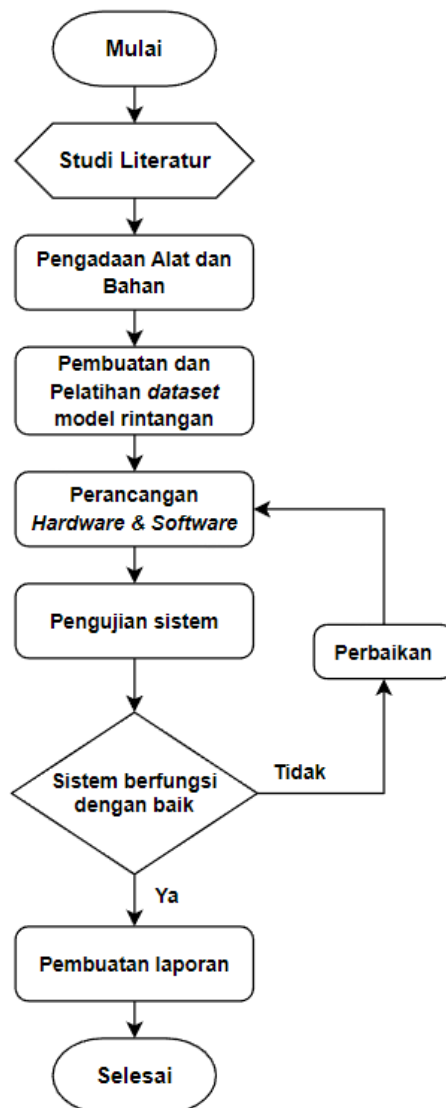


BAB III METODOLOGI PENELITIAN

3.1 Prosedur Penelitian

Dalam melaksanakan penelitian, penulis mengikuti serangkaian prosedur untuk memastikan pencapaian tujuan penelitian. Tahapannya meliputi, studi literatur, pengadaan alat dan bahan, pembuatan dan pelatihan *dataset* model rintangan, perancangan *hardware & software*, pengujian sistem, dan pembuatan laporan. Berikut ini *flowchart* rincian prosedur penelitian yang dilakukan, meliputi:



Gambar 3.1 *Flowchart* Prosedur Penelitian

Flowchart tersebut merupakan gambaran dalam menjalankan penelitian, berikut ini penjelasan rinci untuk setiap tahapannya:

1) Studi Literatur:

Bertujuan untuk memahami dasar teoritis dan praktis mengenai sistem deteksi rintangan bagi penyandang disabilitas netra. Proses ini melibatkan pencarian dan analisis sumber, seperti jurnal ilmiah, konferensi, ataupun buku untuk mendapatkan wawasan tentang topik penelitian.

2) Pengadaan Alat dan Bahan:

Tahap ini mencakup pemilihan dan pengadaan perangkat keras dan perangkat lunak yang diperlukan untuk pembangunan prototipe. Perangkat keras termasuk komponen seperti Raspberry Pi, *night vision camera*, dan perangkat audio untuk output suara. Sementara itu, perangkat lunak mencakup persiapan program pra pelatihan, program pelatihan, dan program pengujian untuk mengembangkan model deteksi objek.

3) Pembuatan dan Pelatihan *Dataset* Model Rintangan:

Proses pembuatan *dataset* dimulai dengan pengambilan gambar sebanyak 2708 gambar yang merepresentasikan 4 jenis rintangan utama yang sering dihadapi penyandang disabilitas netra. Setiap gambar kemudian dilabeli dengan tepat sesuai dengan jenis rintangannya menggunakan perangkat lunak labeliimg. Setelah *dataset* lengkap dibuat, tahap selanjutnya adalah pelatihan model menggunakan TensorFlow Lite di Google Colab. Pelatihan ini ditujukan untuk menghasilkan model yang efisien dalam mengenali dan mengklasifikasi rintangan dengan tingkat akurasi yang tinggi, yang akan memberikan informasi vital kepada pengguna tentang kondisi sekitar mereka.

4) Perancangan *Hardware & Software*:

Desain sistem melibatkan integrasi antara perangkat keras dan perangkat lunak. Ini mencakup perakitan tongkat, perancangan sistem input dan output, serta integrasi model deteksi objek yang telah dilatih ke dalam perangkat keras yang akan digunakan.

5) Pengujian Sistem:

Setelah prototipe sistem dikembangkan, dilakukan serangkaian pengujian untuk mengevaluasi kinerja, akurasi, dan keandalan sistem dalam mendeteksi

rintangan. Pengujian ini meliputi simulasi berbagai skenario yang mungkin dihadapi pengguna dalam kehidupan sehari-hari, serta evaluasi dalam kondisi pencahayaan yang berbeda untuk menguji efektivitas *night vision camera*.

6) Perbaikan:

Analisis terhadap masalah yang diidentifikasi selama pengujian dan penentuan langkah-langkah perbaikan yang diperlukan untuk meningkatkan kinerja dan keandalan sistem. Perbaikan mencakup perbaikan kode program, penyesuaian parameter, dan modifikasi perangkat keras.

