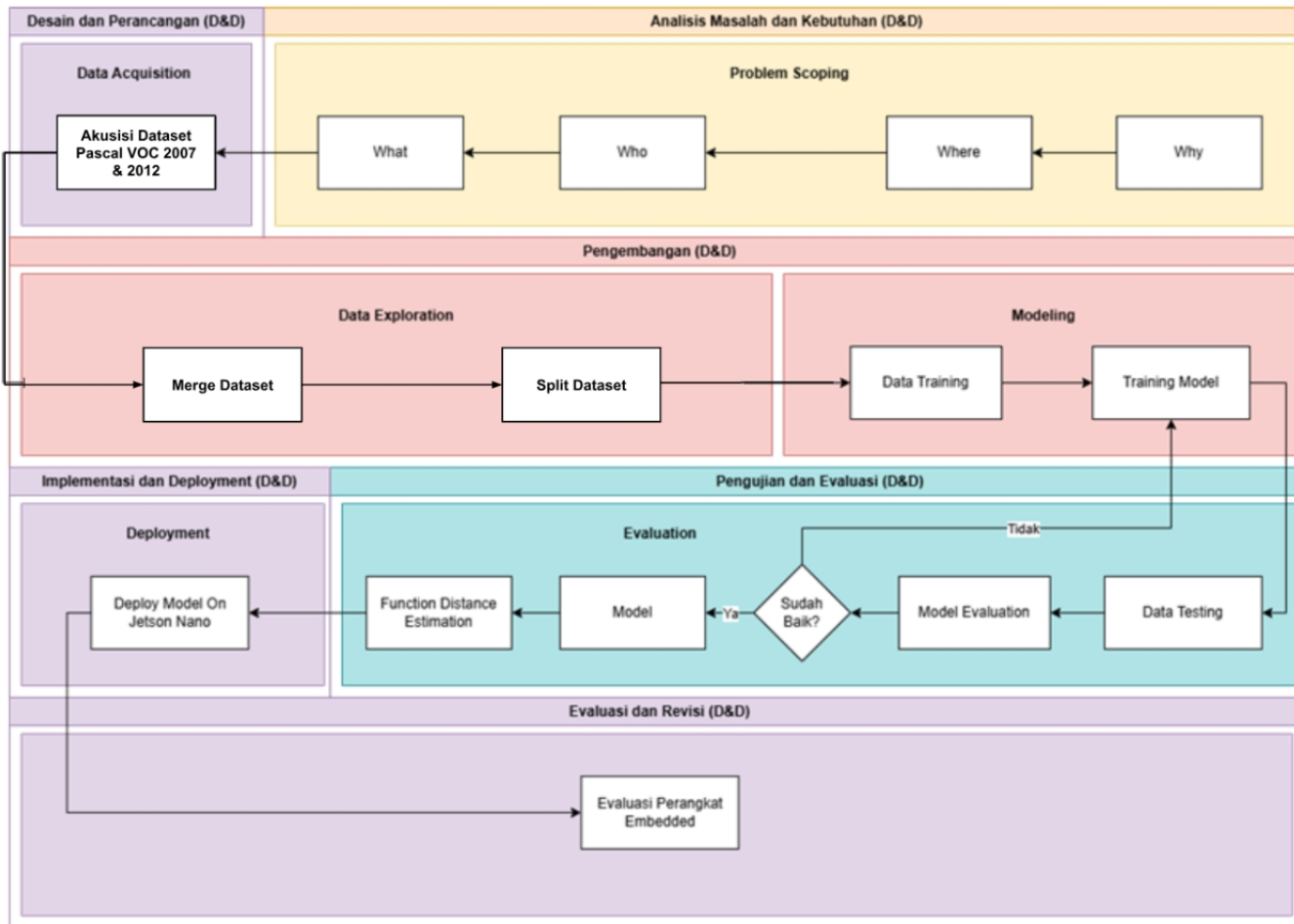
Gambar 3.1 dan Gambar 3.2 merupakan ilustrasi integrasi metode penelitian *Design and Development* (D&D) dengan pendekatan *AI Project Cycle*. Dalam pendekatan ini, tahap-tahap D&D seperti Analisis Masalah dan Kebutuhan, Desain dan Perancangan, Pengembangan, Pengujian dan Evaluasi, Evaluasi dan Revisi, serta Implementasi dan Deployment diintegrasikan dengan tahap-tahap *AI Project Cycle*, yaitu *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modeling*, *Evaluation*, dan *Deployment*.

