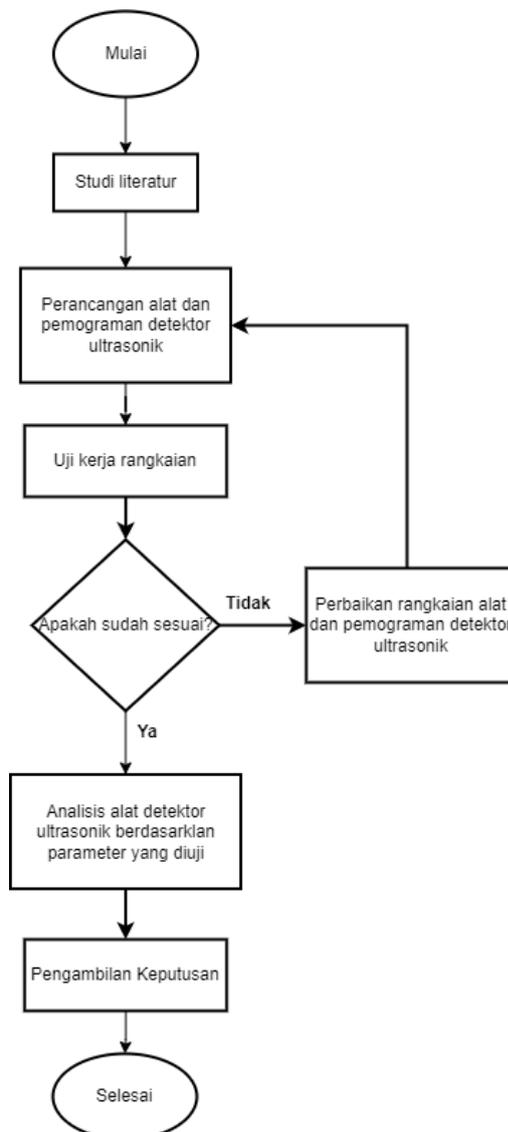


BAB III METODELOGI PENELITIAN

3.1 Metode Penelitian

Metode penelitian ini menggunakan metode eksperimen terencana. Berdasarkan referensi desain dan fungsi alat yang sudah ada, dilakukan pengembangan untuk menciptakan sistem pendeteksi rintangan dengan output suara yang spesifik. Fokus penelitian ini adalah mengembangkan alat yang sesuai dengan target dan tujuan penelitian. Pada Gambar 3.1 dijelaskan diagram alur penelitian.



Gambar 3.1 Diagram alur penelitian

Pada tahap pertama, melakukan tinjauan pustaka guna memperoleh landasan teoritis terkait dengan penelitian yang relevan, serta mempelajari studi-studi yang mendukung teori tersebut agar data yang terkait dapat mengatasi situasi dalam pengembangan sistem sesuai dengan spesifikasi teknis yang diperlukan. Beberapa informasi yang dibutuhkan meliputi pengetahuan tentang teknologi asistif seperti teori *Linear Predictive Coding*, penggunaan mikrokontroler Arduino, Sensor ultrasonik JSN-SR04T, Jack audio TRRS, dan beberapa teori terkait bahasa pemrograman yang digunakan dalam Arduino IDE.

Kemudian melakukan pengembangan alat yang telah direncanakan sebelumnya. Jika terjadi kesalahan pada sirkuit fisik atau dalam script program, proses perancangan perangkaian akan dilakukan perakitan kembali dan dilanjutkan setelah memperbaiki sumber kesalahan sebelumnya. Setelah berhasil merangkai alat, alat tersebut akan diuji sesuai dengan parameter yang telah ditentukan sebelumnya. Pengujian ini akan memeriksa bagaimana jarak objek yang terdeteksi berpengaruh terhadap tingkat kesalahan (*error*) pada alat detektor.

