

LAPORAN TUGAS AKHIR / *CAPSTONE DESIGN*

ADOPTUKAM: Alat Deteksi Objek pada Tongkat Bantu Tunanetra Berbasis Kamera



Penyusun:

Bayu Arif Ramadhan (18524108)

Reyhan Pramadhana (18524076)

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2022

HALAMAN PENGESAHAN

ADOPTUKAM: Alat Deteksi Objek pada Tingkat Bantu Tunanetra Berbasis Kamera

Penyusun:

Bayu Arif Ramadhan (18524108)

Reyhan Pramadhana (18524076)

Yogyakarta, 18 Juli 2022

Dosen Pembimbing 1

Dosen Pembimbing 2


Dwi Ana Ratna Wati, S.T., M.Eng.

035240102


Suatmi Murnani, S.T., M.Eng.

205241301

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2022

LEMBAR PENGESAHAN TUGAS AKHIR

ADOPTUKAM: Alat Deteksi Objek pada Tingkat Bantu Tunanetra Berbasis Kamera



Disusun oleh:

Bayu Arif Ramadhan 18524108

Reyhan Pramadhana 18524076

Telah dipertahankan di depan dewan penguji
pada tanggal: 05 Agustus 2022

Susunan Dewan Penguji

Ketua Penguji

Anggota Penguji 1

Anggota Penguji 2

: Ir. Medilla Kusriyanto, S.T., M.Eng., IPM.

: Elvira Sukma Wahyuni, S.Pd., M.Eng.

: Amarria Dila Sari, S.T., M.Sc., CBC.

Tugas akhir ini telah disahkan sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Teknik

Tanggal: 15 Agustus 2022

Program Studi Teknik Elektro




Abdullah, S.T., M.Eng., Ph.D.

045240101

PERNYATAAN

Dengan ini kami menyatakan bahwa:

1. Tugas Akhir ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi lainnya, dan sepanjang pengetahuan kami juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Tugas Akhir yang terkait hak milik, hak intelektual, paten merupakan milik bersama antara tiga pihak, yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal ini, penggunaan informasi dan materi Tugas Akhir terkait paten maka akan didiskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 09 Agustus 2022



Bayu Arif Ramadhan (18524108)



Reyhan Pramadhana (18524076)

DAFTAR ISI

HALAMAN PENGESAHAN	ii
LEMBAR PENGESAHAN TUGAS AKHIR	iii
PERNYATAAN	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
RINGKASAN TUGAS AKHIR	1
BAB 1 : Definisi Permasalahan	2
BAB 2 : Observasi	5
BAB 3 : Usulan Perancangan Sistem	9
3.1 Usulan Rancangan Sistem	9
3.1.1 Usulan Rancangan Sistem 1	10
3.1.2 Usulan Rancangan Sistem 2	11
3.1.3 Usulan Rancangan Sistem yang Dipakai	12
3.2 Metode Uji Coba dan Pengujian Usulan Rancangan Sistem	19
3.2.1 Perhitungan Fungsionalitas Alat (Metode <i>Confusion Matrix</i>)	19
3.2.2 Performance Metric	20
3.2.2.1 Task Success Rate	20
3.2.2.2 System Usability Scale (SUS)	21
BAB 4 : Hasil Perancangan Sistem	23
4.1 Kesesuaian Usulan dan Hasil Perancangan Sistem	23
4.2 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya	24
4.3 Analisis dan Pembahasan Kesesuaian antara Perencanaan dan Realisasi	25
BAB 5 : Implementasi Sistem dan Analisis	28
5.1 Hasil dan Analisis Implementasi	28
5.1.1 Metode Confusion Matrix	28
5.1.1.1 Percobaan Deteksi Manusia	30
5.1.1.2 Percobaan Deteksi Sepeda	32
5.1.1.3 Percobaan Deteksi Kursi	33
5.1.2 Task Success Rate	38
5.1.3 System Usability Scale (SUS)	38
5.2 Pengalaman Pengguna	40
5.3 Dampak Implementasi Sistem	40
5.3.1 Teknologi/Inovasi	40
5.3.2 Sosial	41
5.3.3 Ekonomi	41
BAB 6 : Kesimpulan dan Saran	42

6.1 Kesimpulan	42
6.2 Saran	42
DAFTAR PUSTAKA	44
LAMPIRAN – LAMPIRAN	45



DAFTAR TABEL

Tabel 2.1 Kumpulan solusi yang sejenis dengan proyek tugas akhir.	5
Tabel 2.2 Hasil survei antara pengembang dan pengguna	7
Tabel 2.3 Hasil survei produk Tingkat pintar yang beredar	7
Tabel 3.1 Tahapan ilustrasi <i>Design Thinking</i>	10
Tabel 3.2 Inventarisasi kebutuhan usulan sistem ADOPTUKAM	18
Tabel 3.3 Variabel Perhitungan <i>Confusion Matrix</i>	19
Tabel 3.4 Pertanyaan SUS	21
Tabel 4.1 Perbandingan usulan dan hasil perancangan sistem	23
Tabel 4.2 Spesifikasi Usulan dan Realisasi yang Digunakan	24
Tabel 4.3 Kesesuaian antara usulan dan realisasi <i>timeline</i> pengerjaan Tugas Akhir 2	25
Tabel 4.4 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi	25
Tabel 5.1 Akumulasi Hasil Percobaan Deteksi Manusia	30
Tabel 5.2 Hasil Percobaan Deteksi Sepeda	32
Tabel 5.3 Hasil Percobaan Deteksi Kursi	33
Tabel 5.4 Nilai Rata-rata Objek	37
Tabel 5.5 Hasil Pengujian <i>buzzer</i>	38
Tabel 5.6 Hasil Pengisian SUS	39
Tabel 5.7 Pengalaman Pengguna	40
Tabel 5.8 Keunggulan ADOPTUKAM	41



DAFTAR GAMBAR

Gambar 3.1 Alur <i>Design Thinking</i>	9
Gambar 3.2 Proses cara kerja Sistem.	13
Gambar 3.3 <i>Architecture Deep Learning MobileNet</i>	13
Gambar 3.4 (a) Proses Sistem Deteksi (b) <i>Function Detection</i>	15
Gambar 3.5 Flowchart Deteksi Jumlah Objek	15
Gambar 3.6 Pengambilan deteksi objek dengan menghitung objek di <i>frame</i>	16
Gambar 3.7 Flowchart Pemisahan Perintah Perekaman Video dengan Deteksi Objek	17
Gambar 3.8 Ilustrasi usulan rancangan sistem secara umum. (a) Proses cara kerja sistem, (b) Desain Sistem Elektronik (c) Gambaran desain sistem ADOPTUKAM	17
Gambar 3.9 Desain 3D tampak atas dan bawah	18
Gambar 4.1 (a) Sistem usulan (b) realisasi yang digunakan	24
Gambar 4.2 Hasil percobaan YoloV4tiny	26
Gambar 4.3 Percobaan Model Deep Learning MobileNet SSD Untuk Sistem Deteksi	27
Gambar 4.4 Percobaan Model <i>Deep Learning</i> MobileNET SSD + FIFO	27
Gambar 5.1 <i>Luxmeter</i>	28
Gambar 5.2 Proses Pengambilan Data Percobaan	29
Gambar 5.3 Hasil Penangkapan Objek Manusia jarak 1 meter	31
Gambar 5.4 Hasil Penangkapan Objek Sepeda	33
Gambar 5.5 Hasil Penangkapan Objek Kursi	35
Gambar 5.6 Grafik Deteksi Objek Manusia	36
Gambar 5.7 Grafik Deteksi Objek Kursi	37
Gambar 5.8 Grafik Deteksi Objek Sepedah	37
Gambar 5.9 Grafik nilai jarak 1 - 5 m	37
Gambar 5.10 Pengambilan Data SUS	37

RINGKASAN TUGAS AKHIR

Manusia dibekali indera penglihatan oleh Allah SWT untuk memperoleh informasi benda-benda yang ada di sekitarnya, Namun lain halnya dengan penyandang tunanetra yang memiliki keterbatasan pada sensori penglihatan sehingga memerlukan bantuan alat khusus dalam melakukan aktivitasnya. Pada tahun 2018, jumlah data penyandang tunanetra di dunia sebanyak 38,5 juta dan tentu angka ini akan meningkat setiap tahunnya. Kekhawatiran akan menabrak suatu rintangan atau objek yang berada di depannya pada saat berkegiatan sehari-hari sering kita jumpai. Oleh karena itu penyandang tunanetra membutuhkan bantuan dari pihak lain dari sesama manusia, anjing terlatih (*guide dog*) dan alat bantu yang umum dipergunakan di Indonesia seperti tongkat. Namun, kekurangan tongkat konvensional terletak pada terbatasnya jangkauan untuk meraba benda di sekitarnya yang hanya sepanjang tongkatnya. Untuk mengatasi permasalahan tersebut, penulis mengusulkan solusi berupa ADOPTUKAM yang merupakan alat bantu pendeteksian objek pada tongkat tunanetra dengan menggunakan kamera yang dapat mendeteksi objek penghalang di depannya secara *real-time* lalu memberikan peringatan berupa bunyi melalui sebuah *buzzer*. Pada realisasinya, terdapat perubahan dari apa yang telah diusulkan yaitu perubahan pada bagian pemrosesan dari Raspberry Pi 3B+ menjadi Raspberry Pi 4B dengan menggunakan algoritme MobileNetSSD dan menggunakan data set terlatih COCO. Dari hasil pengujian didapatkan nilai akurasi pendeteksian jarak 1 hingga 5 meter yang dilakukan di dalam ruangan dengan intensitas cahaya sebesar 421 lux sebesar 82.2%, presisi sebesar 91%, *Recall* 92.2%, dan *F1-Score* 89.4%. Persentase *buzzer* bekerja dalam mengeluarkan bunyi ketika kamera menangkap gambar objek adalah sebesar 100%. Oleh karena itu, dapat disimpulkan bahwa ADOPTUKAM dapat menjadi salah satu teknologi asistif yang ditawarkan untuk membantu mobilitas penyandang tunanetra dengan biaya yang tidak terlalu mahal serta dari dampak segi sosial tongkat ini akan meningkatkan kepercayaan diri penggunaannya dengan berjalan lebih aman dan efektif.



BAB 1 : Definisi Permasalahan

Penyandang disabilitas adalah manusia normal yang memiliki keterbatasan dalam diri seseorang tersebut yang dapat berupa keterbatasan mental, fisik, sensorik, kognitif serta kombinasi dari beberapa keterbatasan diri yang ada pada seseorang. Faktor terjadinya disabilitas antara lain adalah seseorang tersebut mengalami kecelakaan atau musibah, pasca operasi dan juga bawaan dari lahir [1]. Di Indonesia sendiri, mengutip dari data berjalan 2020 Biro Pusat Statistik (BPS), penyandang disabilitas mencapai 22,5 juta atau sekitar 5% dengan 3,75 juta penduduk Indonesia adalah penyandang tunanetra [2]. Pengertian tunanetra sendiri mengutip dari Persatuan Tunanetra Indonesia (Pertuni) yaitu kondisi manusia yang tidak memiliki penglihatan sama sekali (buta total) dan juga memiliki sisa penglihatan namun tidak mampu menggunakan penglihatannya untuk membaca tulisan biasa berukuran 12 point dalam keadaan cahaya normal dari jarak normal baca walaupun sudah menggunakan alat bantu baca berupa kacamata (*low vision*) [3].

Penyandang disabilitas sendiri termasuk salah satu kelompok minoritas karena seringkali pemenuhan hak-hak nya tidak diperhatikan oleh suatu negara sehingga kerap mendapat perlakuan diskriminasi. Adapun penyandang disabilitas memiliki kedudukan, hak, dan kewajiban yang sama dengan masyarakat non disabilitas sehingga sudah seharusnya para penyandang disabilitas mendapatkan perlakuan khusus untuk perlindungan dari berbagai pelanggaran hak asasi manusia. Di Indonesia sendiri, dalam UU Nomor 8 Tahun 2016 tentang penyandang disabilitas, sebagai jawaban dari kondisi para penyandang disabilitas khususnya di Indonesia untuk mendapatkan kesetaraan hak yang sama dalam hidup untuk memberikan ruang bagi mereka untuk dapat hidup sejahtera dan mandiri, salah satu upaya pelaksanaan dan pemenuhannya adalah aksesibilitas, pelayanan publik, dan perlindungan dari bencana [4]. Hal tersebut juga dijelaskan, mengutip Artikel 19 Organisasi PBB, negara-negara yang terdapat di organisasi tersebut mengakui kesetaraan hak bagi penyandang disabilitas untuk hidup dalam bermasyarakat dan mendapat pilihan yang sama untuk hidup serta memfasilitasi akomodasi yang layak seperti alat bantu kesehatan dan pelayanan publik khusus [5]. Alat bantu kesehatan atau media dalam membantu penyandang disabilitas sering disebut juga dengan Teknologi Asistif (*Assistive Technology*) yaitu sebuah teknologi pada alat yang dimodifikasi atau langsung digunakan untuk meningkatkan kemampuan penyandang disabilitas [6]. Menurut WHO, hanya 1 dari 10 orang memerlukan akses untuk mendapatkan suatu teknologi asistif akibat biaya yang tinggi, keterbatasan alat atau tidak ada layanan publik yang memiliki hal tersebut dengan hampir 200 juta orang yang memiliki *low vision* tidak memiliki akses untuk mendapatkan produk teknologi asistif [7].

Penyandang tunanetra umumnya menggunakan alat bantu berupa tongkat tunanetra konvensional untuk membantu mobilitas kegiatan sehari-hari dengan cara menghentakkan tongkat

tersebut untuk meraba area sekitar sehingga jangkauannya hanya terbatas dari seberapa panjang tongkat yang digunakan dan mengalami kesulitan menghindari rintangan seperti menghindari benda padat, lubang dan objek lainnya. Minimnya teknologi pada tongkat konvensional yang belum mampu mendeteksi objek atau benda di depannya membuat penyandang tunanetra memiliki salah satu kesulitan sendiri apabila harus melakukan mobilitas di tempat yang terdapat suatu objek di depannya.

Pada langkah awal pengembangan, terdapat beberapa Batasan Realistis *Engineering Aspect* yang penulis buat yaitu:

1. Intensitas Cahaya
Pengujian deteksi objek dilakukan dengan intensitas cahaya yang bagus dan terang, alat tidak diuji dalam kondisi gelap.
2. Catu Daya
Baterai sebagai catu daya pemrosesan agar tongkat dapat melakukan pendeteksian pada saat penggunaan sehingga dapat digunakan ketika perjalanan
3. Biaya
Melihat dari faktor ekonomis, perlu dilakukakan penekanan biaya agar alat yang dibuat dapat terjangkau oleh semua kalangan.
4. Tidak Tahan Air.
5. Alat bantu pendeteksian objek hanya berupa kamera.