Tahapan Analisis Masalah dan Kebutuhan (D&D) melibatkan tahapan *Problem Scoping* dari *AI Project Cycle*, di mana masalah yang dihadapi diidentifikasi dan tujuan dari pengembangan AI maupun rancangan sistem perangkat *embedded* dirumuskan. Tahapan Desain dan Perancangan (D&D) serta Pengembangan (D&D) diselaraskan dengan tahapan *Data Acquisition*, *Data Exploration*, Perancangan alat dan *Modeling* dari *AI Project Cycle*, di mana data dikumpulkan dan model AI serta rancangan perangkat *embedded* dikembangkan. Tahapan Pengujian dan Evaluasi (D&D) diintegrasikan dengan

tahapan *Evaluation* dari *AI Project Cycle*, yang menjadi tahapan untuk menguji dan mengevaluasi model yang dikembangkan. Tahapan Implementasi dan *Deployment* (D&D) selaras dengan tahapan *Deployment* dari *AI Project Cycle*, Tahapan ini merupakan tahapan di mana model AI yang telah melewati proses iterasi pengembangan diimplementasikan ke dalam sistem perangkat *embedded*. Kemudian tahapan terakhir yaitu Evaluasi dan Revisi (D&D) melibatkan revisi berdasarkan hasil pengujian dari implementasi AI pada perangkat.



Gambar 3. 1 Metode Penelitian



Gambar 3. 2 Alur Penelitian

3.1.1. *Problem Scouping*

Tahap *problem scouping* merupakan tahapan yang meliputi identifikasi dan perumusan masalah yang ingin diselesaikan oleh pada penelitian. Tabel 3.1. menampilkan aspek – aspek yang dianalisis pada tahapan *problem scouping*.

Tabel 3. 1
Problem Scouping

| Aspek | Keterangan |
|-------|--|
| What | Apa yang menjadi fokus penelitian. |
| Who | Siapa yang menjadi target atau pengguna akhir dari solusi yang dikembangkan. |
| Where | Lokasi atau konteks di mana solusi akan diterapkan. |
| Why | Alasan dan tujuan dari penelitian ini. |

Aspek-aspek yang diidentifikasi tersebut merupakan aspek yang ditujukan untuk mendefinisikan ruang lingkup suatu proyek penelitian dan memastikan bahwa penelitian terarah dan relevan dengan kebutuhan dan konteks yang ingin dicapai.

3.1.2. *Data Acquisition*

Tahap data *acquisition* merupakan tahapan *data collecting* dalam penelitian yang bertujuan untuk mengumpulkan dataset yang akan digunakan sebagai bahan dasar dalam pengembangan model deteksi objek. Tahapan *acquisition* terdiri dari tahapan identifikasi sumber data yang relevan, mengumpulkan data dari berbagai sumber, dan melakukan pengolahan awal untuk membersihkan dan memformat data tersebut agar siap digunakan dalam pemodelan (De Silva & Alahakoon, 2022). Data yang digunakan dalam penelitian ini adalah dataset yang berisikan kumpulan gambar *common object* sehari-hari yang digunakan sebagai input untuk sistem deteksi objek dengan arsitektur *deep learning*. Dataset yang digunakan adalah dataset *Pascal VOC*, yang merupakan dataset gambar-gambar bersifat *open source* dan dapat diakses secara bebas.

3.1.3. *Data Exploration*

Tahapan data *exploration* merupakan tahapan dimana data yang telah dikumpulkan akan dianalisis untuk memahami karakteristik dan strukturnya. Tahapan Ini melibatkan fase eksplorasi statistik deskriptif, visualisasi data, dan analisis eksploratif lainnya untuk mendapatkan wawasan tentang data dan memandu keputusan pemodelan

berikutnya.(De Silva & Alahakoon, 2022). Fase eksplorasi statistik deskriptif merupakan langkah yang melibatkan pengumpulan informasi statistik dasar dari dataset, seperti rasio dataset, jumlah total gambar, jumlah objek per kelas, serta rata-rata dan distribusi objek. Kemudian fase visualisasi data merupakan fase yang ditujukan untuk membuat interpretasi dataset secara visual. Visualisasi dataset ini bisa termasuk visualisasi histogram jumlah objek per kelas, box plot yang menunjukkan variasi ukuran objek, atau bahkan preview sampel gambar untuk setiap kategori.