Pada tahap akhir perancangan, alat akan diuji coba untuk memastikan bahwa semua komponen berfungsi dengan baik. Data yang dihasilkan dari pengujian ini akan menjadi acuan bagi penyandang tunanetra saat berjogging di lintasan. Alat akan memberikan peringatan jika terdeteksi adanya objek di depannya, sehingga penyandang tunanetra dapat menghindarinya. Alat tersebut akan menjadi panduan yang berguna bagi mereka dalam menjaga keamanan dan menghindari potensi bahaya selama beraktivitas jogging. Sistem pendeteksian sensor yang telah dikembangkan akan menjalani rangkaian pengujian untuk mengevaluasi kinerjanya.

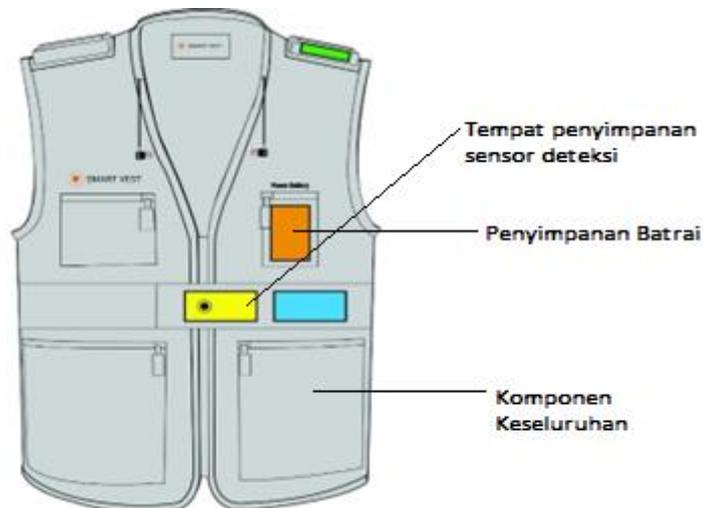
3.2 Perangkat dan Penunjang

Sistem pendeteksi rintangan dengan output suara untuk tunanetra dirancang menggunakan beberapa komponen dan perangkat. Mikrokontroler yang digunakan adalah Arduino UNO, sedangkan sensor jarak yang digunakan adalah JSN-SR04T. Selain itu, terdapat juga komponen TRRS yang berfungsi sebagai penghubung antara sensor dan *earphone* yang digunakan sebagai output suara.

Selain komponen tersebut, dalam penelitian ini juga digunakan perangkat dan sistem tambahan. Salah satunya adalah *library Talkie* yang digunakan untuk menghasilkan suara output. Selain itu, terdapat juga *source vocab* yang dirancang menggunakan bahasa pemrograman dalam Arduino IDE, yang digunakan untuk mengatur kata-kata dan frasa yang akan diucapkan oleh sistem.

Dengan menggunakan kombinasi komponen dan perangkat tersebut, sistem pendeteksi rintangan dapat mengukur jarak dengan menggunakan sensor ultrasonik, dan menghasilkan suara melalui *earphone* berdasarkan hasil pendeteksian.

Semua komponen tersebut di letakkan dalam sebuah rompi yang dinamakan *Smart Vest*, Rompi ini berfungsi sebagai media dari penyimpanan komponen sensor. Dengan media rompi ini diharapkan memberikan kemudahan dan kenyamanan bagi pengguna tunanetra dalam mengenakan *prototype* sensor jarak pendeteksi rintangan. Desain rompi ini juga harus mempertimbangkan faktor-faktor seperti optimalisasi penyimpanan sensor, serta kemudahan dalam pemeliharaan dan pengisian daya. Berikut desain rompi yang akan digunakan pada Gambar 3.2