Lingkup permasalahan / Batasan Masalah pada proyek *capstone* ini adalah sebagai berikut:

1. Peruntukkan Tongkat
Tongkat ditujukan untuk membantu kebutuhan para penyandang disabilitas tunanetra pada saat berkegiatan.
2. Lokasi Pendeteksian
Dalam langkah awal pengembangan, pengujian dilakukan di dalam Laboratorium Pengolahan Sinyal Telekomunikasi (PST) Teknik Elektro Universitas Islam Indonesia serta Aula Kecamatan Karanganyam Yayasan LB dengan tidak melakukan pengujian di jalanan.
3. Ukuran Panjang Tongkat 110 cm.
Panjang tongkat yang digunakan mengikuti standar di Indonesia yang digunakan untuk usia dewasa yaitu 110 cm.
4. Jenis Objek yang diuji.
Tongkat hanya mendeteksi Manusia, Kursi, dan Sepeda dalam keadaan diam.
5. Hanya memanfaatkan indera pendengaran sebagai komunikasi menggunakan *buzzer*.

Berdasarkan permasalahan yang telah dijelaskan sebelumnya, maka tujuan dari pengembangan tongkat pada penelitian ini adalah alat atau tongkat mampu membantu penyandang tunanetra berjalan lebih efektif dengan memudahkan mereka dengan pendeteksian objek sehingga dapat melewati berbagai rintangan yang dihadapi pada saat melakukan perjalanan. Hal ini diharapkan menjadi langkah awal sebuah penelitian yang dapat diterapkan pada Yayasan Tunanetra di daerah Karanganyar.



BAB 2 : Observasi

Proses observasi dilakukan dengan tujuan untuk memastikan bahwa rancangan sistem yang diusulkan sesuai dengan batasan yang dibuat sebelumnya. Langkah awalnya adalah dengan melakukan studi literatur yang berkaitan dengan alat atau sistem yang telah dibuat oleh penelitian-penelitian sebelumnya melalui jurnal, artikel, dan referensi lainnya yang kredibel dan juga beberapa produk tongkat pendeteksi yang telah beredar serta pengumpulan informasi-informasi yang dibutuhkan dari penyandang tunanetra. Langkah selanjutnya adalah menentukan pemilihan spesifikasi komponen yang diharapkan mampu memenuhi kebutuhan pengguna yaitu penyandang tunanetra.

Pada Tabel 2.1 terdapat beberapa kumpulan solusi yang sejenis dengan pengerjaan proyek tugas akhir untuk memperkaya informasi dalam mengetahui solusi yang mungkin diusulkan yaitu pendeteksi objek yang dapat membantu penyandang tunanetra dalam beraktivitas yang didapatkan dari studi literatur oleh penulis.

Tabel 2.1 Kumpulan solusi yang sejenis dengan proyek tugas akhir.

Penulis	Usulan Solusi	Hasil / Evaluasi
A. Nivedita, dkk (2019) [8]	<i>Smart Cane Navigation for Visually Impaired</i>	Hasil tongkat tunanetra pendeteksi objek menggunakan Raspberry Pi 3B dengan sensor ultrasonik dan <i>Webcamera</i> sebagai sensor pendeteksi objek yang ada di depannya dan <i>microphone</i> sebagai penerima <i>feedback</i> namun tidak dijelaskan hasil eksperimen yang didapatkan bagaimana hanya menampilkan tampilan gambar yang tertangkap oleh <i>webcam</i> akan dikirimkan ke Raspberry.
S. Fuady, dkk (2020) [9]	Deteksi Objek Menggunakan Metode Single Shot Multibox Detector pada Alat Bantu Tongkat Tunanetra Berbasis Kamera	Penelitian dengan menggunakan Raspberry Pi dan Webcam dengan metode Single Shot Multibox Detector yang mampu memproses citra menjadi <i>real-time</i> dengan tingkat akurasi yang tinggi dengan akurasi terbesar yaitu 92% didapatkan karena kondisi citra saat pengujian yaitu keadaan terang dan posisi yang tepat adapun dengan menggunakan <i>webcam</i> maka akan membebani pengguna akibat penambahan beban yang terjadi pada tongka.
T. Gowthaman, dkk (2015) [10]	<i>Smart Cane for Visually Impaired People</i>	Sistem yang meliputi kamera digunakan untuk menangkap citra dan mengirimkannya ke PC lalu gambar objek tersebut akan dipindai menggunakan <i>image processing scanner</i> menggunakan algoritma Hough Transform dengan RS232 digunakan untuk komunikasi serial antara dengan gambar yang terdaftar atau cocok akan mengeluarkan suara melalui speaker dan sensor ultrasonik digunakan untuk menentukan jarak objeknya. Sistem akan sulit untuk dibawa oleh pengguna tongkat karena terlalu banyaknya alat yang digunakan serta akan menghabiskan biaya

Penulis	Usulan Solusi	Hasil / Evaluasi
		yang besar dan pendeteksian gambar belum secara <i>real-time</i>
R. Cahya, dkk (2019) [11]	<i>Object Detection System</i> Sebagai Alat Bantu Mendeteksi Objek Sekitar untuk Penyandang Tunanetra	Sistem yang berhasil dibuat tanpa menggunakan tongkat tunanetra sebagai media teknologi asistifnya melainkan menggunakan <i>smartphone</i> selama 1 hingga 2 detik perhitungan manual yang dilakukan dari proses pengambilan gambar hingga selesai berbicara dengan menggunakan metode <i>Harris Corner Detection</i> sebagai mendeteksi sudut dari objek pada gambar yang telah diambil dengan akurasi yang didapatkan sebesar 88%.

Berdasarkan hasil studi literatur melalui referensi-referensi yang saling berkaitan antara penelitian satu dan lainnya, umumnya alat dirancang menggunakan komponen elektronik yang membebani tongkat sehingga menjadi lebih berat dan harga menjadi lebih mahal. Pendeteksian objek dapat terlaksana dengan menggunakan alat bantu penangkapan gambar berupa kamera dan sensor ultrasonik. Sensor ultrasonik umum digunakan karena memanfaatkan gelombang ultrasonik. Kemudian dalam pemrosesan deteksi objek dapat menggunakan komputer pribadi (PC) dan Raspberry Pi. Raspberry Pi menjadi solusi paling baik karena memanfaatkan kemudahan dalam aksesibilitas sehingga dapat dibawa karena bentuknya yang kecil sehingga ideal untuk diterapkan pada tongkat tunanetra meskipun demikian, terdapat pengorbanan berupa waktu pemrosesan di mana pemrosesan menggunakan Raspberry Pi akan lebih lambat dibandingkan menggunakan PC. Teknologi yang digunakan yaitu dengan menggunakan *Computer Vision* yang mana merupakan sebuah teknologi yang dapat bekerja menyerupai kerja penglihatan manusia yang contoh kegunaannya dalam *face recognition*, *color detection*, dan *object detection* dengan menggunakan OpenCV sebagai *library* yang sangat familiar dalam pengolahan citra *Computer Vision* yang didesain untuk penggunaan secara *real-time* [12]. Model yang paling populer digunakan adalah YOLO (*You Only Look Once*) dan SSD yang merupakan *one stage object* yang cocok digunakan karena mampu mengatasi kebutuhan *computing* yang besar. Hal ini dikarenakan untuk melakukan *object detection*, dibutuhkan perangkat seperti GPU atau spesifikasi PC yang cukup tinggi. Oleh karena itu, pemodelan menggunakan SSD MobileNet atau YOLO merupakan pilihan yang tepat. Aktuator berupa *buzzer* dan *vibration motors* digunakan untuk memberikan peringatan kepada pengguna tongkat tersebut.

Proses setelah melakukan observasi kebutuhan sistem adalah proses survei untuk mendapatkan informasi tambahan dengan mengajukan beberapa pertanyaan terkait penggunaan tongkat tunanetra kepada penyandang tunanetra. Penyandang tunanetra yang diberikan pertanyaan bernama Rizky yang merupakan salah satu siswa SMALB di Klaten, Jawa Tengah. Hasil survei dirangkum pada Tabel 2.2.

Tabel 2.2 Hasil survei antara pengembang dan pengguna

Pertanyaan	Jawaban/tanggapan
Seberapa sering penggunaan tongkat dilakukan dan kurang lebih berapa jam?	Menggunakan tongkat disaat waktu waktu tertentu seperti dalam perjalanan dari satu tempat ke yang lainnya. tidak pasti waktunya mas tapi yang pasti tidak sampai 24 Jam
Apakah anda keberatan ketika tongkat tunanetra tradisional diberikan komponen elektronis untuk membantu mobilitas anda?	Tidak terlalu bermasalah bagi asal tongkat tidak terlalu ribet dan membuat pegal
Masalah apa saja yang kerap terjadi meskipun sudah menggunakan tongkat tunanetra?	Seringkali salah arah, menabrak suatu benda walaupun sudah hafal daerah namun lebih sering terjadi di lingkungan baru namun menurut saya itu hal yang lumrah
Upaya apa yang anda lakukan ketika hal yang disebutkan sebelumnya(menabrak, jatuh) terjadi?	Salah satu bagian yang dapat mempengaruhi gagal panen atau tidak, sehingga memang perlu cara yang lebih efektif untuk memperhatikan kondisi tanah
Apakah anda bersedia mengeluarkan biaya 2-7 Juta untuk Tongkat yang dapat mendeteksi objek yang membantu mobilitas anda?	Tidak kalau di atas 4 juta, walaupun ada pendamping saya lebih memilih bersama pendamping saya dibanding menggunakan tongkat yang terlalu mahal
Apakah teknologi yang dapat melakukan deteksi objek lalu memberikan <i>warning</i> kepada anda untuk membantu anda dari menabrak dan terjatuh membantu ?	Sangat membantu mas asal saya masih dapat menggunakan tongkat selayaknya tongkat biasa

Berdasarkan informasi yang didapatkan dari hasil survei/wawancara dengan pengguna tongkat tunanetra dan penyandang tunanetra dan penelusuran beberapa literatur/teknologi yang telah dikembangkan, selanjutnya mencari harga tongkat tunanetra yang dapat membantu melakukan mobilitas penyandang tunanetra yang telah beredar di pasaran beserta fituranya. Informasi harga tongkat tunanetra tersebut dirangkum pada Tabel 2.3.

Tabel 2.3 Hasil survei produk Tongkat pintar yang beredar

Nama Produk	Spesifikasi	Sumber	Harga (Rp.)
WeWalk Smart Cane	Berat: 252 gr (tanpa tongkat) Panjang: 28 cm (tanpa tongkat) Jarak Deteksi Benda: 80 cm - 170 cm Baterai: 1000mAh Li-ion Sensor: Ultrasonic Dapat terkoneksi dengan <i>Smartphones</i> menggunakan Bluetooth 4.2	Product Details - WeWALK Smart Cane	7.500.000,00
BAWA Cane	Berat: Kurang dari 350 gr Panjang: tidak dijelaskan Jarak Deteksi Benda: Up to 1.2 m cm Baterai: Up to 10 Jam Sensor: Dual Sensors (tidak dijelaskan) Mengetahui lokasi pengguna melalui aplikasi BAWA pada <i>Smartphone</i> lalu diarahkan ke tempat tujuan melalui <i>voice Speaker</i>	BAWA Cane - Recreating Hope for the Visually Impaired	10.000.000,00

Berdasarkan hasil studi literatur yang dilakukan penulis, diperlukan spesifikasi tongkat bantu tunanetra yang dapat mendeteksi objek untuk membantu mobilitas penyandang tunanetra. Untuk mengatasi permasalahan tersebut, penulis merancang daftar spesifikasi lengkapnya sebagai berikut.

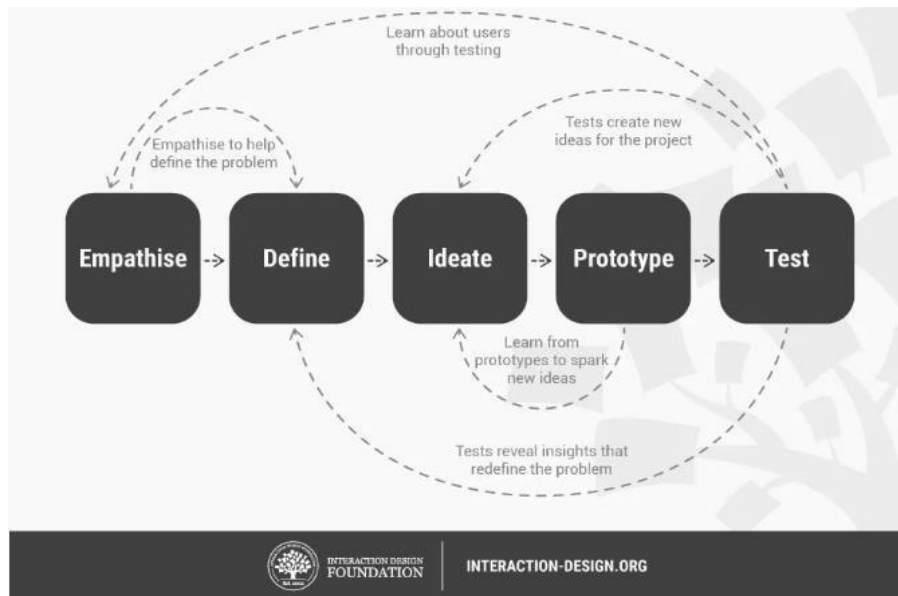
1. Tongkat pendeteksi objek tunanetra menggunakan bahan Alumunium Galvanis dan memiliki panjang 110 cm - 120 cm, yang merupakan standar panjang untuk penyandang tunanetra kategori orang dewasa.
2. Sistem yang dirancang mampu mendeteksi objek penghalang di depan pengguna dengan rentang jarak 0.5 m - 5 m.
3. Alat bantu untuk mendeteksi sebuah objek yang digunakan berupa kamera.
4. Menggunakan catu daya baterai sebesar 10.000 mAh.
5. Memberikan peringatan atau notifikasi adanya halangan berupa suara melalui *buzzer*.



BAB 3 : Usulan Perancangan Sistem

3.1 Usulan Rancangan Sistem

Dalam proses perancangan produk tingkat deteksi objek berbasis kamera, beberapa tahapan dilakukan dengan menggunakan *Design Thinking* yang merupakan proses pemecahan masalah dengan enam buah tahapan, yaitu *Empathize*, *Define*, *Ideate*, *Prototype*, *Test*, dan *Implement*. Tahapan tersebut ditunjukkan pada Gambar 3.1 dan Tabel 3.1 namun dapat terjadi perubahan dalam prosesnya dalam mencapai target spesifikasi perancangan.



Gambar 3.1 Alur *Design Thinking* (Sumber: [Stage 2 in the Design Thinking Process: Define the Problem and Interpret the Results | Interaction Design Foundation \(IxDF\) \(interaction-design.org\)](https://www.interaction-design.org/learn/article/stage-2-in-the-design-thinking-process-define-the-problem-and-interpret-the-results))

Pada tahapan *empathize* hingga *define* dilakukan dalam bentuk inspirasi yang didasari oleh seorang penyandang disabilitas tunanetra yang selalu membutuhkan tongkat untuk menunjang aktivitas sehari-harinya. Selama tahapan itu juga terlihat jika tongkat yang digunakan terdapat beberapa bagian yang menjadi kekurangan pada penggunaannya, mulai dari terjadinya tabrakan dengan objek di depannya dan tidak adanya pemberi isyarat/peringatan bagi pengguna maupun orang di sekitarnya jika berhadapan dengan objek di depannya.