3.1.4. *Modeling*

Tahap *Modeling* berkaitan dengan pengembangan model AI berdasarkan data yang telah diolah. Pemilihan algoritme yang sesuai, pelatihan model menggunakan set data latih, dan penyetelan parameter untuk mengoptimalkan kinerja model termasuk kedalam fase *modelling* (De Silva & Alahakoon, 2022). Tahapan *modelling* merupakan tahapan yang ditujukan untuk membuat model deteksi objek berbasis *deep learning* yang akan digunakan pada penelitian. Arsitektur yang digunakan dan dibangun pada tahapan *modelling* ialah arsitektur *Single Shot Multibox Detector*, dan YOLOv8n. Model akan dilatih terlebih dahulu menggunakan *training set*, kemudian divalidasi dengan *validation set*. proses pelatihan akan melibatkan penyesuaian *hyperparameter*. *Hyperparameter* yang disesuaikan meliputi jumlah *epochs*, ukuran *batch size*, dan jenis *optimizer*. Perancangan model deteksi objek dengan menggunakan arsitektur *Single Shot Multibox Detector dan YOLOv8n* akan dilakukan pada platform Kaggle. Kaggle merupakan platform layanan pelatihan model secara gratis yang berguna untuk pengembangan model. *Environment* yang dibangun untuk pelatihan model menggunakan beberapa *library* untuk mendukung proses pengembangan dan pelatihan, Adapun *library* yang digunakan meliputi Ultralytics 8.1.44, OpenCV 4.6.0, Torch 1.8.0, Torchvision 0.9.0, Python versi 3.10.

3.1.5. *Evaluation*

Tahap evaluasi ditujukan untuk menguji model menggunakan test set untuk mengukur kinerja model setelah pelatihan. Evaluasi model dilakukan dengan menggunakan lima matrix evaluasi, yaitu mAP, *accuracy*, *precision*, *recall*, *f1-score*, dan *mean Average Precision*. *Accuracy* merupakan metrik yang mengukur proporsi prediksi

yang benar dari total prediksi yang dibuat oleh model. *Precision* merupakan metrik yang mengukur seberapa akurat model dalam memprediksi positif dari semua prediksi positif yang dibuat. *Recall* merupakan metrik yang mengukur seberapa baik model dalam mendeteksi semua kasus positif yang sebenarnya. *F1-Score* merupakan rata-rata harmonik dari *precision* dan *recall*, sehingga memberikan gambaran gabungan tentang kedua metrik ini. *Mean Average Precision* merupakan metrik khusus untuk tugas deteksi objek, metrik ini digunakan untuk mengukur seberapa baik model dalam mendeteksi dan mengklasifikasikan objek dalam gambar.

Ketika model sudah mencapai performa yang memadai, model tersebut akan diimplementasikan program *distance estimation* sehingga ketika objek terdeteksi, estimasi jarak terhadap objek tersebut dapat diperoleh. Model *deep learning* digunakan untuk mengenali dan mengklasifikasikan objek dalam gambar, serta menentukan lokasi (*bounding box*) dari objek-objek tersebut. Kemudian fungsi tambahan *distance estimation* diimplementasikan dengan mengambil nilai berdasarkan ukuran *bounding box* dan informasi tinggi objek yang sebenarnya.

3.1.6. *Deployment*

Pada penelitian, tahap *deployment* merupakan tahapan yang ditujukan untuk mengintegrasikan model kedalam perangkat keras sehingga menghasilkan *output* akhir berupa perangkat *embedded system*. Model akan diimplementasikan kedalam perangkat *single board computer* yang berfungsi sebagai perangkat komputasi. Jetson Nano dipilih sebagai *single board computer* (SBC) yang digunakan karena kemampuannya yang kuat dalam pemrosesan komputasi, terutama dalam tugas-tugas yang melibatkan implementasi *deep learning*.

3.2. Rancangan Sistem

Sistem yang akan dirancang dalam penelitian ini adalah sebuah sistem berbasis *deep learning* yang dirancang khusus sebagai alat bantu pengenalan objek bagi tunanetra. Sistem ini akan diintegrasikan ke dalam perangkat keras berupa *Single Board Computer* (SBC) yang merupakan komputer mini dengan kemampuan komputasi yang cukup untuk menjalankan model *deep learning* secara *real-time*. Oleh karena itu, kajian pustaka ini akan mencakup pembahasan mengenai *flowchart system* pengembangan model dan

flowchart sistem perangkat, *block diagram*, *architecture diagram*, *hardware system diagram*, *wiring diagram* dan *design 3D*.

