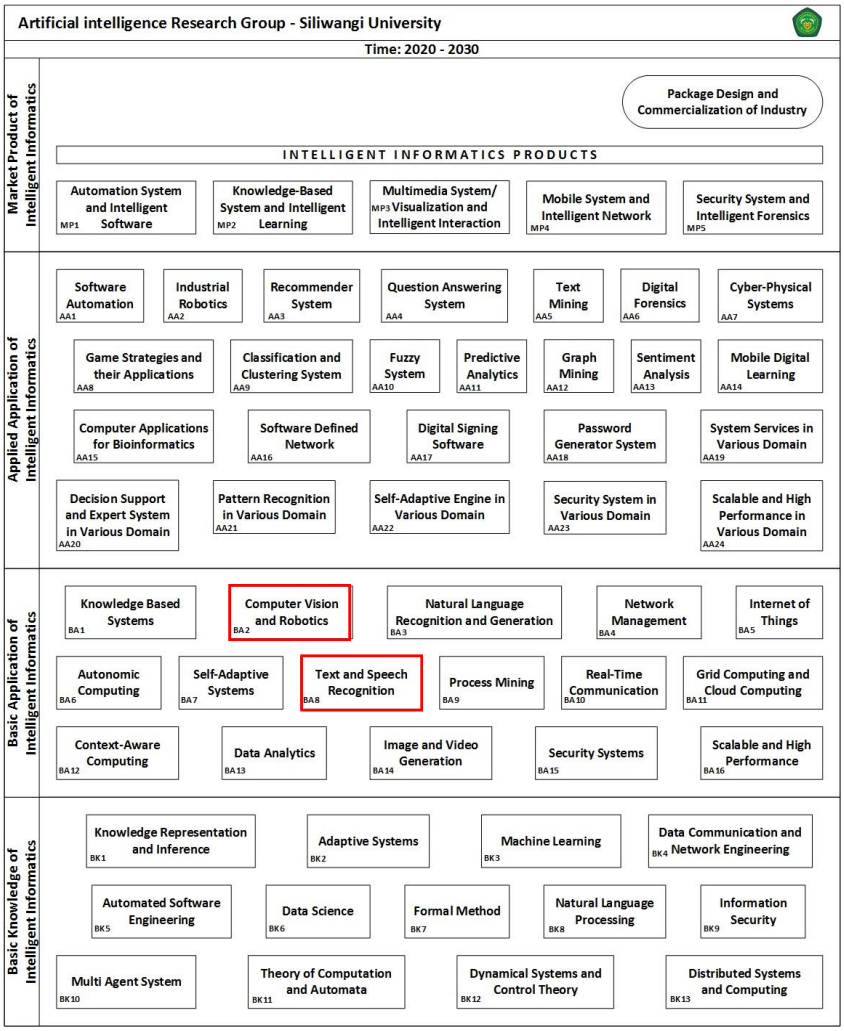
BAB II

METODOLOGI

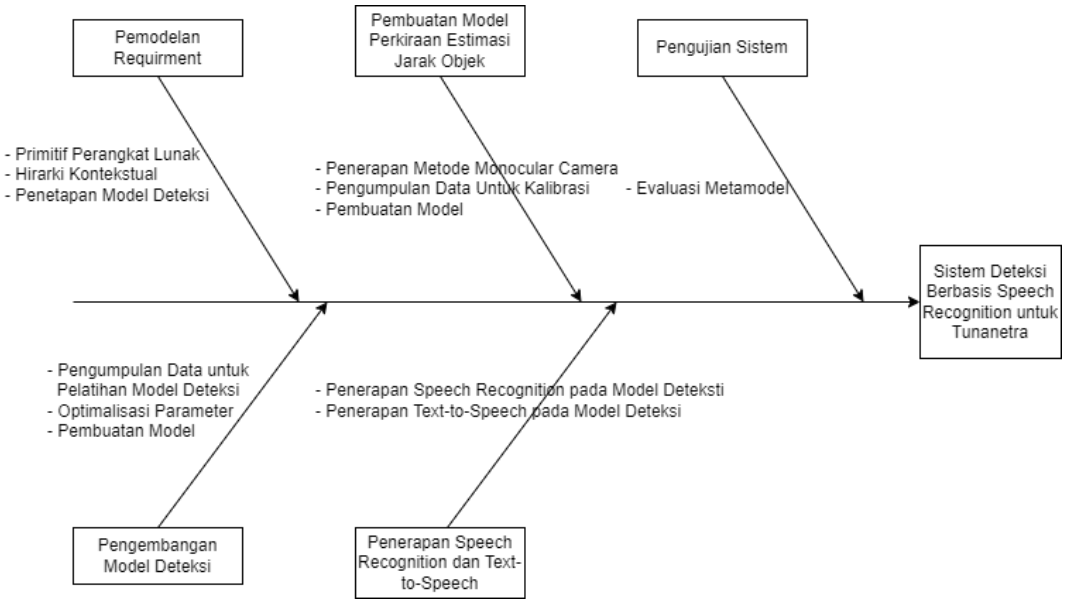
3.1 Peta Jalan *(Roadmap)* Penelitian

Secara umum *roadmap* dari penelitian ini sejalan dengan peta jalan penelitian Universitas Siliwangi pada sub bidang *artificial intelligence*. Topik penelitian yang digunakan pada penelitian ini berkaitan dengan *text and speech recognition* dan *computer vision and robotics* yang berada pada ranah *basic application of intelligent informatics*. *Roadmap* tersebut ditampilkan pada Gambar 3.



Gambar 3.1 *Roadmap* penelitian (AIS Universitas Siliwangi, 2024)

Pemilihan ranah *basic application of intelligent informatics* dimaksudkan agar dapat membuat dasar yang kokoh untuk ranah selanjutnya. Sehingga diharapkan dapat menciptakan penerapan aplikasi yang lebih baik dan berdaya saing tinggi dengan penelitian-penelitian yang berada pada satu keilmuan. Selain itu, penelitian yang dilakukan ini merupakan pengembangan dari penelitian sebelumnya dengan judul "*Pengembangan Sistem Kacamata Cerdas untuk Penderita Tunanetra Berbasis Self-Adaptive Cyber-Physical Systems*" yang dilakukan oleh Aradea dkk (Aradea dkk., 2023). Gambar 3.2 menunjukkan *fishbone diagram* penelitian sebagai turunan dari bidang yang dipilih dari *roadmap* penelitian pada gambar 3.1.