Sehingga dari tahap *empathize* dan *define* tersebut dilakukan sebuah inovasi yang didasari oleh tahapan *ideate* hingga *prototype*, inovasi yang dibuat meliputi sistem untuk alat deteksi objek pada tongkat bantu tunanetra berbasis kamera yang dapat disingkat sebagai ADOPTUKAM. Dengan fokus inovasi pada pembaharuan sistem deteksi dan juga pengurangan biaya produk, sehingga harapannya ADOPTUKAM dapat memudahkan penyandang disabilitas tunanetra untuk dapat menghindari objek yang sudah ditentukan.

Tabel 3.1 Tahapan ilustrasi *Design Thinking*

Tahapan	Keterangan
<i>Empathize</i>	Penulis melakukan studi literatur terkait permasalahan yang sering terjadi kepada para penyandang disabilitas tunanetra menggunakan tongkat tradisional. tahapan dilakukan dengan membaca atau <i>review</i> berbagai jurnal terkait dan referensi kredibel lainnya serta survei dengan melakukan wawancara dengan salah satu penyandang Tunanetra.
<i>Define</i>	Informasi - informasi yang telah dikumpulkan sebelumnya lalu menjadi dasar proses perumusan masalah yang akan dikerjakan dalam hal ini fokus permasalahan terletak pada kekurangan pada tongkat tunanetra tradisional, .
<i>Ideate</i>	Mencari solusi-solusi alternatif yang ada untuk mendapatkan solusi terbaik dari usulan tersebut salah satu solusi tersebut yaitu pembuatan tongkat yang dapat melakukan deteksi objek benda atau penghalang di depannya untuk membantu mobilitas penyandang tunanetra
<i>Prototype</i>	Melaksanakan perancangan alat sesuai dengan solusi yang telah diusulkan dengan memperhatikan komponen ataupun fitur yang harus ada pada alat, fitur apa yang berfungsi dan tidak agar alat yang diusulkan pada <i>prototype</i> ini mampu menjawab permasalahan tersebut.
<i>Test</i>	Dilakukan sebuah uji coba dari alat yang telah dibuat dengan tujuan mengetahui apakah alat yang dibuat sesuai dan dapat menyelesaikan masalah atau tidak.
<i>Implement</i>	Tongkat Tunanetra yang dapat mendeteksi objek tersebut kemudian dilakukan pengujian langsung di ruangan <i>indoor</i> Laboratorium PST Elektro UII serta Yayasan Karangnom Klaten

3.1.1 Usulan Rancangan Sistem 1

Usulan rancangan sistem pertama ADOPTUKAM menggunakan alat bantu untuk mengambil gambar atau citra sebuah objek dengan menggunakan tongkat tunanetra yang dapat dilipat dengan bahan Alumunium Galvanis dengan panjang 110 cm. Modul kamera Raspberry Pi CAM V1.3 dengan integrasi mikroprosesor Raspberry Pi 4B 4GB dan baterai Li-Po Powerbank untuk sebuah pemrosesan citra sebagai pendeteksi objek. kemudian *buzzer* akan berbunyi ketika objek telah berhasil dideteksi oleh kamera hasil tangkapan gambar objek yang terdeteksi. kelebihan dan kekurangan dari usulan rancangan sistem pertama, yaitu.

- Kelebihan:
 1. Dapat mendeteksi objek secara *real-time*.
 2. Dapat menampilkan gambar objek dengan jelas
 3. Jarak deteksi optimal dalam rentang 1 m sampai 5m.
 4. Dimensi Kemasan 3D print dengan Bahan PLA dengan $P \times L \times T = 12 \times 9 \times 11$ cm
 5. Bobot alat kurang dari 1 kg.

6. FPS yang didapatkan saat mendeteksi sekitar 1 sampai 3 FPS
 7. Pengisian daya pada alat menggunakan USB type C
 8. Resolusi kamera sebesar 5MP.
- Kekurangan:
 1. Menggunakan VNC Viewer untuk menampilkan hasil tangkapan gambar sebuah objek.
 2. Harus terhubung dengan jaringan internet.
 3. Alat membutuhkan *Port* HDMI untuk mengoperasikan alat Raspberry perlu terhubung dengan monitor.

3.1.2 Usulan Rancangan Sistem 2

Usulan rancangan sistem kedua ADOPTUKAM menggunakan alat bantu tongkat tunanetra yang dapat dilipat dengan bahan Alumunium Galvanis dengan panjang 110 cm untuk mengambil gambar sebuah objek dengan menggunakan modul Raspberry Pi CAM V2 yang terintegrasi dengan mikroprosesor Raspberry Pi 3B+ 1GB dengan penambahan modul Intel NCS Movidius untuk mempercepat proses komputasi dan mempercepat pengembangan aplikasi *deep neural network*. Kelebihan dan kekurangan dari usulan rancangan sistem kedua, yaitu.

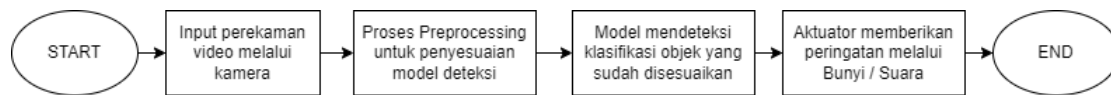
- Kelebihan:
 1. Dapat mendeteksi objek secara *real-time*.
 2. Dapat menampilkan gambar objek dengan jelas
 3. Jarak deteksi dari 1 meter hingga 5 meter.
 4. Resolusi kamera sebesar 8MP
 5. FPS yang didapatkan sekitar 1 - 3 FPS
- Kekurangan:
 1. Bobot hampir 1 kg akibat penambahan komponen modul Intel NCS
 2. Menggunakan VNC Viewer untuk menampilkan hasil tangkapan gambar sebuah objek.
 3. Harus terhubung dengan jaringan internet.
 4. Alat membutuhkan *Port* HDMI untuk mengoperasikan alat Raspberry perlu terhubung dengan monitor
 5. Biaya lebih mahal

3.1.3 Usulan Rancangan Sistem yang Dipakai

ADOPTUKAM dirancang dengan menggunakan standar ISO 9999:2016 mengenai teknologi asistif yang digunakan dan menjamin produk yang diproduksi secara umum untuk membantu penyandang disabilitas [13]. Sistem pada ADOPTUKAM mampu mendeteksi sebuah objek menggunakan model *deep learning* yaitu *MobileNet* dengan metode SSD (*Single Shot Multibox Detector*) dengan pemrosesan menggunakan Raspberry Pi 4B yang telah diberikan set data terlatih *Common Object in Context Detection Challenge (COCO)*. COCO dirancang mendeteksi objek secara *real-time* dengan tujuan membantu mobilitas para penyandang tunanetra guna meminimalisir terjadinya kecelakaan yang terjadi seperti menabrak benda yang ada di sekitarnya dengan menekan biaya produk agar dapat terjangkau oleh semua kalangan. ADOPTUKAM juga dilengkapi dengan sebuah kamera yang bekerja sebagai perekaman keadaan untuk nantinya perekaman tersebut diolah oleh *microprocessor* untuk dapat mendeteksi adanya objek sebelum tongkat dan pengguna mendekati objek tersebut dengan cara memberikan peringatan yang dilakukan oleh *Buzzer* berupa bunyi/suara. Peringatan tersebut digunakan karena para penyandang tunanetra lebih peka dengan indera lainnya.

Objek yang dipilih merupakan objek umum yang ada pada dalam ruangan seperti Kursi, Manusia, dan pemilihan Sepeda atau Motor ketika berada diluar ruangan dan pada jumlah objek yang bisa dideteksi akan dibatasi karena akan mempengaruhi kerja *processing* dalam *microprocessor* dari sistem yang mengakibatkan terjadinya *delay* pada saat pengambilan citra. Peringatan yang berupa suara itu merupakan tanda bahwa ada objek yang akan dihadapinya sehingga pengguna akan menghindari laju tersebut dan perlahan mengubah arah untuk menghindari terjadinya tabrakan dengan objek tersebut.

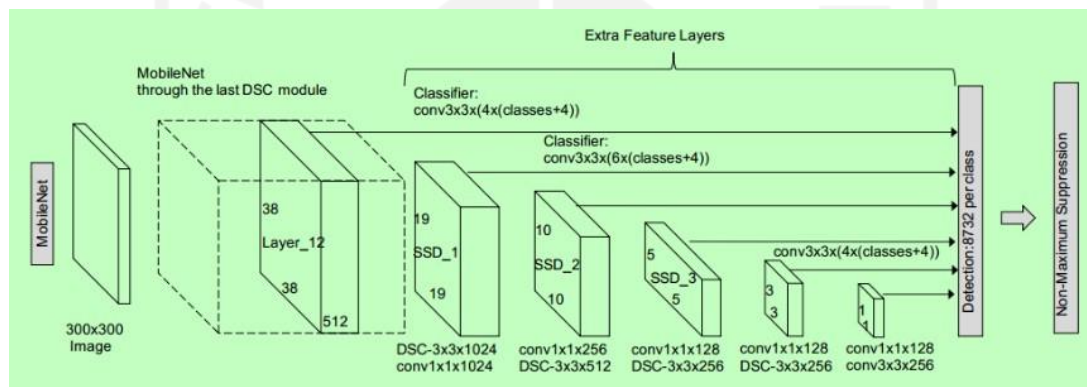
ADOPTUKAM dinyalakan melalui *software VNC (Virtual Network Computing) Viewer* proses tersebut dilakukan untuk memanggil model dan juga sistem objek deteksi di dalam *microprocesor* yaitu Raspberry Pi 4B. Selanjutnya, kamera yang tersambung dengan *microprocessor* akan melakukan pengambilan gambar secara *real-time* dengan proses yang terpisah oleh model pengenalan untuk mendeteksi objek yang sudah ditentukan sebelumnya. Jika kamera mendeteksi objek yang sesuai, maka *microprocesor* akan memberikan *output* untuk memicu aktifnya suara oleh *buzzer* untuk memberitahukan bahwa ada objek yang berada di sekitar deteksi. Gambar 3.2 merupakan penjelasan proses cara kerja sistem secara umum dari ADOPTUKAM.



Proses Deteksi Objek Secara Umum

Gambar 3.2 Proses cara kerja Sistem.

Pemrosesan citra dalam mendeteksi objek dilakukan dengan menggunakan model *deep learning MobileNet* dengan metode SSD (*Single Shot Multibox Detection*). Pendeteksian suatu objek dilakukan dengan sistem *layer* tunggal (*Single Layer*). Metode ini akan menandai area objek menggunakan *bounding box* dengan cara menyesuaikan skala dan rasio dari setiap lokasi *feature map* pada penangkapan gambar. Penggunaan metode SSD diperuntukkan agar deteksi bisa membandingkan objek dengan *default bounding box* pada rasio *training data*. Metode ini mempunyai prinsip penggunaan *layer* untuk menyesuaikan skala yang nantinya dapat memberikan hasil terbaik dari objek yang sudah terdeteksi.



Gambar 3.3 Architecture Deep Learning MobileNet (Sumber: [Object Detection using the TensorFlow API \(analyticsvidhya.com\)](https://analyticsvidhya.com))

Ketika citra diolah dengan model MobileNet, metode SSD 6 layer konvolusi ekstra di mana pada 3 layer ekstra dapat menghasilkan enam prediksi untuk setiap sel. Dengan menggunakan metode SSD, sistem dapat menghasilkan total sejumlah 8732 prediksi dalam memanfaatkan 6 *layer* pada model. Pada model *layer* ekstra tersebut juga dihasilkan *feature maps* dalam berbagai ukuran untuk mendeteksi objek dalam berbagai ukuran sehingga mampu memberikan akurasi yang lebih baik terhadap objek-objek yang memiliki ukuran berbeda dalam suatu gambar [14].

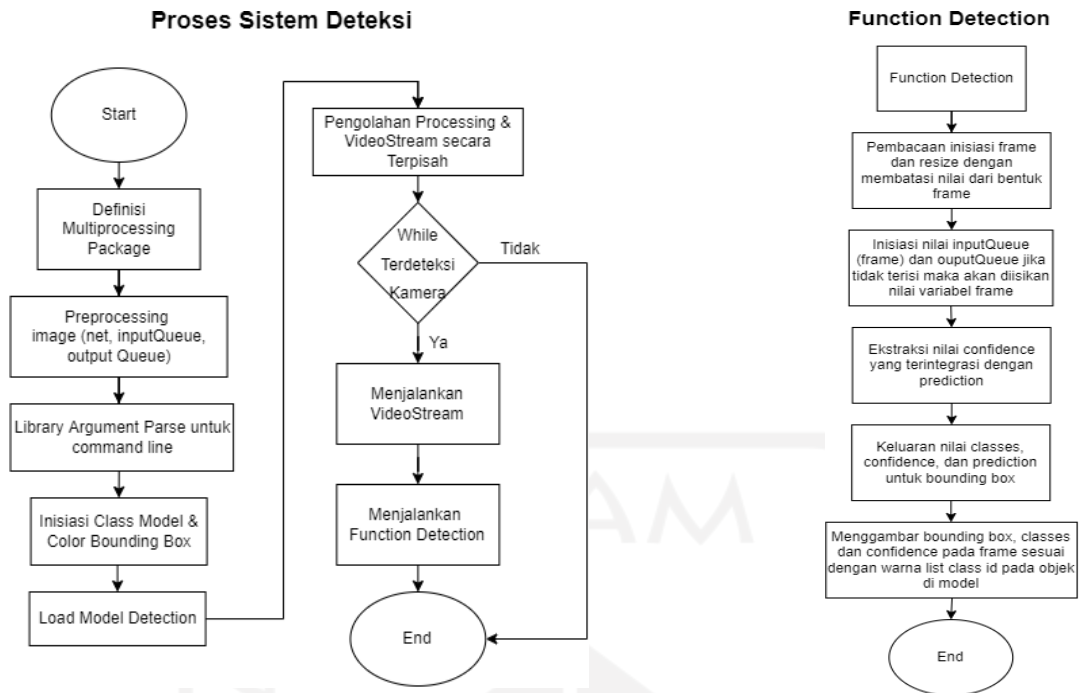
Selanjutnya, dalam menerjemahkan model *deep learning*, terdapat beberapa alur proses bagaimana MobileNet SSD mendeteksi suatu objek mulai dari *frame* perekaman video ke model hingga keluaran yang dihasilkan dan dimunculkan pada *frame* yang terdapat *bounding box*. *flowchart* pada Gambar 3.4. menunjukkan proses dari penerjemahan model. Proses ini juga akan digunakan pada deteksi jumlah objek dan pemisahan perekaman video pada kamera dengan sistem objek deteksinya.