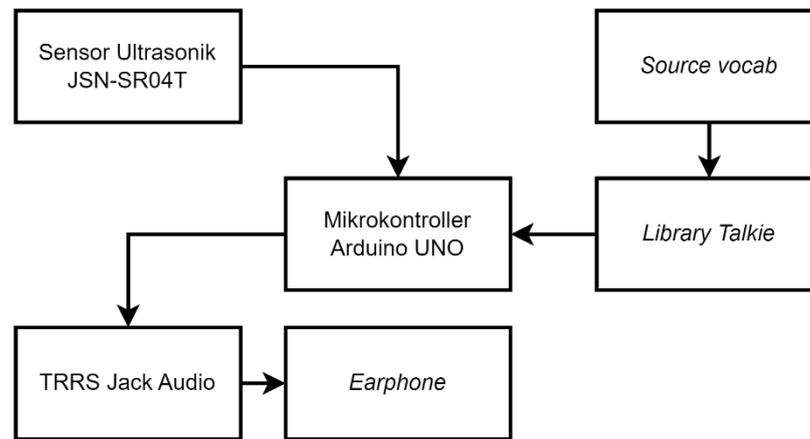


Gambar 3.2 Desain Rompi *Smart Vest*

3.3 Prinsip kerja

Sistem secara keseluruhan beroperasi dengan prinsip kerja untuk mendeteksi objek. Sensor ultrasonik JSN-SR04T bertugas mendeteksi adanya rintangan di depannya dan menghasilkan sinyal deteksi. Sinyal tersebut kemudian dikirim kepada

mikrokontroler Arduino Uno untuk diproses lebih lanjut. Prinsip kerja perangkat dan sistem yang dirancang dapat diilustrasikan melalui diagram Gambar 3.3.



Gambar 3.3 Prinsip kerja

Dalam mikrokontroler, sinyal deteksi yang awalnya berupa angka akan mengalami pemrosesan. *library Talkie* digunakan untuk mengubah sinyal angka menjadi suara. *library* ini memanfaatkan sampel suara yang telah disiapkan sebelumnya dari *source vocab*. Sampel suara tersebut telah dikonfigurasi agar sesuai dengan angka-angka tertentu yang akan dihasilkan sebagai suara output.

Dengan adanya penggunaan *library Talkie* dan *source vocab*, mikrokontroler Arduino Uno memiliki kemampuan untuk memproses sinyal angka menjadi suara yang dapat didengar melalui *earphone* sebagai output. Dengan demikian, informasi tentang adanya rintangan akan disampaikan kepada pengguna melalui suara yang dihasilkan oleh alat tersebut.

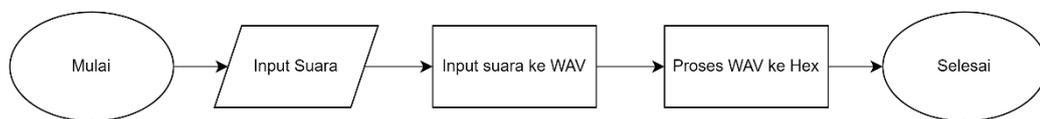
3.4 Algoritma

Dalam merancang sensor pendeteksi jarak dengan output suara, terdapat beberapa algoritma yang digunakan untuk mendukung operasi sistem tersebut, algoritma yang digunakan untuk mendukung jalannya sistem yaitu algoritma pembuatan *library* dan algoritma sensor pendeteksi.

3.4.1 Algoritma *source vocab*

Proses pembuatan *source vocab* dapat dilihat pada Gambar 3.4, agar hasil deteksi sensor mengeluarkan output suara ke dalam earphone. Berikut langkah-langkahnya:

1. Pertama-tama, lakukan perekaman suara yang ingin dimasukkan ke dalam *source vocab*. Setelah suara direkam, impor file suara ke dalam aplikasi Pywizard.
2. Selanjutnya, Pywizard mengubah file suara ke format WAV. Format WAV adalah format audio yang umum digunakan dan kompatibel dengan sebagian besar sistem deteksi suara.
3. Kemudian, gunakan Pywizard, perangkat lunak pengonversi suara ke bahasa hex. Pywizard adalah alat yang dapat digunakan untuk mengonversi file WAV menjadi format hex yang dapat dimanfaatkan dalam sistem deteksi suara.
4. Dalam Pywizard, impor file WAV yang sudah diubah ke format WAV sebelumnya. Terakhir, jalankan proses konversi untuk mengubah suara menjadi bahasa hex. Pywizard akan melakukan analisis suara dan menghasilkan output dalam format hex yang dapat digunakan dalam sistem deteksi suara.