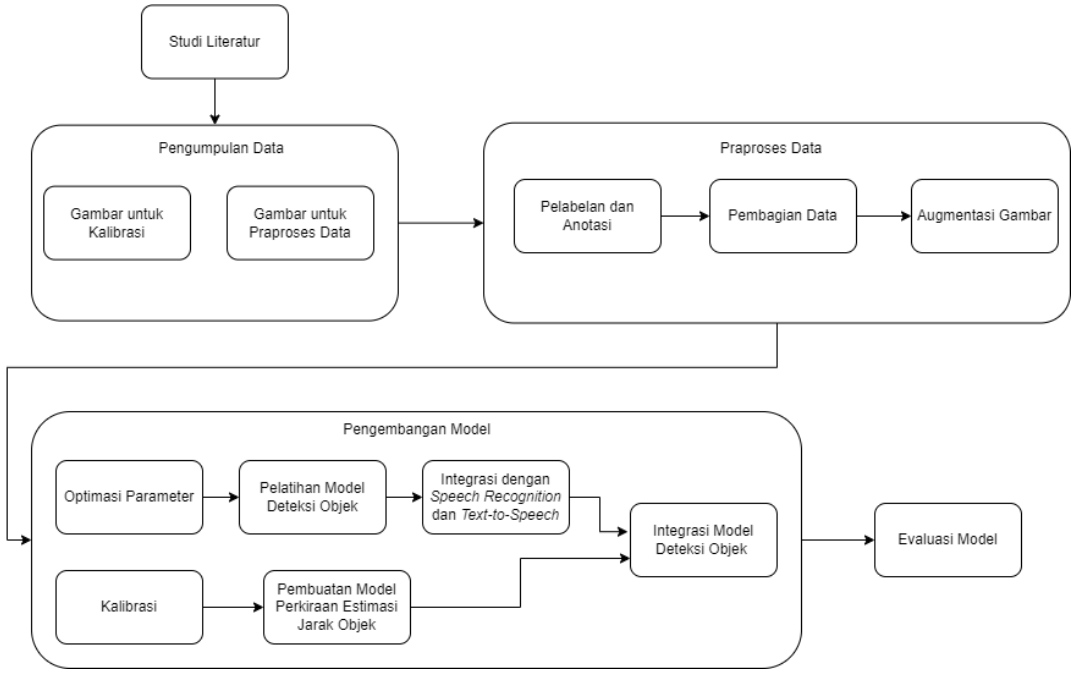
Gambar 3.2 *Fishbone diagram*

Jalannya penelitian ini dimulai dari penetapan-penetapan kebutuhan yang perlu dipersiapkan sebagai landasan untuk setiap tahapan yang nanti dilakukan. Selanjutnya, proses pengembangan model deteksi dilakukan untuk menghasilkan model deteksi yang memiliki performa lebih baik dari penelitian sebelumnya

dengan menerapkan konsep pemilihan parameter secara otomatis dengan menggunakan PSO. Setelah itu, model perkiraan estimasi jarak objek, model *speech recognition*, dan model TTS dibuat serta diterapkan pada model deteksi untuk mendukung fungsionalitas lebih lanjut dari penelitian yang dilakukan. Pengujian sistem tentunya akan dilakukan untuk mengevaluasi hasil model-model yang telah dibuat dan diintegrasikan sehingga dari hasil proses tersebut tercipta sebuah sistem deteksi objek berbasis *speech recognition* dengan performa dan fungsionalitas yang baik.

3.2 Tahapan Penelitian

Target penelitian ini adalah menciptakan suatu model generik dengan kemampuan mendeteksi objek, menghitung perkiraan jarak objek, menerima pesan suara, dan mengeluarkan suara umpan balik. Dengan strategi untuk memperluas arsitektur sistem melalui proses *learning* terhadap seluruh komponen sistem. Sehingga tercipta tahapan penelitian yang mengadopsi pengembangan metode *machine learning/ deep learning* melalui tahapan-tahapan yang dikembangkan oleh (John dkk., 2020). Maka dari itu, rangkaian tahapan penelitian yang ada dilakukan akan seperti pada Gambar 3.3.



Gambar 3.3 Tahapan penelitian

3.2.1 Studi Literatur

Tahap ini pertama kali dilakukan dengan mencari *survey paper* yang berkaitan dengan penelitian yang dilakukan. Pencarian selanjutnya adalah memilih *technical paper* terindeks dan relevan sehingga hasilnya dapat menciptakan lingkungan *state of the art* yang mendukung jalannya penelitian ini.

3.2.2 Pengumpulan Data

Penggunaan data berasal dari pengambilan gambar yang dilakukan secara mandiri terkait dengan objek-objek yang berada di dalam ruangan. Pengumpulan data ini dibagi untuk dua kebutuhan yang berbeda, yaitu data gambar untuk proses pelatihan model dan data gambar untuk kalibrasi. Selain itu, untuk memperlihatkan keunggulan model yang dibangun, penelitian ini menggunakan *Common Object in*

Context (COCO) *dataset* sebagai *benchmark* perbandingan dengan model yang telah dibuat pada penelitian lain.

3.2.3 Kalibrasi

Kalibrasi dilakukan untuk menentukan parameter atau koefisien yang dibutuhkan dalam proses perhitungan jarak objek. Parameter atau koefisien tersebut berhubungan dengan penentuan hubungan antara titik 3D di dunia nyata dengan proyeksi 2D pada gambar yang diambil oleh kamera. Tujuan utama dari kalibrasi ini adalah untuk mendapatkan nilai matriks intrinsik K dan matriks ekstrinsik berupa matriks rotasi R serta vektor translasi t. Pencarian nilai-nilai tersebut didasarkan pada pengetahuan tentang koordinat objek 3D pada dunia nyata (X_w, Y_w, Z_w) dengan koordinat objek 2D pada kamera (u, v). Hubungan koordinat objek pada bidang 2D dengan objek 3D dapat dicari dengan menggunakan Persamaan (1)-(3) (Sadekar & Mallick, 2020).