Proses deteksi objek dengan model *MobileNet SSD* dimulai untuk mendapatkan *frame* perekaman video menggunakan kamera Raspberry Pi V1.3 Cam. Selanjutnya pada model dijalankan dengan DNN (*Deep Neural Network*) *Caffe* pada *cv2*, sebelum *frame* perekaman video diolah model, *frame* akan melalui *preprocessing*.

Selanjutnya *Preprocessing* menggunakan *blobFromImage*. Fungsi ini menggunakan beberapa parameter mulai dari *frame*, *scale*, *size*, *inputQueue*, dan *outputQueue*. Pada input dilakukan pengaturan agar blob ke objek deep learning untuk deteksi dan mendapatkan fungsi deteksi. Selanjutnya dilakukan pembuatan *construct* pada *argument parse* untuk *command line*. Setelah itu dilakukan inisiasi daftar label pada kelas model *MobileNet SSD* yang sudah di-*training* sebelumnya untuk pendeteksi, lalu dilakukan pengaturan warna *bounding box* untuk setiap kelasnya.

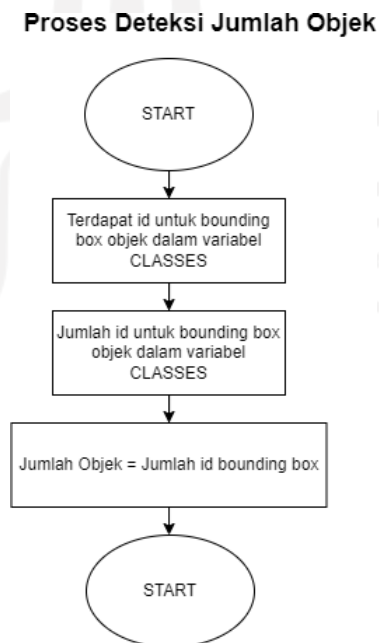
Setelah itu, model *deep learning* dibuat dengan DNN dan membaca lewat *framework caffe*, setelah dilakukan *load model* inisiasi *input (frame)* dan *output (deteksi)* dan juga daftar dari deteksi aktual dikembalikan pada sistem *independent* yaitu tahap *preprocessing*. Selanjutnya dilakukan *preprocessing* dengan perekaman video secara terpisah, di mana pada saat kamera terdeteksi dan objek sudah terdeteksi maka dilakukan perekaman video dan juga menjalankan fungsi *detection*.

Keluaran dari proses deteksi objek diberi nama ID objek, nilai *confidence*, *width*, *height* dari *bounding box* kelas yang terdeteksi. Keluaran tersebut akan digunakan untuk menggambar *bounding box* pada *frame* gambar kamera Raspberry Pi V1.3 Cam. Warna dari *bounding box* juga berbeda tergantung kelas dari ID objek, jika *bounding box* pada ID objek muncul di *frame* akan dihitung berapa objek yang tertampil di *frame*. *Flowchart* pada sistem deteksi sebagai penerjemahan model menjadi pendeteksi objek ditunjukkan pada Gambar 3.4.

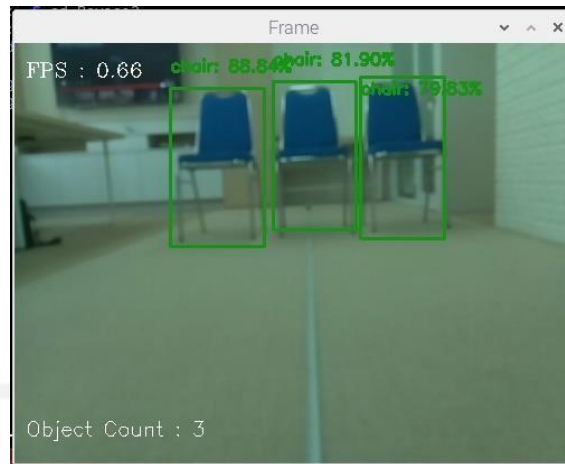


(a) (b)
 Gambar 3.4 (a) Proses Sistem Deteksi (b) *Function Detection*

Proses perhitungan jumlah objek yang terdeteksi dalam *frame*. dimulai dengan pembuatan inisiasi variabel dari jumlah objek dengan nilai 0. Dengan mengambil variabel *CLASSES* pada deteksi objek yang berisikan ID pada *bounding box* yang tertangkap pada *frame*, dan jumlah dari id pada variabel dihitung dan disimpan pada *array* yaitu *len()*. Selanjutnya pada jumlah total objek merupakan nilai dari jumlah semua ID yang ada pada variabel. Berikut *flowchart* dari perhitungan jumlah objek pada *frame* dan hasil yang ditampilkan pada Gambar 3.5 dan 3.6



Gambar 3.5 Flowchart Deteksi Jumlah Objek



Gambar 3.6 Pengambilan deteksi objek dengan menghitung objek di *frame*

Dalam alur pemisahan *Video Capture* dengan sistem deteksi objek menggunakan Raspberry Pi 4B terdapat kekurangan dalam spesifikasi minimum untuk melakukan komputasi model *deep learning* sehingga menyebabkan adanya delay dalam setiap deteksi dan penurunan FPS (*Frame Per Second*), sehingga untuk mempersingkat waktu dari komputasi dan menjaga agar respon deteksi tidak tertinggal dengan *real time* dilakukan pemisahan antara *video capture* dengan sistem deteksi objek. Berikut hasil yang didapatkan sebelum pemisahan.

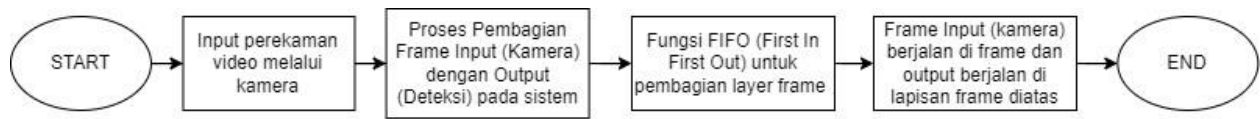
```
$ python capstone.py \
  --prototxt MobileNetSSD_deploy.prototxt.txt \
  --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting video stream...
[INFO] elapsed time: 54.70
[INFO] approx. FPS: 0.90
```

Setelah memisahkan komputasi model untuk deteksi objek dengan perekaman video yang dilakukan untuk dapat meminimalisir *lagging* selama komputasi.

```
$ python capstone.py \
  --prototxt MobileNetSSD_deploy.prototxt.txt \
  --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting process...
[INFO] starting video stream...
[INFO] elapsed time: 48.55
[INFO] approx. FPS: 27.83
```

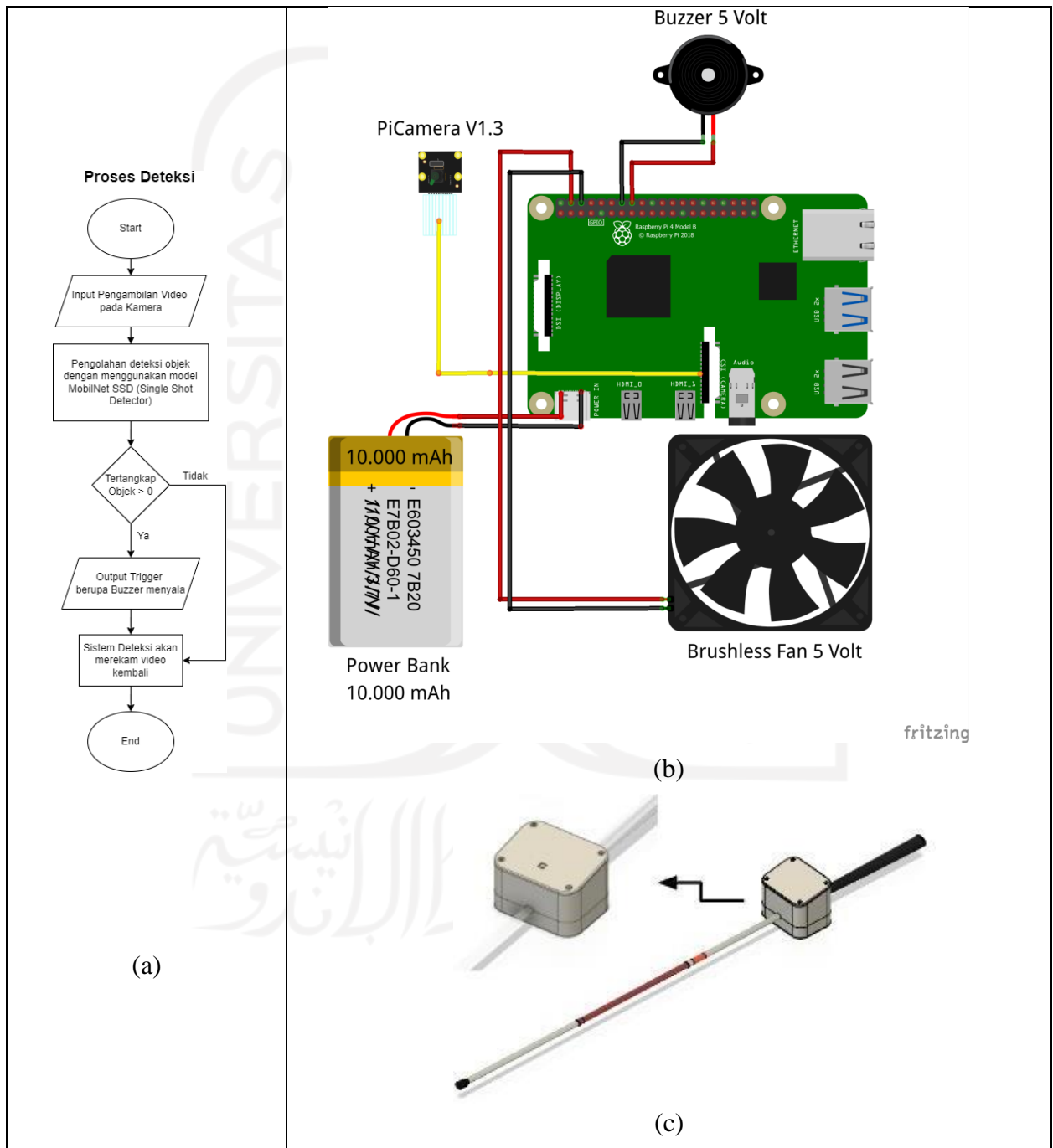
Dengan demikian, pemisahan sistem deteksi objek dengan perekaman video pada kamera bisa disesuaikan dan tidak tertinggal dengan kondisi *real-time*. Pemisahan ini dilakukan dengan fungsi FIFO (*First In First Out*) yaitu pada *input* (kamera) akan menjalankan langsung dijalankan di *frame* dan *output* (deteksi) akan dijalankan di *main thread* atau berada di lapisan atas *frame* agar nantinya tidak berjalan bersama di dalam 1 lapis *frame* karena input (kamera) akan menyesuaikan dengan frame deteksi yang menyebabkan performa deteksi yang kurang sesuai dengan *real time*

kamera. Gambar 3.7 merupakan *flowchart* proses pemisahan sistem perekaman video dengan sistem deteksinya.

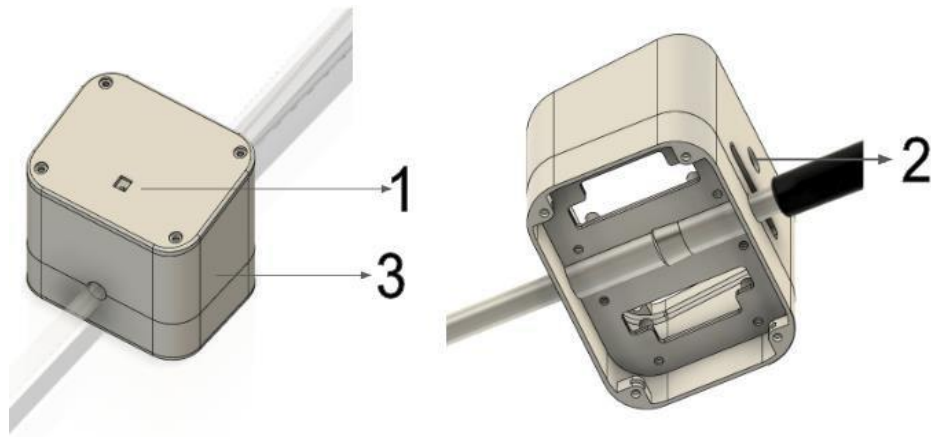


Proses Pemisahan Perekaman Video dengan Deteksi Objek

Gambar 3.7 Flowchart Pemisahan Perintah Perekaman Video dengan Deteksi Objek



Gambar 3.8 Ilustrasi usulan rancangan sistem secara umum. (a) Proses cara kerja sistem, (b) Desain Sistem Elektronis (c) Gambaran desain sistem ADOPTUKAM



Gambar 3.9 Desain 3D tampak atas dan bawah

Gambar 3.9 berisi penjelasan setiap bagian dari desain 3D *case* yang diberikan penomoran untuk mempermudah penjelasan setiap bagiannya. Berikut adalah penjelasan setiap bagian desain 3D *case*.

1. Lubang berukuran 8×8mm untuk penempatan kamera Raspberry Pi Cam V1.3 mendeteksi objek mengambil gambar manusia, kursi, dan sepeda.
2. Lubang berukuran 12mm untuk peletakan *buzzer* untuk memberikan peringatan berupa suara kepada pengguna tongkat ketika akan menabrak suatu objek di depannya.
3. Merupakan tempat Raspberry Pi 4B dan baterai diletakkan agar terhindar dari guncangan dan cuaca panas dan air sehingga meminimalisir terjadinya kerusakan.
4. Dimensi Kemasan 3D print dengan Bahan PLA dengan $P \times L \times T = 12 \times 9 \times 11$ cm

Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.2 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.2 Inventarisasi kebutuhan usulan sistem ADOPTUKAM

No	Nama Alat	Keterangan
1	Kemasan alat (<i>casing</i>)	Dibuat untuk menjadi tempat untuk melindungi komponen elektronik yang ada pada tongkat dari cuaca panas dan hujan serta agar alat terlihat lebih rapi dengan kemasan dibuat dengan bahan filamen 3D printer dengan tipe PLA.
2	Raspberry Pi 4B 4GB	Untuk <i>central processing unit</i> dengan ukuran yang kecil berupa mikroprosesor. Raspberry pi dapat digunakan untuk proses deteksi objek secara <i>real time</i> dan lebih mudah untuk dapat di bawa bersama tongkat tunanetra dibandingkan harus menggunakan komputer sebagai media pemrosesan citranya.
3	Sumber Daya (Baterai)	Baterai yang dapat dengan mudah diganti dan terpisah dari modul utama. Dalam hal ini kami cenderung mencari tipe baterai dengan tegangan kerja 5 V dan kapasitas 10.000 mAh. Ini dapat bertahan dalam jangka waktu kurang lebih 3 jam pemakaian.