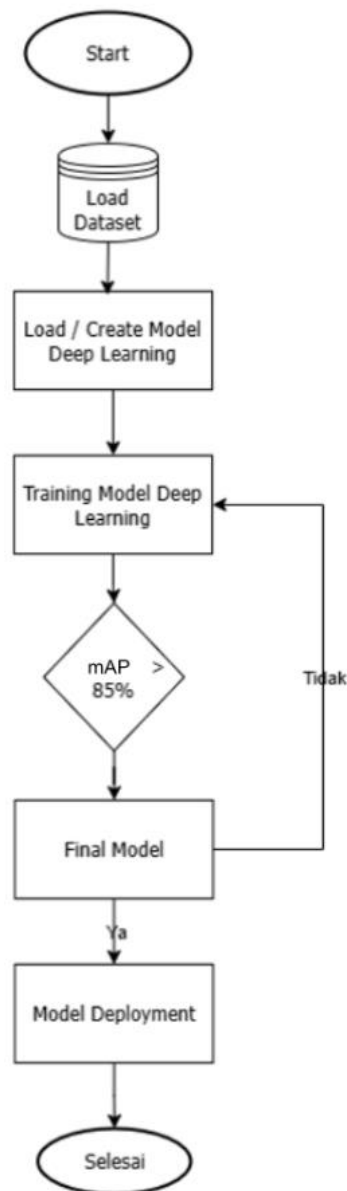
3.2.1. *Flowchart System*

Flowchart adalah representasi grafis dari sebuah proses atau alur kerja yang menggunakan berbagai bentuk simbol untuk menunjukkan langkah-langkah, operasi, atau keputusan dalam suatu proses. *Flowchart* digunakan untuk memvisualisasikan dan menganalisis proses, membuat dokumentasi, merancang sistem baru,

A. *Flowchart Pengembangan Model*

Gambar 3.3. merupakan ilustrasi *flowchart* dari pengembangan model *deep learning*. Proses pengembangan model deteksi objek menggunakan *deep learning* dimulai dengan tahap *start* yang mencakup semua persiapan awal untuk pengembangan model. Selanjutnya, dataset yang akan digunakan diambil dan dimuat dalam langkah *load dataset*. Pada penelitian ini, dataset yang digunakan adalah *Pascal VOC* yang berisi beragam gambar objek sehari-hari. Kemudian masuk pada tahapan berikutnya yaitu *load / create model deep learning*, dimana model *deep learning* yang akan digunakan untuk pelatihan dibuat atau dimuat. Model ini bisa berupa model baru yang dikembangkan dari awal untuk arsitektur *single shot multibox detector* atau model *pre-trained* yang disesuaikan untuk arsitektur YOLOv8. Kemudian, model yang telah dibuat atau dimuat akan dilatih menggunakan *training dataset* dalam tahap *training model deep learning*. Proses ini melibatkan pengoptimalan parameter model berdasarkan data yang tersedia. Setelah pelatihan, model dievaluasi menggunakan dataset validasi untuk menentukan akurasi validasinya. Jika $mAP > 85\%$ maka model dianggap cukup baik dan siap untuk tahap berikutnya. Jika mAP validasi tidak mencapai ambang batas yang diinginkan, dilakukan evaluasi model *deep learning* yang mencakup *tuning hyperparameter*, atau modifikasi arsitektur model untuk meningkatkan kinerja. Setelah model berhasil mencapai akurasi validasi yang diinginkan, model tersebut ditetapkan sebagai *final model* yang merupakan versi model yang telah dioptimalkan dan siap untuk *deployment*. Pada tahap *distance estimation*, model yang telah dilatih dan dioptimalkan, kemudian akan diimplementasikan dengan *function distance estimation* untuk menghitung estimasi jarak objek yang terdeteksi. Kemudian pada tahapan *model deployment* dilakukan dengan

menerapkan *model final* pada perangkat komputasi sehingga hasil akhir berupa perangkat *embedded* yang utuh. Proses ini diakhiri dengan tahap *Finish*, dimana sistem siap digunakan dalam aplikasi nyata untuk deteksi objek dan estimasi jarak. Tahapan-tahapan ini memberikan gambaran jelas mengenai alur pengembangan dari awal hingga akhir, memastikan bahwa setiap langkah dilakukan secara terstruktur dan sistematis.



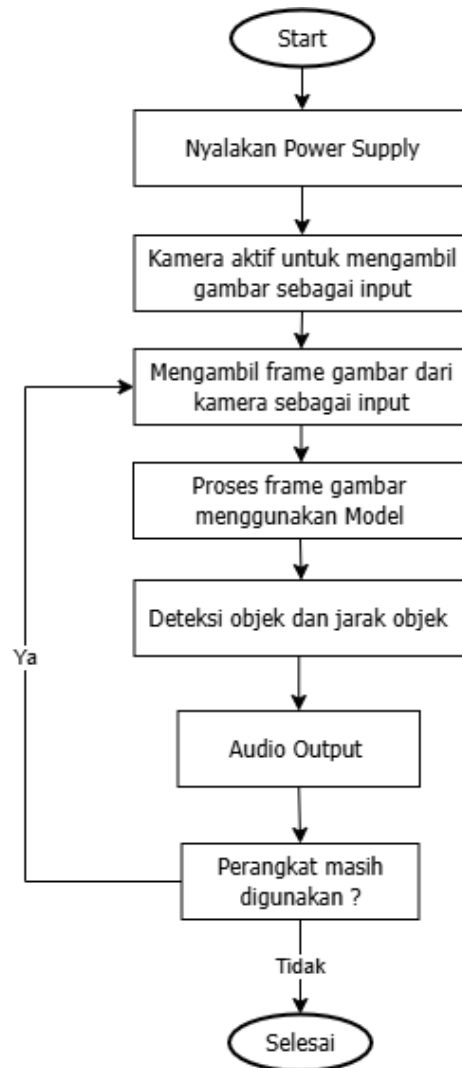
Gambar 3. 3 *Flowchart* Pengembangan Model