7) Pembuatan Laporan:

Pada tahap akhir, dilakukan penyusunan laporan berdasarkan data hasil penelitian yang telah dikumpulkan. Penyusunan laporan juga mengikuti Pedoman Penulisan Karya Ilmiah UPI Tahun 2021, baik dalam hal format maupun gaya penulisan. Laporan ini disusun setelah melalui seluruh proses yang tercantum dalam diagram alir, mulai dari tahap studi literatur hingga tahap perbaikan. Laporan tersebut dimaksudkan untuk menggambarkan secara lengkap dan rinci keseluruhan proses serta hasil penelitian yang telah dilakukan.

3.2 Instrumen Penelitian

Instrumen penelitian yang digunakan dalam pengembangan prototipe ini adalah tongkat tunanetra yang dilengkapi dengan raspberry pi 4b (RAM 4GB), kamera *night vision*, baterai, dan TWS. Instrumen ini berguna untuk mendeteksi rintangan dan memberikan feedback audio kepada pengguna. Adapun perangkat lunak yang dipakai yaitu Thony untuk menjalankan program python, google colab untuk melatih model, dan labelimg untuk memberikan anotasi pada setiap gambar.

3.3 Analisis Data

Analisis data dalam penelitian ini meliputi evaluasi kestabilan *dataset* terhadap kinerja sistem pendeteksian rintangan pada prototipe yang dikembangkan untuk penyandang disabilitas netra. Selain itu, akan dilakukan perbandingan evaluasi berbagai model deteksi objek TensorFlow Lite untuk mengidentifikasi model yang paling ringan, akurat, dan responsif dalam mendeteksi hambatan serta objek-objek penting bagi penyandang disabilitas netra.

3.4 Pengembangan Prototipe

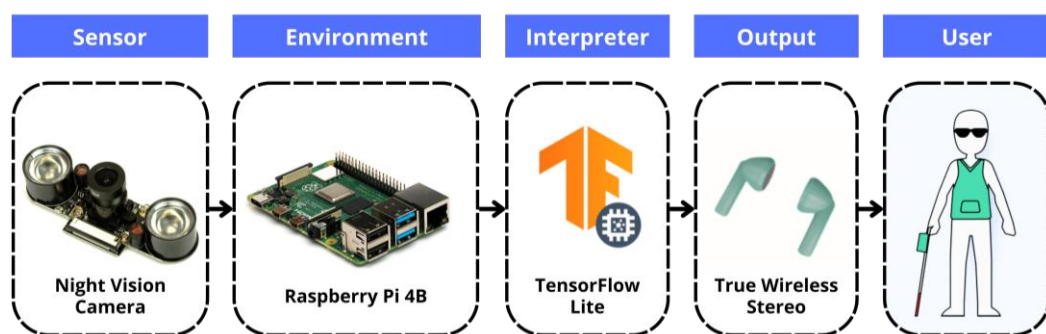
Pada tahap ini, fokus utama adalah mengimplementasikan secara praktis konsep sistem deteksi rintangan bagi penyandang disabilitas netra. Hal ini melibatkan pemilihan komponen perangkat keras yang sesuai, perancangan arsitektur sistem, dan pengukuran kinerja.

3.4.1 Analisis Kebutuhan

Sebelum merancang dan mengembangkan prototipe, analisis kebutuhan dilakukan untuk memahami apa yang dibutuhkan oleh penyandang disabilitas netra dalam keseharian mereka, terutama terkait dengan deteksi rintangan. Analisis ini melibatkan studi literatur di bidang teknologi assistive. Tujuan dari analisis kebutuhan adalah untuk mengidentifikasi jenis rintangan yang sering dihadapi, preferensi terhadap jenis feedback (audio, haptik, atau kombinasi keduanya), dan kriteria desain ergonomis untuk perangkat yang akan digunakan. Hasil dari analisis kebutuhan ini akan membentuk dasar untuk spesifikasi fungsi dan desain sistem

3.4.2 Arsitektur Sistem

Berdasarkan kebutuhan yang telah diidentifikasi, arsitektur sistem dibangun dengan komponen-komponen perangkat keras dan perangkat lunak yang terintegrasi sehingga menciptakan sebuah sistem yang efektif dan responsif. Desain prototipe memperhatikan aspek ergonomis dan portabilitas untuk memastikan kenyamanan pengguna. Berikut ini arsitektur prototipe sistem deteksi rintangan bagi penyandang disabilitas netra:



Gambar 3.2 Arsitektur Sistem yang Diusulkan

Pada tahap ini, setiap komponen dalam arsitektur sistem dipertimbangkan dengan cermat. Berikut adalah penjelasan untuk setiap bagian arsitektur:

1) *Sensor: Night Vision Camera*

Krisna Dwi Nurikhsani, 2024

PENGEMBANGAN PROTOTIPE SISTEM DETEKSI RINTANGAN BAGI PENYANDANG DISABILITAS NETRA

Universitas Pendidikan Indonesia | repository.upi.edu | perspustakaan.upi.edu

Sensor ini digunakan untuk mendeteksi rintangan atau halangan di sekitar pengguna, terutama dalam kondisi pencahayaan rendah atau gelap. Dengan kemampuan penglihatan malam, sensor ini memungkinkan pengguna untuk mendapatkan informasi visual yang diperlukan untuk navigasi yang aman.