No	Nama Alat	Keterangan
4	Raspberry Pi - Camera V1.3	Alat bantu berupa modul kamera yang digunakan dalam perekaman video menggunakan kamera Raspberry Pi - Cam yang telah terintegrasi dengan raspberry pi dengan menghubungkan kabel FPC langsung pada port CSI dengan resolusi 5MP yang cukup untuk pengambilan citra dengan harga yang masih terjangkau.
5	Buzzer	Modul Buzzer berperan sebagai aktuator yang berfungsi untuk pemberian peringatan pada pengguna tunanetra melalui bunyi.
6	Tongkat Tunanetra	Tongkat dengan panjang 110 cm - 120 cm yang merupakan standar ukuran panjang tongkat umur dewasa dengan berbahan Aluminium Galvanis yang membuat tongkat menjadi ringan. Tongkat ini sangat ringan dengan fitur 4 kelipatan

3.2 Metode Uji Coba dan Pengujian Usulan Rancangan Sistem

Setelah tahap pengerjaan keseluruhan alat selesai dibuat, kemudian dilanjutkan ke tahap pengujian alat atau *prototype* yang telah dibuat untuk mengetahui performa dan sebagai evaluasi bila terjadinya suatu kesalahan. Proses pengujian meliputi beberapa tahap yaitu:

1. Percobaan dilaksanakan di dalam ruangan pada Laboratorium Sistem Telekomunikasi dan Pengolahan Sinyal Teknik Elektro Universitas Islam Indonesia.
2. Mempersiapkan objek yang akan dideteksi dan dihitung yaitu berupa manusia, kursi, dan sepeda.
3. Jumlah objek yang diambil mengikuti jumlah aktual yaitu dengan objek sebanyak 4 manusia, 3 kursi, dan 1 sepeda.
4. Percobaan dilakukan dengan menghitung objek sesuai dengan jumlah aktual dan jumlah yang terdeteksi oleh sistem.
5. Peletakkan objek dari jarak 1 m hingga 5 m di depan tongkat yang nanti akan ditangkap oleh kamera.

3.2.1 Perhitungan Fungsionalitas Alat (Metode *Confusion Matrix*)

Setelah melaksanakan tahapan uji coba, tahap selanjutnya adalah melakukan evaluasi dengan menggunakan *Confusion Matrix*. Metode ini umumnya digunakan untuk mengukur fungsionalitas permasalahan model *Deep Learning* yang dapat mengukur nilai akurasi, presisi, *recall*, dan *F1 - Score*.

Tabel 3.3 Variabel Perhitungan *Confusion Matrix*

Tabel Variabel Kondisi		Kondisi Aktual	
		Ada	Tidak Ada
Hasil Deteksi	Terdeteksi	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>

Tabel Variabel Kondisi		Kondisi Aktual	
		Ada	Tidak Ada
	Tidak Terdeteksi	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Keterangan:

1. *True Positive* (TP) = Kondisi ketika objek berhasil dideteksi oleh sistem dalam hal ini yaitu Manusia, Kursi, Sepeda.
2. *False Positive* (FP) = Kondisi ketika sistem menganggap objek lain sebagai objek yang dituju.
3. *False Negative* (FN) = Kondisi ketika objek tidak terdeteksi oleh sistem
4. *True Negative* (TN) = Kondisi ketika sistem mengabaikan objek selain Manusia, Kursi, dan Sepeda.

Setelah mendapatkan empat parameter yang dibutuhkan untuk mengukur Akurasi, Presisi, *Recall*, dan *F1 - Score* alat dengan menggunakan Rumus berikut:

$$\text{Akurasi} = \frac{(TP + TN)}{(TP + FN + FP)} \times 100\% \quad (3.1)$$

$$\text{Presisi} = \frac{(TP)}{(TP + FP)} \times 100\% \quad (3.2)$$

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \times 100\% \quad (3.3)$$

$$\text{F1 - Score} = \frac{2 \times (\text{Recall} \times \text{Presisi})}{\text{Recall} + \text{Presisi}} \times 100\% \quad (3.4)$$

3.2.2 Performance Metric

3.2.2.1 Task Success Rate

Untuk mengetahui tingkat keberhasilan *aktuator* atau *buzzer* memberikan peringatan ketika berhasil mendeteksi objek apakah dalam kondisi bunyi mengeluarkan suara atau tidak. Parameter *true selection* menunjukkan bahwa hasil pengujian sistem berhasil mengeluarkan atau jumlah keberhasilan bunyi atau peringatan ketika mendeteksi objek. Pengukuran dapat dilakukan dengan perhitungan sebagai berikut:

$$\text{Task Success Rate} = \frac{\text{number of true selection}}{\text{number of total task}} \times 100\% \quad (3.5)$$

3.2.2.2 System Usability Scale (SUS)

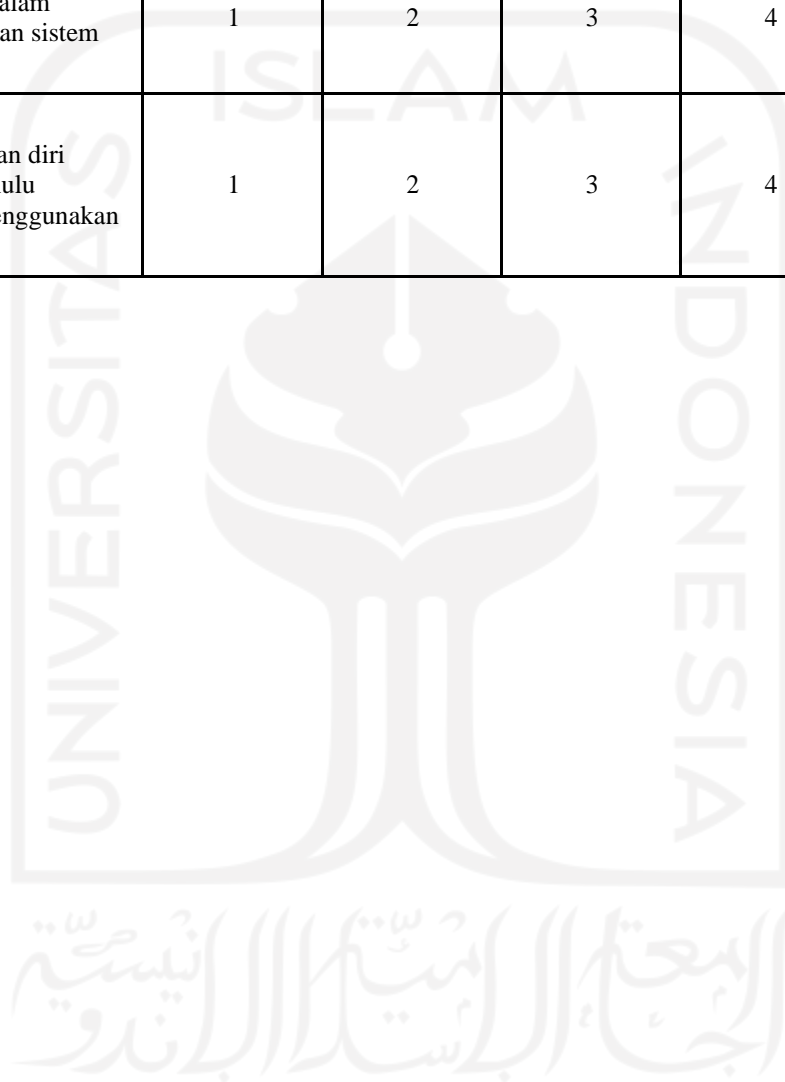
SUS digunakan sebagai Uji Performansi karena alat merupakan salah satu langkah solusi untuk membantu penyandang tunanetra melakukan mobilitas dan meminimalisir kecelakaan yang terjadi, maka nilai kenyamanan dan fungsionalitas diperlukan langsung oleh penyandang disabilitas. Terdapat 10 buah pertanyaan dengan rentang nilai 1 - 5 untuk tiap aspek seperti pada Tabel 3.4 dengan aturan menghitung perhitungan skor SUS:

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

Tabel 3.4 Pertanyaan SUS

No.	Pertanyaan	Jawaban				
		STS (Sangat Tidak Setuju)	TS (Tidak Setuju)	N (Netral)	S (Setuju)	SS (Sangat Setuju)
1.	Saya berpikir akan menggunakan sistem ini lagi	1	2	3	4	5
2.	Saya merasa sistem ini rumit untuk digunakan	1	2	3	4	5
3.	Saya merasa sistem ini mudah digunakan	1	2	3	4	5
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini	1	2	3	4	5
5.	Saya merasa fitur - fitur sistem ini berjalan dengan semestinya	1	2	3	4	5
6.	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini	1	2	3	4	5
7.	Saya merasa orang lain akan memahami cara menggunakan sistem	1	2	3	4	5

No.	Pertanyaan	Jawaban				
		STS (Sangat Tidak Setuju)	TS (Tidak Setuju)	N (Netral)	S (Setuju)	SS (Sangat Setuju)
	ini dengan cepat					
8.	Saya merasa sistem ini membingungkan	1	2	3	4	5
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	1	2	3	4	5
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini	1	2	3	4	5

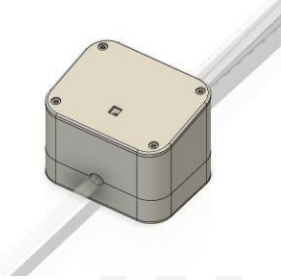
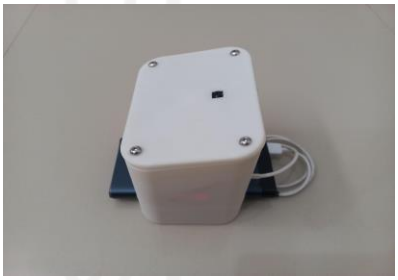




BAB 4 : Hasil Perancangan Sistem

4.1 Kesesuaian Usulan dan Hasil Perancangan Sistem

Realisasi usulan yang dirancang mengalami perubahan-perubahan dengan perencanaan usulan yang telah disepakati sebelumnya oleh tim penulis. Terjadinya perubahan spesifikasi usulan dan hasil perancangan sistem atau realisasi akibat usulan atau rencana awal yang disepakati tidak menunjukkan hasil yang optimal. *Head-to-head comparison* dari rencana awal dan realisasi ditunjukkan pada Tabel 4.1.

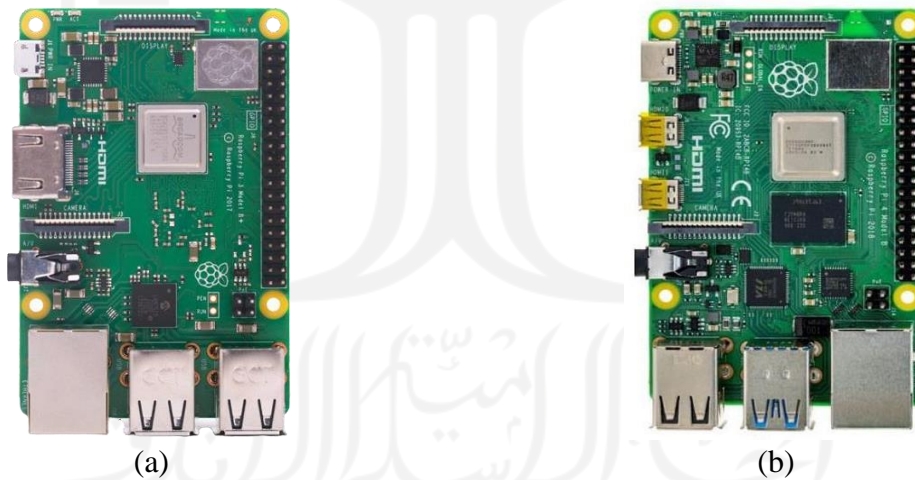
Tabel 4.1 Perbandingan usulan dan hasil perancangan sistem

No	Spesifikasi	Usulan	Realisasi
1	Baterai	5V 10.000 mAh	5V 10.000 mAh
2	Panjang Tongkat	110 cm	110 cm
3	Berat Tongkat	400 gr	1 kg
4	Bahan Tongkat	Aluminium Galvanis	Aluminium Galvanis
5	Kemasan (<i>Casing</i>)	Akrilik	Filament 3D printer (PLA)
6	Sensor Kamera	Raspberry Pi CAM V1.3	Raspberry Pi CAM V1.3
7	Jarak Pendeteksian	1 - 5 m	1 - 5 m
8	Desain Kemasan		
9	Mikroprosesor	Raspberry Pi 3B+ 1GB	Raspberry Pi 4B 4GB
10	Metode	YOLOTinyV4	MobileNet SSD Deploy
11	Desain Tongkat		

Sistem yang dibuat awalnya menggunakan Raspberry Pi 3B+ 1GB. Namun setelah dilakukan uji coba dan validasi memberikan hasil pendeteksian objek pada sistem sangat lambat. Hal ini terjadi karena spesifikasi dari Raspberry Pi 3B+ dengan Raspberry Pi 4B berbeda dari kapasitas RAM yang dimiliki dengan kecepatan CPU juga lebih cepat. Kemudian *processor* yang digunakan terdapat pembaharuan dalam segi performa yang nantinya dapat memberikan respon objek deteksi yang berbeda dari Raspberry Pi 3B+ dari hasil pengujian didapatkan nilai FPS yang didapatkan pada Raspberry Pi 4B mendeteksi objek lebih baik dibandingkan Raspberry Pi 3B+. Berikut perbedaan spesifikasi yang ditunjukkan pada Tabel 4.2 dan Gambar 4.1.

Tabel 4.2 Spesifikasi Usulan dan Realisasi yang Digunakan

Spesifikasi Usulan (Raspberry Pi 3 Model B)		Spesifikasi Realisasi (Raspberry Pi 4 Model B)	
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz	Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	1GB LPDDR2 SDRAM	RAM	4GB LPDDR4 with on-die ECC
Environment	Temperature 0 – 50°C	Environment	Temperature 0 - 50°C
GPIO	Extended 40-pin GPIO header	GPIO	Standard 40-pin GPIO header



Gambar 4.1 (a) Sistem usulan (Sumber: [Buy a Raspberry Pi 3 Model B+ – Raspberry Pi](#)) (b) realisasi yang digunakan (Sumber: [Buy a Raspberry Pi 4 Model B – Raspberry Pi](#))

4.2 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya

Dalam pengerjaan sistem, perencanaan dan manajemen waktu berperan penting dalam mengetahui kapan proyek akan dimulai serta selesainya proyek tersebut. Hal tersebut juga untuk menjaga kesesuaian antara usulan dan realisasinya. Pada kegiatan pengerjaan ADOPTUKAM perbandingan *Head-to-head* antara perencanaan dan realisasi *timeline* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Kesesuaian antara usulan dan realisasi *timeline* pengerjaan Tugas Akhir 2

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
1	Pengumpulan proposal TA1 / <i>Capstone Design</i> dan Seminar	Januari – Februari	Januari – Februari
2	Pembelian Komponen	Februari – Maret	April – Mei
3	Perancangan Sistem	Februari – April	Mei – Juni
4	Uji Coba Performa	Juni – Juli	Juni – Juli
5	Uji Coba Fungsionalitas	Juni – Juli	Juni – Juli
6	Pengumpulan Laporan akhir	Juni	Juli
7	Expo	Juli	Agustus

Dalam perencanaan sistem manajemen tim, perancangan sistem ditargetkan untuk dikerjakan dalam bulan Februari hingga April untuk mengikuti program percepatan. Namun terjadi kemunduran pada *timeline* yang telah diusulkan oleh tim penulis akibat terkendala pada pendeteksian yang masih mengalami nilai FPS (*frame per second*) yang rendah sehingga pendeteksian mengalami *delay* selama 15 detik saat mendeteksi objek kemudian perubahan jadwal dalam pengumpulan laporan akhir dan expo yang ditetapkan oleh prodi sehingga mempengaruhi perubahan *timeline* usulan dan Rencana anggaran biaya (RAB) yang ditunjukkan pada Tabel 4.4.