B. *Flowchart* Sistem Perangkat

Gambar 3.4. merupakan ilustrasi *Flowchart* yang menjelaskan tahapan operasional sistem deteksi objek pada perangkat. Proses dimulai dengan tahap *start*, dimana pengguna melakukan persiapan awal untuk mengaktifkan perangkat dengan menyalakan daya listrik pada perangkat yang berasal dari accu sebagai *power supply*. Langkah ini memastikan semua komponen sistem mendapatkan suplai daya yang diperlukan untuk beroperasi. Setelah perangkat menyala, kamera webcam pada perangkat akan aktif, webcam ini berfungsi untuk menangkap gambar dari lingkungan sekitar sebagai data input. Pada tahapan tersebut sistem akan mengambil frame gambar dari kamera sebagai input, di mana kamera terus-menerus akan mengambil frame gambar yang akan diproses lebih lanjut.

Pada tahap proses frame gambar menggunakan model pada perangkat komputasi, frame gambar yang diambil oleh kamera diproses menggunakan model deteksi objek *deep learning* yang telah diimplementasikan pada perangkat komputasi, Jetson Nano. Model ini bertugas untuk menganalisis gambar dan mengidentifikasi objek yang ada di dalam gambar. Setelah analisis dilakukan, deteksi objek dan jarak objek pada perangkat komputasi melibatkan proses di mana sistem mendeteksi objek serta memperkirakan jarak objek tersebut dari perangkat.

Output dari proses ini diubah menjadi *audio output*, di mana hasil deteksi objek dan estimasi jarak dikonversi menjadi *output* audio. *Output audio* ini disalurkan melalui perangkat *earphone*. Selama pengguna masih menggunakan perangkat, sistem akan terus melakukan iterasi atau pengulangan ke tahap pengambilan frame gambar dan proses identifikasi objek. Namun, jika pengguna menghentikan penggunaan perangkat, sistem akan berhenti beroperasi yang ditandai dengan tahap selesai.



Gambar 3.4 *Flowchart* Sistem Perangkat

3.2.2. *Block Diagram*

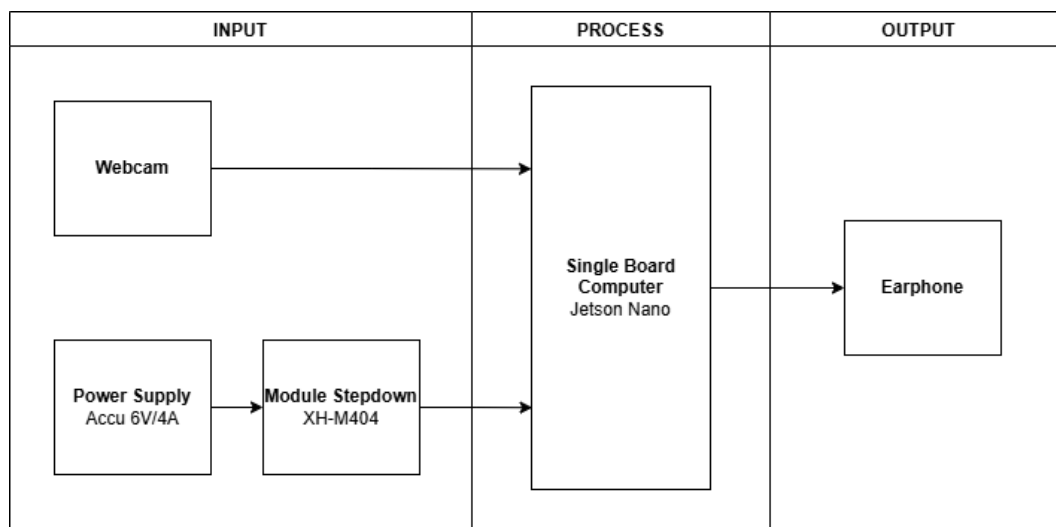
Gambar 3.5. merupakan representasi dari *block diagram* yang menggambarkan perangkat sistem deteksi objek yang terdiri dari tiga bagian yaitu *input*, *process*, dan *output*. Pada bagian *input*, terdapat komponen Webcam DNMR yang bertugas menangkap gambar atau video dari lingkungan sekitar dan mengirimkannya untuk dianalisis lebih lanjut kepada Jetson Nano sebagai perangkat komputasi. Konektivitas antara Webcam DNMR dengan Jetson Nano dilakukan melalui port USB. Webcam DNMR, yang memiliki antarmuka USB2.0, dihubungkan langsung ke salah satu port USB pada Jetson Nano. Sambungan ini memungkinkan webcam untuk mengirimkan data