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

$$u = \frac{u'}{w'} \quad (2)$$

$$v = \frac{v'}{w'} \quad (3)$$

Dimana, P merupakan matriks proyeksi dengan ukuran 3x4 yang memiliki dua bagian utama, yaitu matriks intrinsik K dengan ukuran 3x3 dan matriks ekstrinsik ([R|t]) dengan R berukuran 3x3 serta t berukuran 3x1. P dirumuskan pada persamaan (4) (Sadekar & Mallick, 2020).

$$P = K \times [R|t] \quad (4)$$

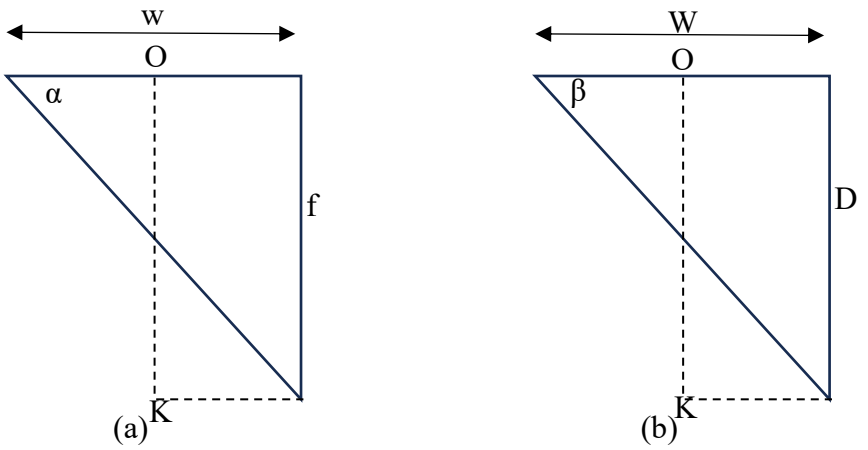
Sehingga untuk mendapatkan matriks K, dapat dilakukan dengan mencari matriks pada Persamaan (5) (Sadekar & Mallick, 2020).

$$K = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$(f_x,\ f_y)$ merupakan nilai *focal length* yang menjadi fokus utama dalam kalibrasi. Karena nilai inilah yang akan digunakan untuk menghitung estimasi jarak objek dengan kamera. *Focal length* biasanya menggunakan satuan pixel. Dalam memudahkan proses kalibrasi, langkah ini dilakukan dengan menggunakan modul OpenCV *python* sehingga menghasilkan kalibrasi kamera yang lebih tepat dan akurat.

3.2.4 Pembuatan Model Perkiraan Jarak Objek

Proses ini dilakukan dengan menggunakan persamaan proporsi segitiga tangen. Hubungan antara objek dengan kamera akan menciptakan segitiga siku-siku yang merepresentasikan konteks jarak dunia nyata dan kamera. Lebih jelasnya dapat dilihat pada Gambar 3.



Gambar 3.4 (a) representasi kamera terhadap objek dalam bentuk segitiga (b) representasi dunia nyata terhadap objek dalam bentuk segitiga

Gambar 3.4 (a) menunjukkan representasi penglihatan kamera terhadap objek dalam bentuk segitiga dengan nilai-nilai yang diketahui, yaitu w dan f. w merupakan nilai lebar objek dalam satuan pixel yang dihitung berdasarkan perspektif kamera. Nilai f menunjukkan nilai *focal length* yang didapatkan dari proses kalibrasi pada tahap sebelumnya. α adalah sudut yang digunakan untuk menentukan persamaan proporsi pada segitiga. Gambar 3.4 (b) menunjukkan hal sama, tetapi dalam representasi dunia nyata dengan W sebagai lebar objek dalam satuan cm dan D adalah jarak objek dengan kamera dalam satuan cm. Sedangkan untuk β merupakan sudut yang digunakan untuk menentukan persamaan proporsi.