Tabel 4.4 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi

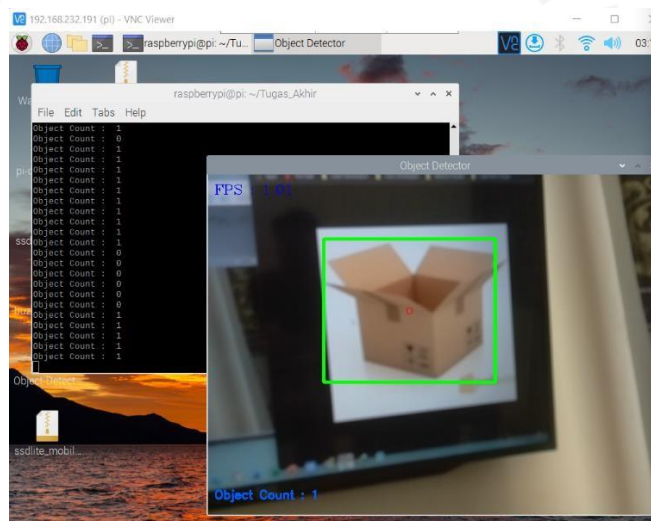
No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	Raspberry Pi Camera V1.3 5MP	1 pcs	Rp. 300.000,-	1 pcs	Rp. 350.000,-
2	Raspberry Pi 4B 4GB	1 pcs	Rp. 2.790.000,-	1 pcs	Rp. 2.790.000,-
3	Baterai Power Bank 10.000 mAH	1 pcs	Rp. 80.000,-	1 pcs	Rp. 80.000,-
4	Tongkat Tunanetra	1 pcs	Rp. 65.000,-	1 pcs	Rp. 73.500,-
5	Kemasan 3D print	256 gr	Rp. 400.000,-	256 gr	Rp. 400.000,-
6	Buzzer	1 pcs	Rp. 3.000,-	2 pcs	Rp. 6.000,-
7	Micro SD 64 GB	-	-	1 pcs	Rp. 96.000,-
7	Kabel Jumper	2 pcs	Rp. 2.000,-	4 pcs	Rp. 4.000,-
8	Kabel FPC kamera	1 pcs	Rp. 8.000,-	2 pcs	Rp. 16.000,-
9	Mur 2mm	12 pcs	-	12 pcs	-
Total		Rp. 3.648.000,-		Rp. 3.815.500,-	

4.3 Analisis dan Pembahasan Kesesuaian antara Perencanaan dan Realisasi

Dalam proses perencanaan dan realisasinya mengalami beberapa perubahan untuk mencapai hasil yang terbaik pada sistem. Tentunya perubahan tersebut didasari oleh pertimbangan dengan melakukan observasi kembali melalui referensi yang kredibel dan hasil diskusi antar tim

dan Dosen pembimbing adalah sebagai Berikut:

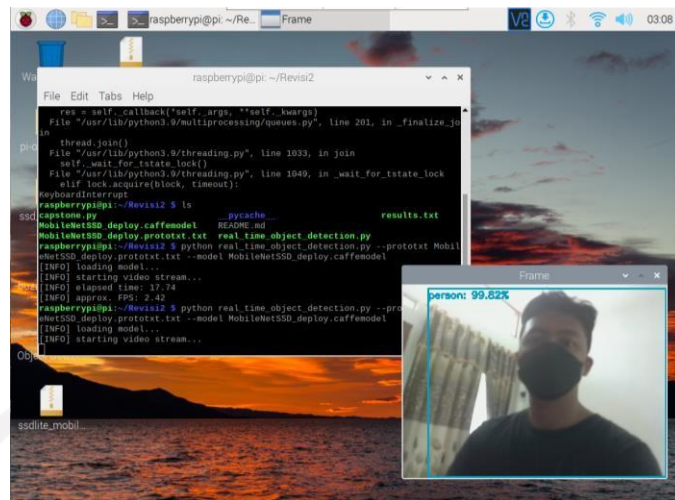
1. Kesesuaian antara perencanaan dan realisasi sebesar 60% berdasarkan Tabel 4.1 hingga Tabel 4.4.
2. Hasil dari sistem deteksi dan pembuatan alat yang dihasilkan mencapai target yang diinginkan dengan menyesuaikan spesifikasi yang diusulkan namun terdapat beberapa perubahan dari sisi spesifikasi hardware yang sebelumnya dijelaskan bahwa perubahan pada pemrosesan menggunakan Raspberry Pi 4B.
3. Pembahasan ketidaksesuaian selanjutnya terletak dalam model *deep learning* yang digunakan untuk sistem deteksi objek, mulanya dilakukan training objek untuk model *deep learning* YOLOv4-tiny (*You Only Look Once*). Didapatkan nilai maksimal pada FPS pada deteksi sebesar 1.0 FPS pada Gambar 4.2 berikut.



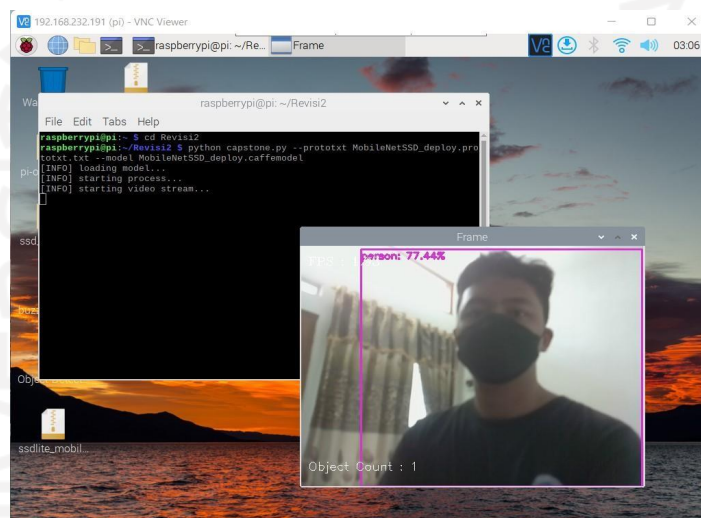
Gambar 4.2 Hasil percobaan YoloV4tiny

Sehingga dilakukan perubahan dan observasi mengenai model *deep learning* menggunakan model MobileNet dengan metode SSD (*Single Shot Multibox Detector*). Didapatkan hasil maksimal FPS deteksi sebesar 3.0 FPS walaupun terdapat peningkatan dari jumlah FPS dalam segi respon deteksi masih kurang baik dalam penggunaan *real-time* perekaman video dengan melakukan metode pemisahan untuk meningkatkan respon *real-time* saat pendeteksian

yang ditunjukkan pada Gambar 4.3 dan 4.4.



Gambar 4.3 Percobaan Model Deep Learning MobileNet SSD Untuk Sistem Deteksi



Gambar 4.4 Percobaan Model Deep Learning MobileNET SSD + FIFO

الجامعة الإسلامية
الاستاذ الدكتور

BAB 5 : Implementasi Sistem dan Analisis

5.1 Hasil dan Analisis Implementasi

Hasil implementasi didapatkan sistem yang telah dirancang penulis telah dilakukan dua buah pengujian. Pertama dengan melakukan Uji Fungsionalitas, dalam pengujian ini dilakukan untuk mengetahui performa pendeteksian objek tiap jarak yaitu 1 – 5 m dengan Metode *Confusion Matrix* sehingga didapatkan nilai Akurasi, Presisi, *Recall*, *F1- Score*. Kemudian Uji performansi (*performance metric*) menggunakan *Task Success Rate* untuk mengetahui tingkat keberhasilan yang dihasilkan oleh *buzzer* berbunyi atau tidak ketika berhasil mendeteksi objek. Dan menggunakan *System Usability Scale (SUS)*.

5.1.1 Metode Confusion Matrix

Pengujian sistem dilakukan dengan menguji fungsionalitas serta membatasi jumlah objek pendeteksian sebanyak 3 buah yaitu manusia, kursi, dan sepeda yang dilakukan di dalam ruangan dengan intensitas cahaya sebesar 421 lux yang didapatkan menggunakan *luxmeter* sebanyak 5 kali percobaan.



Gambar 5.1 *Luxmeter*

Selanjutnya, pengambilan data dilakukan di Laboratorium Pengolahan Sinyal Telekomunikasi (PST) Elektro Universitas Islam Tapi (UII) dengan meletakkan objek kursi, manusia, dan sepeda untuk pengujian pendeteksian dengan rentang jarak 1 – 5 meter dengan menggunakan *meteran* manual sebagai acuan seperti pada Gambar 5.2 berikut.



Gambar 5.2 Proses Pengambilan Data Percobaan

Pada pengujian, untuk membandingkan hasil sistem mendeteksi dengan jumlah aktual dan sistem deteksi dengan jumlah aktual 4 Manusia, 1 buah sepeda, dan 3 buah kursi dengan jumlah yang sama pada tiap jaraknya dari jarak 1 meter hingga 5 meter. Setelah dilaksanakan sebanyak 5 kali percobaan. Didapatkan nilai keseluruhan 4 parameter *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Apabila jumlah aktual objek dan sistem mendeteksi jumlah objek sama maka didapatkan nilai *True Positive* (TP). Jika pada jumlah aktual terdapat 1 objek namun pada sistem mendeteksi 2 objek, maka terjadi data negatif yang menjadi positif karena objek yang bukan seharusnya dideteksi tetapi terdeteksi oleh sistem sehingga didapatkan nilai *False Positive* (FP), sedangkan *False negative* (FN) adalah ketika sistem tidak mampu mendeteksi objek yang akan dideteksi. Akurasi yang baik menunjukkan bahwa nilai terdeteksi dan aktual sangat dekat, Presisi menunjukkan tingkat ketepatan antara informasi yang diminta pengguna dengan jawaban yang diberikan oleh sistem. *Recall* menunjukkan bagaimana keberhasilan sistem dalam menemukan kembali sebuah informasi tersebut dengan *F1-Score* adalah rata rata nilai harmonik yang dihasilkan dari Presisi dan *Recall*.

5.1.1.1 Percobaan Deteksi Manusia

Hasil pengukuran objek manusia pada tiap jarak 1 hingga 5 meter yang ditunjukkan pada Tabel 5.1 berikut. maka langkah selanjutnya adalah mencari nilai Akurasi, Presisi, *Recall* dan *F1-Score* menggunakan persamaan (3.1), (3.2), (3.3), dan (3.4) sebelumnya.

Tabel 5.1 Akumulasi Hasil Percobaan Deteksi Manusia

Total Objek Manusia = 100	TP	TN	FP	FN
Jarak (m)				
1	14	-	-	6
2	18	-	-	2
3	17	-	-	3
4	19	-	-	1
5	19	-	2	1

$$\begin{aligned}
 \text{Akurasi} &= \frac{(TP + TN)}{(TP + FN + FP)} \times 100\% \\
 &= \frac{(14 + 0)}{(14 + 6 + 0)} \times 100\% \\
 &= 70\%
 \end{aligned}$$

$$\begin{aligned}
 \text{Presisi} &= \frac{(TP)}{(TP + FP)} \times 100\% \\
 &= \frac{14}{(14 + 0)} \times 100\% \\
 &= 100\%
 \end{aligned}$$

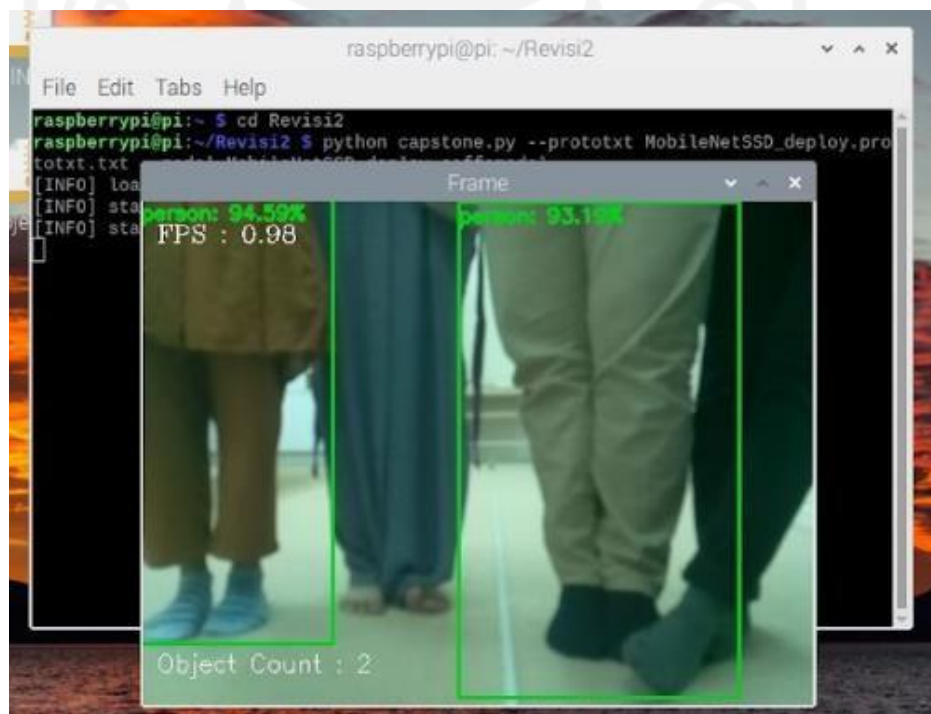
$$\begin{aligned}
 \text{Recall} &= \frac{(TP)}{(TP + FN)} \times 100\% \\
 &= \frac{14}{(14 + 6)} \times 100\% \\
 &= 70\%
 \end{aligned}$$

$$\text{F1 - Score} = \frac{2 \times (\text{Recall} \times \text{Presisi})}{\text{Recall} + \text{Presisi}} \times 100\%$$

$$= \frac{2 \times (0.7 \times 1)}{(0.7 + 1)} \times 100\%$$

$$= 82\%$$

Hasil pengukuran jarak 1 meter mendapatkan hasil akurasi sebesar 70%, dengan presisi sebesar 100%, *recall* 70%, dan *F1-score* 82%. Nilai tersebut baik dikarenakan manusia merupakan objek yang paling besar diantara ketiga objek lainnya. Posisi sudut kamera saat pengambilan gambar mempengaruhi pendeteksian manusia yang hanya berupa kaki manusia. Apabila melihat dari hasil pendeteksian jarak 1 m, kamera tampak sulit mendeteksi sebuah objek manusia dengan warna gelap tetapi tidak pada jarak 3 – 5 meter. Hal ini pengaruh resolusi kamera yang dapat mempengaruhi hal tersebut karena kualitas gambar yang ditangkap merupakan salah satu faktor yang menentukan keberhasilan proses deteksi [15]. Objek manusia yang tidak masuk kedalam *frame* mengakibatkan terjadi *False Negative* (FN). Sehingga, walaupun sudah ada objek manusia di depannya namun tidak berhasil menangkap gambar keseluruhan seperti yang ditampilkan pada Gambar 5.3.