video secara *real-time* ke Jetson Nano, yang kemudian dapat diproses oleh sistem komputasi.

Sistem pada perangkat didukung oleh *power supply* yang menggunakan Accu Motoled 6V/4A sebagai sumber daya untuk perangkat Jetson Nano. Akan tetapi karena Jetson Nano membutuhkan daya input sebesar 5V/4A untuk beroperasi, maka digunakan modul *stepdown* XH-M404 untuk menurunkan tegangan dari 6V menjadi 5V. Modul *stepdown* ini difungsikan untuk mengatur tegangan outputnya sehingga sesuai dengan kebutuhan Jetson Nano sehingga memastikan sistem mendapatkan daya yang stabil dan sesuai. Konektivitas daya ke Jetson Nano disalurkan melalui port jack DC yang tersedia pada perangkat Jetson Nano. Accu yang dihubungkan ke modul *stepdown* XH-M404 mengirimkan daya yang telah disesuaikan ke Jetson Nano melalui kabel dengan konektor jack DC.

Pada bagian *process*, *Single Board Computer* Jetson Nano berfungsi sebagai unit pemrosesan utama. Jetson Nano menerima data gambar dari webcam DNMR dan menggunakan algoritma model *deep learning* untuk mendeteksi dan mengidentifikasi objek yang ada dalam gambar. Selain itu, Jetson Nano juga melakukan pemrosesan penghitungan estimasi jarak objek yang terdeteksi. Hasil dari pemrosesan ini kemudian dikirim ke *output*.

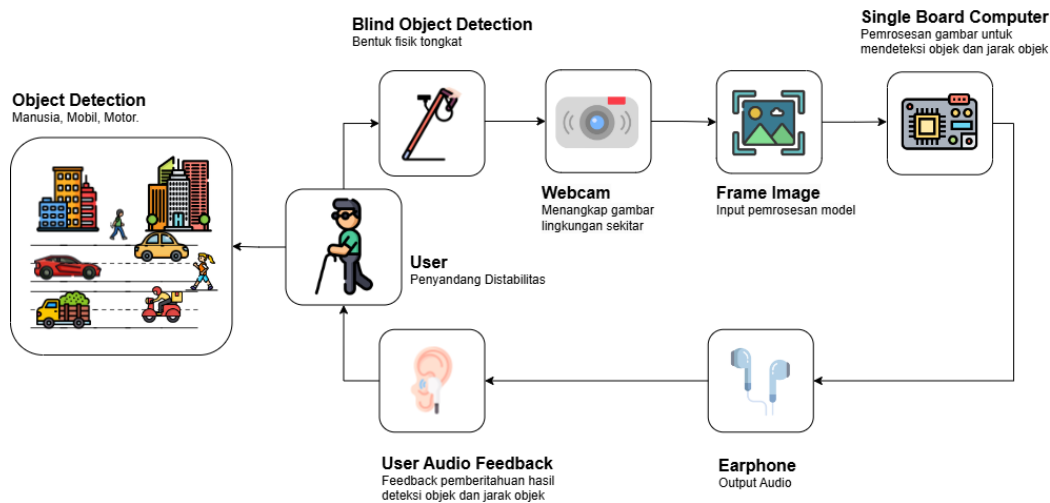
Pada bagian *output*, hasil deteksi yang telah diproses termasuk identifikasi objek dan estimasi jarak, disampaikan kepada pengguna melalui *earphone*, yang mana Sistem akan memutar audio hasil deteksi yang memberikan deskripsi objek yang terdeteksi serta jaraknya dalam bentuk audio.