Kedua segitiga tersebut memiliki hubungan proporsi karena memiliki ukuran dan bentuk serupa sehingga dapat berlaku persamaan proporsi untuk tangen.

$$\tan \alpha = \tan \beta \tag{6}$$

Dimana $\tan \alpha = \frac{f}{w}$ dan $\tan \beta = \frac{D}{W}$, maka untuk menentukan jarak objek dengan kamera pada dunia nyata dapat dihitung dengan Persamaan (7) (Rosebrock, 2015).

$$\frac{f}{w} = \frac{D}{W}$$

$$D = \frac{fW}{w} \tag{7}$$

3.2.5 Praproses Data

Terdapat tiga tahapan pada proses praproses data, yaitu augmentasi gambar, pelabelan dan anotasi, dan pembagian data.

1) Pelabelan dan anotasi dilakukan dengan memberikan kelas pada setiap data gambar serta membuat koordinat *bounding box* pada objek yang berada pada gambar. Hasil dari pelabelan dan anotasi disimpan pada sebuah *file* yang nanti akan dibaca ketika proses pelatihan sedang berlangsung.
2) Penelitian ini menggunakan pembagian *dataset* dengan rasio 80:10:10, yaitu 80% data untuk proses pelatihan, 10% untuk validasi, dan 10% untuk data uji. Rasio tersebut didasarkan pada rasio *default* yang digunakan pada YOLOv8 (Jocher dkk., 2023).
3) Penggunaan augmentasi gambar dimaksudkan untuk meningkatkan keragaman data tanpa menambah jumlah kelas yang digunakan. Proses ini melibatkan penggunaan rotasi gambar dan perubahan perspektif penampilan gambar dengan kamera di depannya. Hal ini berguna untuk meningkatkan akurasi dan pemahaman model pada.

Ketiga tahapan tersebut dilakukan dengan bantuan *software* roboflow sehingga memudahkan dalam proses praproses data.

3.2.6 Optimasi Parameter

Adam optimizer merupakan salah satu metode optimasi yang banyak digunakan dalam proses pengoptimasian model deteksi objek maupun model lainnya yang menggunakan *neural network*. Namun, yang menjadi masalah adalah bagaimana memilih nilai parameter dengan tepat sehingga menciptakan model yang paling optimal. Salah satu nilai parameter yang paling krusial adalah nilai dari *learning rate*. Tetapi, pemilihan nilai tersebut dilakukan secara manual yang menjadikan model *overfitting* atau memberikan penurunan nilai *loss* yang sangat rendah. Maka

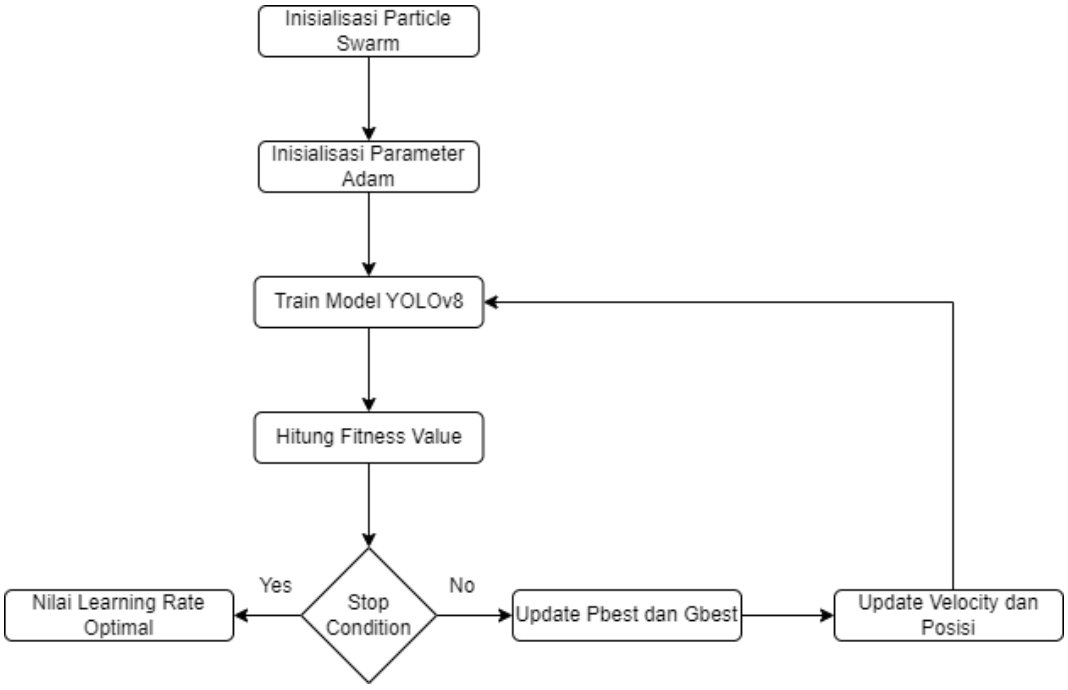
dari itu, digunakan metode *Particle Swarm Optimization* (PSO) untuk memilih nilai *learning rate* paling optimal secara otomatis berdasarkan karakteristik dari model yang akan dibangun.