Gambar 3.4 Algoritma *source vocab*

3.4.2 Algoritma sensor pendeteksi

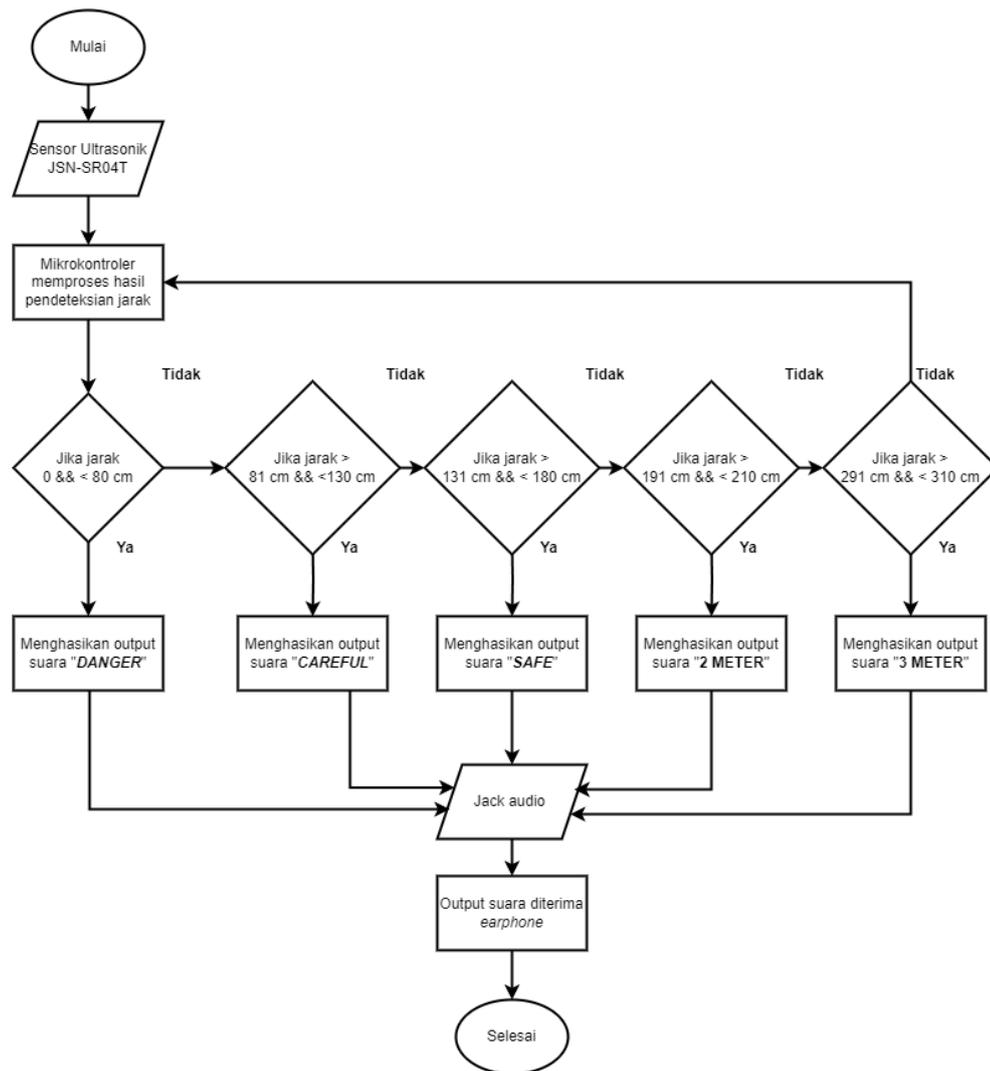
Pada Gambar 3.5, pemrograman untuk mikrokontroler menggunakan bahasa program "C+" dan dilakukan melalui *software* Arduino IDE. Tujuan dari pemrograman ini adalah menghasilkan output sesuai dengan yang dijelaskan berikut:

- a. Jarak 0 sampai 80 cm, output suara yang dihasilkan adalah “*DANGER*”
- b. Jarak 81 sampai 130 cm, output suara yang dihasilkan adalah “*CAREFUL*”
- c. Jarak 131 sampai 180 cm, output suara yang dihasilkan adalah “*SAFE*”
- d. Jarak 191 sampai 210 cm, output suara yang dihasilkan adalah “2 METER”
- e. Jarak 291 sampai 310 cm, output suara yang dihasilkan adalah “3 METER”