2) *Environment*: Raspberry Pi 4B

Sebagai sebuah mikrokomputer, Raspberry Pi 4B menyediakan platform yang kokoh untuk menjalankan berbagai proses komputasi yang diperlukan dalam sistem, termasuk pengolahan data dari sensor, analisis gambar menggunakan TensorFlow Lite, dan pengaturan output suara ke True Wireless Stereo.

3) *Interpreter*: TensorFlow Lite

Platform ini digunakan sebagai interpreter untuk menganalisis data yang diperoleh dari sensor Night Vision Camera. Dengan menggunakan model deteksi objek yang telah dilatih, TensorFlow Lite mampu mengenali objek yang terdeteksi di sekitar pengguna.

4) *Output*: True Wireless Stereo

Berbagai informasi tentang lingkungan sekitar, akan disampaikan kepada pengguna melalui suara yang dihasilkan oleh TWS. Ini memungkinkan pengguna dengan disabilitas netra untuk menerima informasi dengan cepat dan efisien.

5) *User*: Disabilitas Netra

Pengguna sistem adalah individu dengan disabilitas netra yang menjadi fokus utama pengembangan. Desain prototipe harus mempertimbangkan kebutuhan dan keterbatasan pengguna ini dalam interaksi dengan sistem, seperti memastikan bahwa informasi yang disampaikan oleh sistem dapat dipahami dengan jelas oleh pengguna.

3.4.3 Pengukuran Kinerja

Setelah prototipe dikembangkan, pengukuran kinerja sistem dilakukan untuk mengevaluasi efektivitasnya dalam mendeteksi rintangan. Beberapa parameter yang diukur dalam tahap ini meliputi:

1) Kualitas *Dataset*

Untuk menjamin efektivitas model deteksi objek, pengujian kualitas *dataset* menjadi langkah esensial, mencakup berbagai aspek vital seperti kestabilan masing-masing kelas, penilaian anotasi objek rintangan pada setiap gambar, analisis ukuran gambar rata-rata, dan rasio median gambar diperlukan untuk memahami distribusi ukuran objek dalam *dataset*, memungkinkan penyesuaian pada model untuk mengoptimalkan deteksi objek dengan berbagai proporsi dan mengurangi kebutuhan sumber daya komputasi.

2) Kecepatan Inferensi

Kecepatan inferensi dipengaruhi oleh berbagai faktor, termasuk kompleksitas model, jumlah dan ukuran lapisan dan node dalam neural network, serta optimisasi yang telah diterapkan pada model. Kecepatan inferensi diukur dilakukan dengan alat tolak ukur kinerja selama proses inferensi TensorFlow Lite dari Google (TensorFlow, 2022). Pengukuran ini dilakukan dengan menghitung rata-rata waktu yang diperlukan oleh model untuk memproses setiap gambar dari *dataset* uji atau kita dapat menyebutnya dengan latensi. Waktu tersebut dihitung dari saat gambar diberikan sebagai input hingga output (prediksi rintangan) dihasilkan oleh model.

3) Akurasi

Metrik untuk mengukur akurasi model adalah mAP (*mean average precision*), yang menilai kemampuan model dalam mengidentifikasi lokasi dan kategori objek dalam gambar dengan tepat. mAP adalah rata-rata dari *Average Precision* (AP) untuk setiap kelas objek, yang menggabungkan presisi (proporsi prediksi positif yang benar) dan recall (proporsi positif aktual yang teridentifikasi) pada berbagai ambang batas deteksi. Skor mAP yang lebih tinggi menunjukkan bahwa model lebih akurat dalam mendeteksi dan mengklasifikasikan objek dalam gambar. Untuk pengujian, model menjalankan inferensi pada 175 gambar uji, dengan hasilnya dibandingkan terhadap data kebenaran dasar, untuk menghitung skor mAP penulis menggunakan kalkulator mAP sumber terbuka dari github (Cartucho dkk., 2018).

4) Penggunaan Memori

Pengukuran penggunaan memori dilakukan dengan memantau jumlah memori yang diperlukan oleh model selama proses inferensi. Ini mencakup

memori yang digunakan untuk menyimpan parameter model, data input, dan data *intermediate* yang dihasilkan. Penggunaan memori yang efisien memungkinkan sistem untuk berjalan pada perangkat dengan sumber daya terbatas, seperti raspberry pi tanpa mengurangi kinerja sistem secara keseluruhan. Pengoptimalan memori dapat dicapai melalui teknik seperti quantization, yang mengurangi presisi data tanpa menurunkan akurasi model secara signifikan.