Gambar 5.3 Hasil Penangkapan Objek Manusia jarak 1 meter

5.1.1.2 Percobaan Deteksi Sepeda

Tabel 5.2 Hasil Percobaan Deteksi Sepeda

Total Objek Sepeda = 25	TP	TN	FP	FN
Jarak (m)				
1	5	-	4	-
2	3	-	-	2
3	5	-	1	-
4	5	-	1	-
5	5	-	2	-

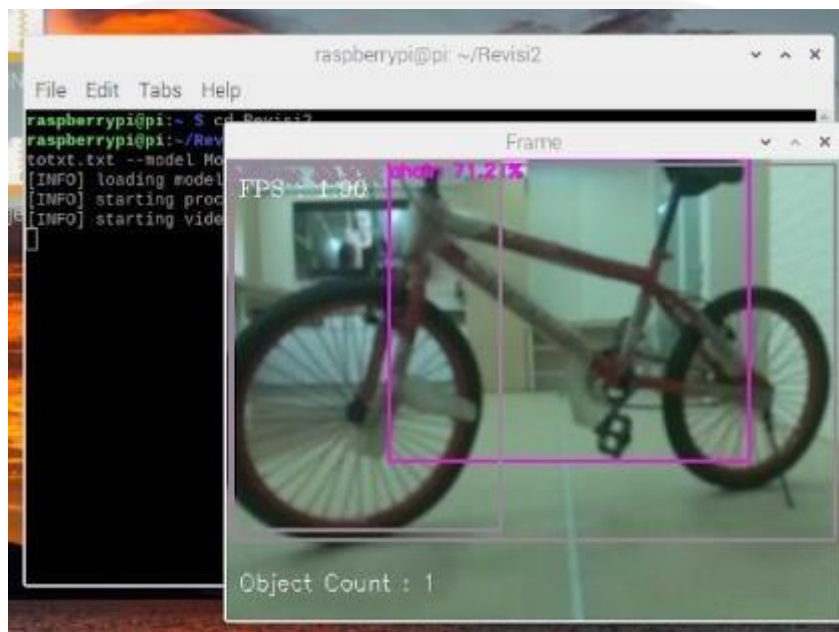
$$\begin{aligned} \text{Akurasi} &= \frac{(TP + TN)}{(TP + FN + FP)} \times 100\% \\ &= \frac{5 + 0}{5 + 4 + 0} \times 100\% \\ &= 55\% \end{aligned}$$

$$\begin{aligned} \text{Presisi} &= \frac{(TP)}{(TP + FP)} \times 100\% \\ &= \frac{5}{(4 + 0)} \times 100\% \\ &= 55\% \end{aligned}$$

$$\begin{aligned} \text{Recall} &= \frac{(TP)}{(TP + FN)} \times 100\% \\ &= \frac{5}{(5 + 0)} \times 100\% \\ &= 100\% \end{aligned}$$

$$\begin{aligned} \text{F1 - Score} &= \frac{2 \times (\text{Recall} \times \text{Presisi})}{\text{Recall} + \text{Presisi}} \times 100\% \\ &= \frac{2 \times (0.55 \times 1)}{(0.55 + 1)} \times 100\% \\ &= 71\% \end{aligned}$$

Berdasarkan perhitungan pendeteksian sepeda, pengambilan gambar terdapat beberapa sepeda yang tidak berhasil terdeteksi pada jarak 1 meter melainkan mendeteksi objek lainnya berupa kursi yang ada di sepeda sehingga pembacaan akurasi pada jarak 1 meter cukup kecil yaitu 55%. Hal ini bisa terjadi karena kamera mengalami kesulitan dalam mendeteksi sepeda karena terlalu besar objeknya sehingga tidak terdeteksi, kurangnya gambar sepeda dengan posisi dari jarak dekat yang di-*training* pada model tersebut sehingga metode *machine learning* kurang bagus dalam mendeteksi objek tersebut. Pengambilan gambar sepeda untuk jarak lainnya pada Gambar 5.4 berikut.



Gambar 5.4 Hasil Penangkapan Objek Sepeda

5.1.1.3 Percobaan Deteksi Kursi

Tabel 5.3 Hasil Percobaan Deteksi Kursi

Total Objek Kursi = 75	TP	TN	FP	FN
Jarak (m)				
1	15	-	3	-
2	15	-	1	-
3	14	-	1	1
4	15	-	1	-
5	15	-	1	-

$$Akurasi = \frac{(TP + TN)}{(TP + FN + FP)} \times 100\%$$

$$= \frac{15 + 0}{15 + 3 + 0} \times 100\%$$

$$= 83\%$$

$$Presisi = \frac{(TP)}{(TP + FP)} \times 100\%$$

$$= \frac{15}{(15 + 3)} \times 100\%$$

$$= 83\%$$

$$Recall = \frac{(TP)}{(TP + FN)} \times 100\%$$

$$= \frac{15}{(15 + 0)}$$

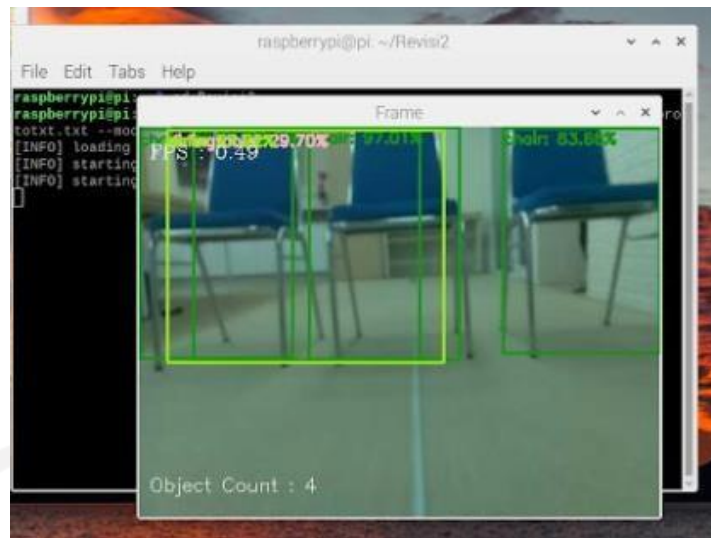
$$= 100\%$$

$$F1 - Score = \frac{2 \times (Recall \times Presisi)}{Recall + Presisi} \times 100\%$$

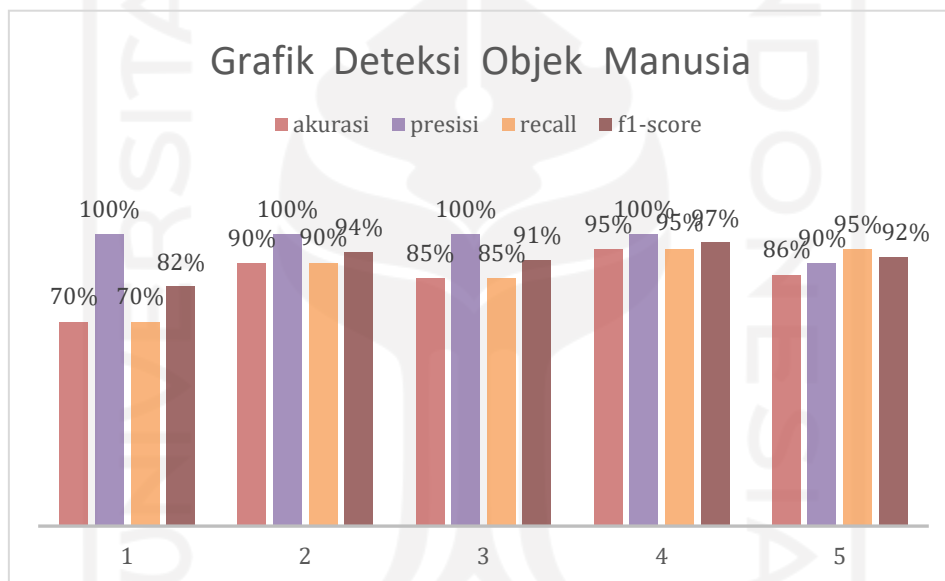
$$= \frac{2 \times (0.83 \times 1)}{(0.83 + 1)} \times 100\%$$

$$= 90\%$$

Berdasarkan pengukuran pengujian deteksi objek kursi pada jarak 1 m, didapatkan nilai akurasi sebesar 83%, presisi 83%, *recall* 100%, dan *F1 - Score* sebesar 90%. dari kelima percobaan didapatkan bahwa pada jarak 1 meter sistem mendeteksi objek selain kursi sebanyak 3 kali sehingga didapatkan nilai *False positive* (FP) sebanyak 3 buah. Gambar 5.5 merupakan salah satu contoh *False Positive* terjadi yaitu 3 kursi terdeteksi menjadi 4 kursi dengan nilai fps saat pendeteksian sebesar 0.49 FPS hal ini bisa terjadi akibat karena bentuk kursi pada percobaan kurang beragam.

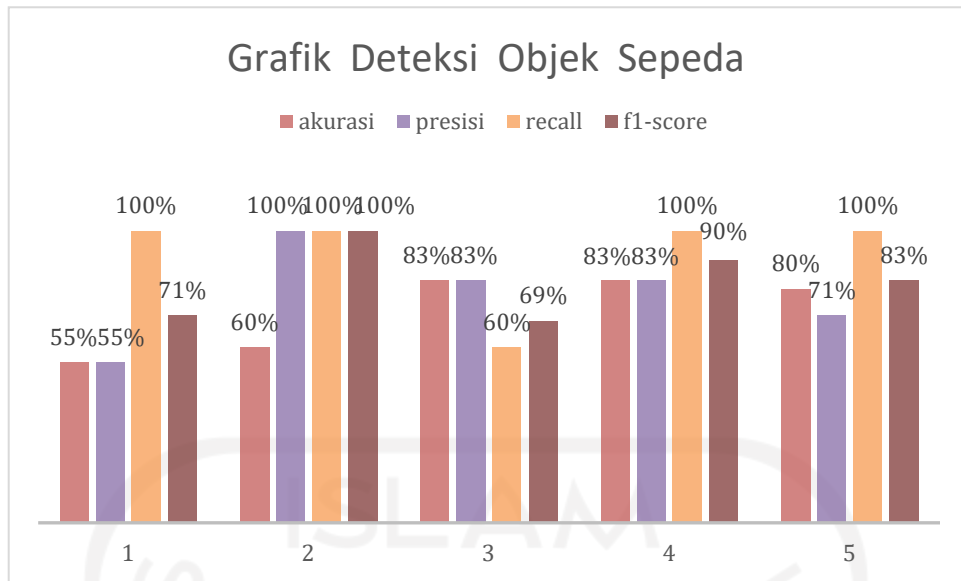


Gambar 5.5 Hasil Penangkapan Objek Kursi



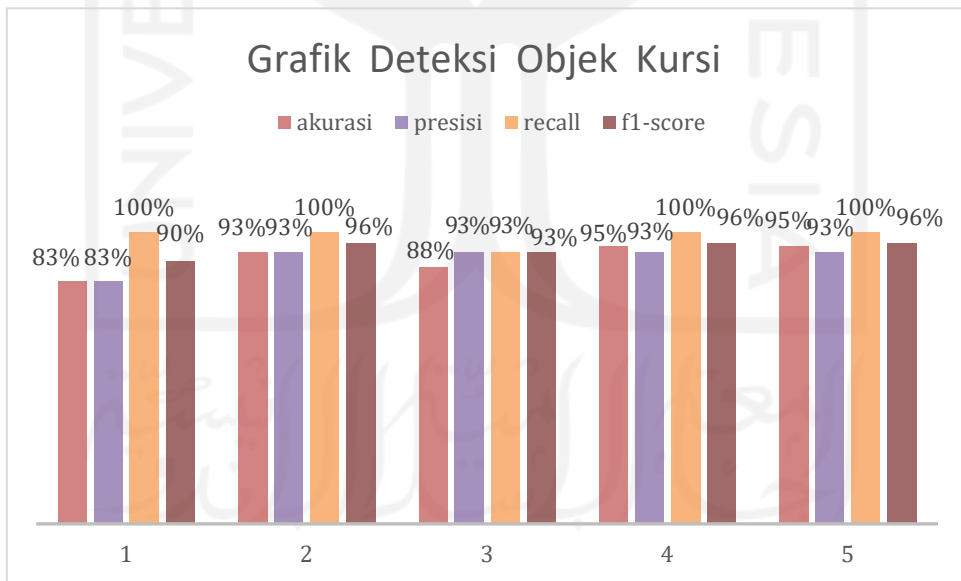
Gambar 5.6 Grafik Deteksi Objek Manusia

Pada hasil jarak pendeteksian 1 hingga 5 meter pada objek manusia, nilai keempat parameter yang dihasilkan sistem pada jarak 1 meter lebih rendah dibandingkan dengan ke empat jarak lainnya. Hal ini karena pada jarak 2 hingga 4 meter kamera menangkap tangkapan gambar bentuk manusia lebih baik dengan jarak 1 meter akibat tidak dapat mendeteksi dalam kondisi tertentu. Dengan nilai presisi menghasilkan 100% dari ke empat jarak.



Gambar 5.7 Grafik Deteksi Objek

Pada hasil jarak pendeteksian 1 hingga 5 meter pada objek sepeda, nilai keempat model pada jarak 1 meter lebih rendah dibandingkan dengan ke empat jarak lainnya. Hal ini terjadi karena objek tersebut sulit mengenali sepeda dalam posisi dekat dengan kamera karena tidak dapat mengenali bentuk sepeda yang telah dilatih sehingga tidak dapat melakukan pendeteksian apabila tidak mengatur posisi sepeda pada saat pengujian dengan baik.