Penggunaan PSO didasari pada beberapa hal sebagai berikut (Kessentini & Barchiesi, 2015).

1) PSO relatif mudah diimplementasikan dibandingkan dengan algoritma optimasi lain seperti Algoritma Genetika atau Algoritma *Simulated Annealing*.
2) PSO sering kali konvergen lebih cepat ke solusi optimal atau dekat optimal dibandingkan dengan metode optimasi lainnya.
3) PSO menggunakan informasi kolektif dari semua partikel di dalam swarm untuk memperbarui posisi mereka, sehingga memungkinkan pencarian yang lebih efisien di ruang solusi.
4) PSO dapat diterapkan pada berbagai jenis masalah optimasi, baik itu masalah optimasi berkelanjutan, diskrit, atau bahkan kombinatorial.

Gambar 3.5 merupakan metode dari (X. Li, Wu, dkk., 2020) yang dimodifikasi sedemikian rupa sehingga menjadi metode untuk pemilihan *learning rate* dengan PSO dalam mengoptimalkan model YOLOv8 melalui pemilihan nilai *learning rate* yang tepat. Tahap inisialisasi *particle swarm* dan parameter *Adam* dilakukan secara manual baik secara acak maupun tidak agar proses pelatihan model YOLOv8 dapat berjalan. Proses pelatihan akan menghasilkan *fitness values* yang berasal dari nilai *training error rate* dan *validation error rate* model YOLOv8. Kedua nilai tersebut digunakan sebagai pertimbangan untuk melakukan pembaruan nilai-nilai yang ada

PSO. Proses pembaruan pada PSO dipengaruhi oleh dua faktor, yaitu letak partikel (x) dan kecepatan partikel (v). Sehingga dalam menghitung letak dan kecepatan partikel dirumuskan pada Persamaan (8) dan (9).



Gambar 3.5 Pemilihan *learning rate* dengan PSO

$$v_i(t+1) = w(t)v_i + c_1 r_1\big(pbest_i - x_i(t)\big) + c_2 r_2(gbest_i - x_i(t)) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

Dimana nilai $v_i(t+1)$ adalah kecepatan partikel pada iterasi $t+1$ yang diperoleh dari perhitungan $w(t)$ yang merupakan bobot inersia pada iterasi ke-t, nilai v_i adalah kecepatan partikel pada iterasi ke-t, c_1(koefisien kognitif) dan c_2 (koefisien sosial) adalah nilai konstanta percepatan, r_1 dan r_2 adalah nilai acak dengan rentang 0 sampai 1, $pbest_i$ adalah posisi terbaik yang dicapai oleh partikel ke-i, $gbest_i$ adalah posisi terbaik yang pernah dicapai oleh seluruh populasi, dan $x_i(t)$ adalah nilai posisi partikel ke-i pada iterasi ke-t. Nilai $x_i(t+1)$ adalah posisi

partikel yang telah diperbarui pada iterasi t+1 yang diperoleh dengan menambahkan nilai $x_i(t)$ dengan $v_i(t+1)$. Bobot inersia $w(t)$ yang digunakan untuk menghitung kecepatan partikel didapatkan dengan Persamaan (10) (Kessentini & Barchiesi, 2015).

$$w(t) = \frac{(w_{max} - w_{min}) \times (n_t - t)}{n_t + w_{min}} \quad (10)$$

w_{max} adalah nilai bobot maksimal yang digunakan dengan $w_{max} = 0.9$ dan w_{min} adalah nilai bobot minimal yang digunakan dengan $w_{min} = 0.3$. Selisih dari kedua bobot tersebut akan dikalikan dengan jumlah total iterasi (n_t) yang dikurangi nilai iterasi (t). Hasilnya akan dibagi dengan penyebut yang berasal dari total iterasi (n_t) ditambah dengan w_{min}.