Gambar 3. 5 Block Diagram

3.2.3. Arsitektur Diagram

Gambar 3.6. merupakan ilustrasi arsitektur diagram rancang bangun alat pengenalan objek untuk tunanetra menggunakan arsitektur *deep learning*. Proses diawali dengan pengguna tunanetra yang sedang menggunakan perangkat berupa tongkat tunanetra dan output *earphone*. Webcam yang terpasang pada tongkat tunanetra akan untuk menangkap gambar dari lingkungan sekitar, dan berfungsi sebagai actuator visual yang memberikan input gambar untuk sistem. Kemudian frame gambar yang diambil oleh webcam diproses sebagai input untuk model deteksi objek. Gambar yang diambil oleh webcam kemudian diproses menggunakan model deteksi objek, dengan arsitektur *deep learning* YOLO atau SSD yang telah dibangun. Pemrosesan deteksi objek tersebut terjadi pada perangkat komputasi seperti Jetson Nano. Setelah objek terdeteksi, Hasil deteksi ini dikonversi kedalam bentuk audio. *Output* audio ini mencakup hasil deteksi objek dan jarak objek dari pengguna.



Gambar 3. 6 Arsitektur Diagram

3.2.4. Hardware System Diagram

Gambar 3.7. merupakan diagram yang menampilkan komponen-komponen perangkat keras utama serta bagaimana komponen-komponen tersebut terhubung satu sama lain dalam sistem. Komponen yang ditampilkan meliputi Jetson Nano, webcam, *earphone*, dan accu 6V/4.5A, serta penjelasan fungsionalitas masing-masing komponen dalam mendeteksi objek di sekitar pengguna. Penempatan perangkat keras atau hardware mulai dari kamera, Jetson Nano, *earphone*, dan accu dirancang sebaik mungkin agar mendapatkan respon sistem yang nyaman, efisien, dan akurat.

Webcam diletakan pada box komponen tongkat yang dipegang oleh pengguna untuk mendapatkan pandangan yang jelas dari area depan box komponen. Penempatan webcam tersebut memberikan stabilitas dan sudut pandang yang optimal untuk mendeteksi objek di jalur yang akan dilalui pengguna. Dengan demikian, kamera dapat menangkap gambar secara efektif tanpa banyak gangguan.

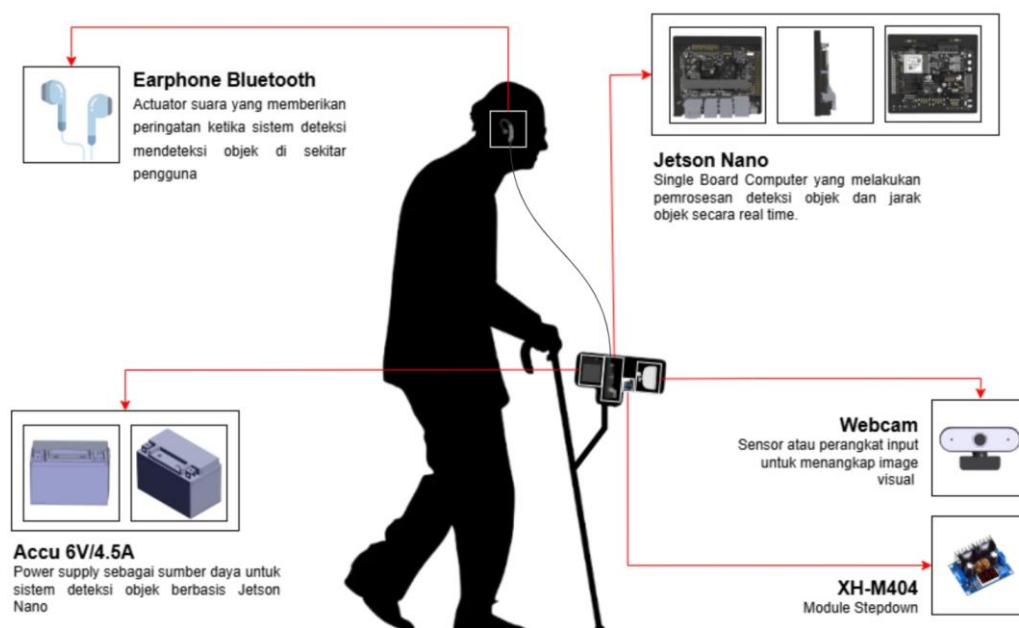
Jetson Nano sebagai perangkat komputas *Single Board Computer* yang melakukan pemrosesan deteksi objek dan pengukuran jarak objek secara real-time yang ditempatkan didalam box komponen. Jetson Nano tersebut diletakan di dekat webcam dan *power supply*. Penempatan Jetson Nano dekat dengan webcam mengurangi panjang kabel yang diperlukan untuk menghubungkan kedua perangkat dan meningkatkan efisiensi pemrosesan gambar. Selain itu, menempatkan Jetson Nano pada kotak yang

terhubung pada tongkat membuatnya lebih portabel dan memudahkan pengguna untuk membawa seluruh sistem dengan nyaman.