Gilang Ramadhan, 2023

RANCANG BANGUN SENSOR JARAK PENDETEKSI RINTANGAN DENGAN OUTPUT SUARA UNTUK PENYANDANG TUNANETRA

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



Gambar 3.5 Algoritma sensor pendeteksi

3.5 Teknik Analisis Data

3.5.1 Menghitung persentase besar *error* detektor

Pendeteksian dilakukan berdasarkan faktor-faktor yang mempengaruhi penggunaan alat detektor ketika sudah dirancang. Salah satu faktor yang harus diperhatikan adalah akurasi jarak objek yang dihasilkan oleh alat tersebut dibandingkan dengan jarak objek sebenarnya.

Proses deteksi dilakukan secara berulang sebanyak lima kali pada jarak yang sama untuk memastikan konsistensi nilai yang dihasilkan oleh alat tersebut. Dari percobaan tersebut, akan diperoleh rata-rata nilai jarak yang dihasilkan oleh alat

dan juga *margin error* yang terkait. Jarak yang diuji akan ditingkatkan sesuai dengan kebutuhan sampel pengujian.

Persentase besar *error* detektor e dihitung berdasarkan rata-rata jarak hasil deteksi alat $\hat{\delta}$ yang dikurangi dengan jarak sebenarnya s . Hasil dari operasi tersebut dibagi kembali dengan s lalu dikali 100%. Dalam bentuk matematis, persamaan untuk skema tersebut dapat dituliskan sebagai berikut: (Astuti, 2019).

$$e = \frac{\hat{\delta} - s}{s} \times 100\% \quad (3.1)$$

3.5.2 Menghitung jarak detektor

Sistem kerja sensor ultrasonik yang mengirimkan gelombang suara dan menerima pantulan dari gelombang tersebut. Jarak yang dihasilkan dari proses ini akan menjadi dua kali lebih besar dari jarak objek yang sebenarnya. Oleh karena itu, diperlukan penyesuaian pada jarak akhir objek d sebagai berikut: (Frima Yudha & Sani, 2019).

$$d = \frac{s}{2} = \frac{v \times t}{2} \quad (3.2)$$

Nilai d merupakan jarak yang ingin diukur menggunakan sensor ultrasonik. Jarak ini diukur dalam satuan yang sesuai, seperti meter atau centimeter. Nilai s merupakan waktu yang dibutuhkan gelombang ultrasonik untuk melakukan perjalanan dari sensor ke objek yang ingin diukur jaraknya, lalu kembali ke sensor. Nilai v merupakan kecepatan gelombang ultrasonik. Gelombang ultrasonik biasanya memiliki kecepatan yang tetap dalam medium tertentu, seperti udara yaitu 340 m/s, dengan mengasumsikan suhunya $\pm 20^\circ$ celcius sesuai dengan suhu di dalam ruangan.. Kecepatan ini diukur dalam satuan meter per detik. Terakhir t merupakan waktu tempuh atau waktu yang dibutuhkan gelombang ultrasonik untuk melakukan perjalanan dari sensor ke objek dan kembali lagi. Waktu ini diukur dalam satuan detik.

3.6 Perancangan Metode LPC

Linear Predictive Coding (LPC) adalah metode pemrosesan sinyal yang digunakan dalam analisis dan sintesis suara. Metode ini mengambil sampel suara digital dan menggunakan model prediksi linier untuk mengestimasi parameter vokal dan filter

resonansi yang terkandung dalam suara (Rabiner & Juang, 1993). Dengan pendekatan prediksi linier, LPC dapat mengurangi dimensi data suara yang diolah dan menghasilkan representasi yang lebih efisien.

Ada beberapa cara untuk melakukan konversi dari suara ke LPC, dalam penelitian ini menggunakan aplikasi python sebagai perantara untuk mengkonversi yaitu melakukan install aplikasi Python dari websitenya, dan juga install Pywizard sebagai recording suara yang ingin diubah. Tahap selanjutnya melakukan konfigurasi *Environment Variables* yang berguna untuk memudahkan dalam pengkodean nanti dan meminimalisir *error*.