Rumus pada *Adam optimizer* yang akan dipengaruhi oleh pembaruan PSO adalah bagian menghitung momentum (m_t') dan pembaruan bobot (ω_t) pada *hidden layer* arsitektur YOLOv8.

$$m_t' = \beta_1 . m_t + (1 - \beta_1) . g_t \quad (11)$$

$$\omega_t = \omega_{t-1} - \alpha \frac{m_t'}{\sqrt{v_t'} + \varepsilon} \quad (12)$$

Dimana nilai $\alpha = gbest$ merupakan nilai *learning rate* yang akan terus diperbarui untuk mendapatkan nilai α paling optimal. Penggunaan rumus (8)-(10) akan terus dipakai jika *stop condition* pada gambar 4 belum tercapai. *Stop condition* yang digunakan berupa jumlah iterasi pada proses pencarian nilai $\alpha = gbest$ dengan menggunakan PSO. Selain mencari nilai α yang paling optimal, nilai β_1 akan dicari sebagai penyeimbang α agar tidak mengakibatkan *overfitting* pada proses pelatihan. Sehingga Persamaan (11) dapat diubah menjadi Persamaan (13).

$$m_t' = gbest[1].m_t + (1 - gbest[1]).g_t \quad (13)$$

$$\omega_t = \omega_{t-1} - gbest[0]\frac{m_t'}{\sqrt{v_t'+\varepsilon}} \quad (14)$$

Nilai $gbest$ sendiri akan berbentuk vektor yang memiliki dua nilai di dalamnya. $gbest[0]$ dapat digunakan sebagai pengganti dari nilai α, sedangkan $gbest[1]$ dapat digunakan sebagai pengganti nilai β_1.

3.2.7 Pelatihan

Pelatihan model dilakukan jika nilai *learning rate* paling optimal telah didapatkan pada tahap sebelumnya. Pengembang YOLOv8 (Jocher dkk., 2023) telah menyediakan *library* khusus pada *python* untuk membuat model deteksi objek sehingga mempermudah dalam proses pelatihan. Proses pelatihan dilakukan dengan spesifikasi pelatihan yang nanti dijelaskan pada bagian hasil. Selain itu, nilai α dan β_1 menjadi nilai penting yang nanti sangat berpengaruh pada hasil model setelah proses pelatihan. Bagian TTS dan *speech recognition* tidak dilakukan proses pelatihan ulang sehingga penelitian ini hanya menggunakan modul yang sudah tersedia pada *python*.

3.2.8 Integrasi Model dengan *Speech Recognition* dan *Text-to-Speech*

Tahap ini dilakukan setelah model deteksi telah berhasil dibuat pada proses pelatihan. Selanjutnya model diintegrasikan dengan modul *speech recognition* dan TTS yang tersedia pada *python* sehingga kedua modul tersebut hanya berupa penerapan dari model yang sudah ada.

3.2.9 Evaluasi

Terdapat dua jenis evaluasi pada penelitian ini, yaitu evaluasi untuk model deteksi objek dan evaluasi terhadap hasil prediksi jarak yang dilakukan oleh model. Evaluasi untuk model deteksi dilakukan dengan menggunakan metode *Average Precision* (AP) atau *mean Average Precision* (mAP) *metric*. Metode tersebut merupakan metode yang biasa digunakan dalam mengukur performa dari model pengenalan objek (J. R. Terven & Cordova-esparza, 2024). mAP *metric* akan berisi perhitungan *precision-recall metrics* dan menghitung keakuratan *positive prediction* dengan menggunakan *Intersection over Union* (IoU). Selanjutnya, untuk evaluasi terhadap prediksi jarak dilakukan dengan menghitung selisih antara jarak sebenarnya dengan jarak yang diprediksi terhadap beberapa percobaan. Setelah itu, selisih yang ada diubah menjadi bentuk persen dan selanjutnya diakumulasikan untuk melihat secara general tingkat kesalahan yang ada.