Earphone sebagai actuator output suara yang memberikan peringatan kepada pengguna ketika ada objek di sekitar mereka. Terhubung secara nirkabel dengan Jetson Nano untuk memberikan *feedback* audio langsung kepada pengguna.

Kemudian Accu 6V/4.5A sebagai sumber daya untuk seluruh sistem, ditempatkan pada box komponen dibagian belakang, sehingga memastikan bahwa sumber daya selalu tersedia dekat dengan perangkat yang membutuhkan. Hal ini ditujukan guna mengurangi masalah kabel yang berserakan dan memudahkan manajemen daya seluruh sistem. Selain itu, peletakan accu diletakan berdekatan dengan Jetson Nano dan webcam, hal tersebut ditujukan untuk meminimalisir kehilangan daya atau gangguan listrik.

Module XH-M404 digunakan sebagai modul *stepdown* yang ditempatkan di antara accu dan Jetson Nano. Fungsi utama modul ini adalah sebagai perantara yang menurunkan tegangan dari accu sebelum disuplai ke Jetson Nano.

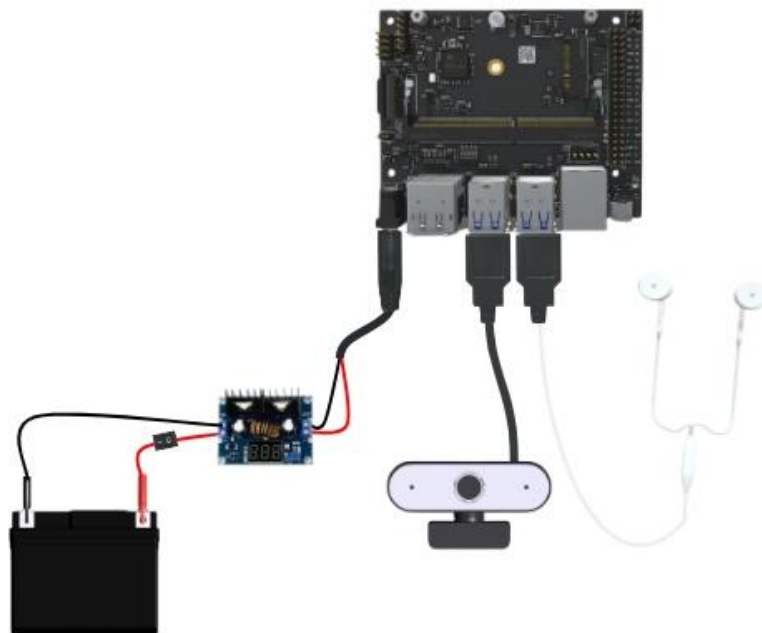


Gambar 3. 7 Hardware System Diagram

3.2.5. Wiring Diagram Component

Gambar 3.8. merupakan ilustrasi *wiring hardware compoenent* diagram yang menunjukkan bagaimana berbagai komponen perangkat keras saling terhubung dan bekerja bersama untuk menyediakan sistem deteksi objek real-time yang membantu pengguna tunanetra. Jetson Nano bertindak sebagai pusat pemrosesan, webcam sebagai *input* visual, dan *earphone* Bluetooth sebagai output audio, dengan daya yang disuplai oleh accu 6V/4.5A, dan Module XH-M404 sebagai module untuk menurunkan tegangan.

Suplai daya yang berasal dari accu terhubung ke module XH-M404 sebelum ke Jetson Nano dengan Barrel jack cable terhubung pada port jack DC pada Jetson Nano, yang merupakan konektor standar untuk suplai daya pada perangkat ini. Webcam terhubung kepada Jetson Nano melalui port USB, sehingga memungkinkan transfer data gambar dari webcam ke Jetson Nano untuk diproses secara real-time. *Earphone* sebagai *actuator* penerima umpan balik suara kepada pengguna berdasarkan data yang diproses oleh Jetson Nano terkoneksi melalui port USB.



Gambar 3. 8 *Wiring Diagram Component*

3.2.6. Design 3D

Gambar 3.9. merupakan ilustrasi design 3D tongkat tunanetra yang dibuat menggunakan aplikasi Blender. Desain ini menunjukkan tongkat dengan beberapa sudut pandang, memberikan gambaran lengkap mengenai bentuk dan fungsi dari tongkat tersebut. Gambar tersebut menampilkan sudut pandang tampak samping kiri, tampak depan, dan tampak samping kanan.



Gambar 3. 9 3D *Design*