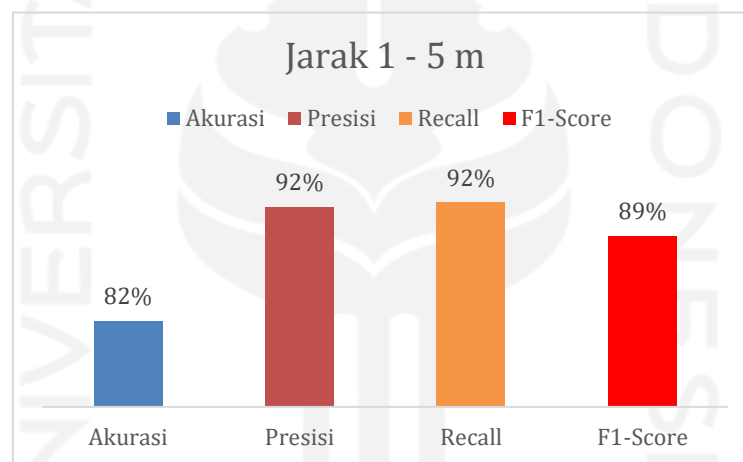
Gambar 5.8 Grafik Deteksi Objek Kursi

Berdasarkan hasil pengujian yang dilakukan, jarak efektif terjadi di 2 m hingga 4 meter. Nilai akurasi, presisi, *recall* dan *f1-score* mengalami penurunan dibandingkan setelah jarak 4 meter. Hal ini tentu sesuai dengan perancangan sistem karena objek akan semakin kecil terlihat serta dari kualitas kamera sudah tidak mampu mengenali objek tersebut. Maka jarak tersebut untuk pemberian peringatan pada tongkat kurang begitu efektif karena masih berada pada jarak aman

sebelum menabrak objek di depannya. Nilai pendeteksian tinggi karena ketiga objek yang diuji merupakan objek yang bentuknya besar sehingga mudah terdeteksi oleh sistem.

Tabel 5.4 Nilai Rata-rata Objek

Objek				
Jarak (m)	Akurasi	Presisi	Recall	F1-score
1	69%	94%	90%	81%
2	81%	97%	96%	96%
3	83%	92%	79%	84%
4	91%	92%	98%	95%
5	87%	84%	98%	91%



Gambar 5.6 Grafik nilai jarak 1 - 5 m

Berdasarkan nilai rata-rata yang tertera pada Tabel 5.4, maka didapatkan hasil nilai rata-rata pendeteksian dalam rentang 1 – 5 m pada Gambar 5.9. Nilai akurasi jarak 1 sampai 5 meter sebesar 82.2% dan nilai presisi sebesar 92%, *recall* 92%, dan *F1-score* 89%. Hal ini dikarenakan posisi kamera pada tongkat menangkap gambar objek lebih baik berada di rentang jarak tersebut karena objek masuk ke dalam *frame* sehingga jenis objek yang tertangkap kamera lebih jelas. Pada dasarnya bukan jarak saja yang menjadi faktor keberhasilan sistem, hasil yang didapatkan juga karena faktor intensitas cahaya yang bagus. Hal ini telah dibuktikan juga oleh penelitian sebelumnya di mana semakin rendah intensitas cahaya maka semakin sulit bagi sistem untuk mendeteksi objek [16]. Varian objek yang diuji memberikan nilai yang tinggi karena variasi objeknya belum beragam dan jumlah pendeteksian sedikit. Tentunya nilai akan mengalami penurunan apabila objek yang dideteksi heterogen atau bermacam dan dengan berbagai macam skenario.

5.1.2 Task Success Rate

Tabel 5.5 Hasil Pengujian *buzzer*

Jarak (m)	Kondisi Buzzer (Bunyi / Tidak)
1	Bunyi
2	Bunyi
3	Bunyi
4	Bunyi
5	Bunyi

$$\text{Task Success Rate} = \frac{\text{number of true selection}}{\text{number of total task}} \times 100\%$$

$$\text{Task Success Rate} = \frac{75}{75} \times 100\%$$

$$\text{Task Success Rate} = 100\%$$

Hasil percobaan *Task Success Rate* untuk mengetahui nilai persentase kinerja *buzzer* mengeluarkan bunyi ketika kamera berhasil mendeteksi gambar sebesar 100% sehingga uji performansi pada alat berjalan dengan semestinya. bunyi dari *buzzer* ini berguna sebagai peringatan untuk penyandang tunanetra agar terhindari dari terjadinya tabrakan dengan objek di depannya.

5.1.3 System Usability Scale (SUS)

Pengujian kebergunaan alat dilakukan dengan salah satu penyandang disabilitas yang merupakan siswa SMALB Klaten di daerah Karanganyam bernama Rizky. Tabel 5.6 menampilkan hasil SUS yang didapatkan hasil sebagai berikut.

Tabel 5.6 Hasil Pengisian SUS

No.	Pertanyaan	Jawaban				
		STS (Sangat Tidak Setuju)	TS (Tidak Setuju)	N (Netral)	S (Setuju)	SS (Sangat Setuju)
1.	Saya berpikir akan menggunakan sistem ini lagi				4	
2.	Saya merasa sistem ini rumit untuk digunakan			3		
3.	Saya merasa sistem ini mudah digunakan					5
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini				4	
5.	Saya merasa fitur - fitur sistem ini berjalan dengan semestinya					5
6.	Saya merasa ada banyak hal yang tidak konsisten pada sistem ini	1				
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat					5
8.	Saya merasa sistem ini membingungkan	1				
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini			3		
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini				4	
Total		35				

$$\begin{aligned} \text{Skor SUS} &= 2.5 \times \text{Total} \\ &= 87.5 \end{aligned}$$

Didapatkan Skor SUS berada pada nilai >68 melalui satu orang responden penyandang tunanetra yang menunjukkan bahwa sistem memiliki fungsi dan kegunaannya diatas rata-rata. berdasarkan hasil dari SUS dapat disimpulkan bahwa Tingkat ADOPTUKAM mudah digunakan dan mudah dipahami dengan cepat sehingga pengguna mampu menggunakan sistem ini dengan baik dengan sedikit pembiasaan terlebih dahulu.



Gambar 5.10 Pengambilan Data SUS

5.2 Pengalaman Pengguna

Pada uji performansi yang telah dilakukan oleh tim penulis dan pengguna, sistem pendeteksian ADOPTUKAM sudah berjalan cukup baik ketika menangkap citra objek yang ada di depannya. Terdapat beberapa aspek capaian dan aksi terhadap sistem yang telah dibuat dengan pengguna pada Tabel 5.7

Tabel 5.7 Pengalaman Pengguna

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
1	Fungsi	Fungsi sebagai pendeteksian sistem berjalan dengan baik,.	Memaksimalkan pendeteksian agar tidak terjadinya delay pada saat pengiriman ke aktuator dari bagian pemrosesan.
2	Kemudahan	Pengoperasian tongkat cukup mudah	Membuat <i>on/off</i> langsung tanpa harus terhubung dengan VNC Viewer.
3	Keamanan	Keamanan sudah cukup baik untuk pendeteksian namun belum terlalu bisa menanggulangi permasalahan <i>real</i> lainnya seperti deteksi lubang, dan api.	Menambahkan sistem pendeteksian yang lebih banyak lagi untuk dapat mendeteksi api, dan lubang.
4	Peringatan	Fitur peringatan berupa suara <i>buzzer</i>	Penambahan modul getar atau <i>headset</i> yang dapat langsung digunakan pengguna agar lebih peka

5.3 Dampak Implementasi Sistem

5.3.1 Teknologi/Inovasi

Sistem ADOPTUKAM untuk penyandang tunanetra yang telah dibuat memiliki beberapa keunggulan terhadap sistem yang telah dirancang oleh peneliti terdahulu. Dengan ketiga sistem

yang dibuat telah berhasil mendeteksi benda secara *real-time* seperti sistem yang dibuat oleh S. Fuady, dkk dengan judul “*Deteksi Objek Menggunakan Metode Single Shot Multibox Detector pada Alat Bantu Tongkat Tunanetra Berbasis Kamera*”, “*Smart Cane Navigation for Visually Impaired*” oleh A. Nivedita, dkk dan S. Farhan dengan judul “*Pendeteksian Objek Manusia secara Real Time dengan Metode MOBILENET-SSD Menggunakan Movidius Neural Compute Stick pada Raspberry Pi*”. Berikut perbandingan inovasi yang ditunjukkan pada Tabel 5.8

Tabel 5.8 Keunggulan ADOPTUKAM

No	Fitur/Komponen	ADOPTUKAM	Sistem A (S. Fuady, dkk)	Sistem B (A. Nivedita, dkk)	Sistem C (S. Farhan)
1	Cara Kerja Sistem	<i>real-time</i>	<i>real-time</i>	<i>real-time</i>	<i>real-time</i>
2	Mini Komputer	Raspberry Pi 4B 4GB	Raspberry Pi	Raspberry Pi 3 B	Raspberry Pi 3B+IntelNCS2
3	Notifikasi deteksi objek	<i>Buzzer</i>	<i>Buzzer dan Vibrator</i>	-	-
4.	Nilai rata-rata akurasi jarak 1-5 meter terhadap objek	82.2%	92%	-	91.67%

5.3.2 Sosial

ADOPTUKAM mampu memberikan kepercayaan diri bagi penyandang tunanetra dalam berinteraksi sosial serta melakukan mobilitas kegiatan sehari-hari dikarenakan produk ini dapat beroperasi sebagai penjagaan terhadap penyandang tunanetra dan menjadi pengetahuan kepada khalayak banyak orang bahwa tongkat tunanetra merupakan salah alat yang dapat dikembangkan dengan dampak yang luar biasa bagi penyandang tunanetra.

5.3.3 Ekonomi

Produk yang telah dibuat menghabiskan sebesar Rp. 3.815.000,- bila membandingkan dengan salah satu tongkat yang telah beredar di pasaran seperti *WeWalk Cane* dengan kisaran harga Rp. 7.000.000,- dengan fungsi dan maksud tujuan yang sama, tongkat terbilang masih cukup terjangkau

BAB 6 : Kesimpulan dan Saran

6.1 Kesimpulan

Dari hasil pengujian sistem yang telah dilakukan, didapatkan beberapa kesimpulan antara lain:

1. Alat yang dibangun sebagai proses deteksi objek dengan menggunakan *pre - trained label COCO* dengan model MobileNetSSD sebagai arsitektur datanya. faktor yang mempengaruhi pendeteksian berupa pencahayaan dan jarak. faktor jarak yang di uji dengan nilai intensitas cahaya sebesar 421 *lux* dengan hasil pengujian rata-rata akurasi sebesar 69% pada jarak 1m, 81% pada jarak 2m, 83% pada jarak 3m, 91% pada jarak 4m, dan 87% pada jarak 5m. Sehingga jarak uji 1m - 5m memiliki akurasi sebesar 82.2% dan presisi sebesar 91% terbilang efektif untuk pendeteksian dan pemberian peringatan kepada pengguna untuk tidak menabrak yang mana apabila lebih dari 5 meter dirasa kurang efektif karena pengguna masih terlalu jauh terhadap objek sehingga tidak perlu diberikan peringatan untuk mengganti arah haluan saat berjalan. Semakin jauh jarak objek yang dideteksi maka nilai akan menurun serta kualitas gambar yang diterima sangat mempengaruhi keberhasilan pendeteksian.
2. Pada penelitian ini, sistem yang dirancang berfungsi dengan baik karena mendapatkan skor SUS >68 dengan kesimpulan bahwa sistem yang dibuat sangat mudah dipahami dan mempertimbangkan untuk menggunakan sistem ini lagi dengan nilai persentase kerja *buzzer* sebesar 100% karena berhasil mengeluarkan bunyi ketika sistem berhasil mendeteksi objeknya.

6.2 Saran

Berdasarkan kesimpulan diatas, terdapat beberapa hal yang direkomendasikan untuk pengembangan selanjutnya diantaranya:

1. Perlu dilakukan penambahan berupa sensor api atau kamera yang dapat mendeteksi api dan lubang pada tongkat.
2. Komponen elektronika dan kemasan dapat dibuat lebih ringkas sehingga dapat mengurangi beban dari tongkat tunanetra agar tidak terlalu berat.
3. Pengujian bisa lebih dibuat beragam dalam pendeteksian sehingga untuk mengetahui seberapa besar tingkat pendeteksian apabila hanya mendeteksi objek tunggal dengan objek jamak.
4. Penambahan modul *text to speech* yang dapat membaca tulisan yang ingin dibacakan menjadi suara yang dapat didengarkan pengguna sehingga penyandang mengetahui jenis

objek yang akan dihadapinya dan pengembangan menggunakan modul GPS dan GSM agar dapat memberikan lokasi di mana penyandang tunanetra berada kepada keluarga atau kerabat ketika penyandang tunanetra tersesat.



DAFTAR PUSTAKA

- [1] “Definisi Penyandang Disabilitas | Kementerian Sosial Republik Indonesia.” <https://kemensos.go.id/definisi-penyandang-disabilitas> (accessed Jul. 18, 2022).
- [2] “Kemensos Dorong Aksesibilitas Informasi Ramah Penyandang Disabilitas | Kementerian Sosial Republik Indonesia.” <https://kemensos.go.id/kemensos-dorong-aksesibilitas-informasi-ramah-penyandang-disabilitas> (accessed Jun. 27, 2022).
- [3] “Apa itu Tunanetra,” *Persatuan Tunanetra Indonesia*. <https://pertuni.or.id/> (accessed Jun. 15, 2021).
- [4] “Kesetaraan Hak bagi Penyandang Disabilitas | Puspensos.” <https://puspensos.kemensos.go.id/kesetaraan-hak-bagi-penyandang-disabilitas> (accessed Jul. 18, 2022).
- [5] “Article 19 – Living independently and being included in the community | United Nations Enable.” <https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-19-living-independently-and-being-included-in-the-community.html> (accessed Jul. 18, 2022).
- [6] M. Arifin and H. Widyastono, “Studi Komparasi Negara Indonesia Dengan Negara Lain Tentang Pemanfaatan Teknologi Dalam Pendidikan Khusus Menggunakan Studi Komparasi Teknologi Asistif Negara Maju dan Berbasis Literasi,” *BEST J. Biol. Educ. Sains Technol.*, vol. 3, no. 1, Art. no. 1, Apr. 2020, doi: 10.30743/best.v3i1.2469.
- [7] “Assistive technology.” <https://www.who.int/news-room/fact-sheets/detail/assistive-technology> (accessed Nov. 17, 2021).
- [8] N. A., M. Sindhuja, G. Asha, R. S. Subasree, and S. Monisha, “Smart cane navigation for visually impaired,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 190–192, Apr. 2019.
- [9] S. Fuady, N. Nehru, and G. Anggraeni, “Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera,” *J. Electr. Power Control Autom. JEPCA*, vol. 3, no. 2, Art. no. 2, Dec. 2020, doi: 10.33087/jepca.v3i2.38.
- [10] Indhujah. R, Sowmya. K, and T. Gowthaman, “Smart Cane For Visually Impaired People,” *Int. J. Eng. Sci. IJES*, pp. 128–133, Mar. 2015.
- [11] T. K. Rohcastu, C. Rahmad, and Rawansyah, “Object Detection System Sebagai Alat Bantu Mendeteksi Objek Sekitar untuk Penyandang Tunanetra.,” *Semin. Inform. Apl. Polinema*, pp. 81–88, 2019.
- [12] C. Lesmana, R. Lim, and L. W. Santoso, “Implementasi Face Recognition menggunakan Raspberry pi untuk akses Ruang Pribadi,” *J. Infra*, vol. 7, no. 1, Art. no. 1, Jul. 2019.
- [13] 14:00-17:00, “ISO 9999:2016,” *ISO*. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/05/60547.html> (accessed Jul. 23, 2022).
- [14] F. Sindy, “Pendeteksian Objek Manusia Secara Real Time dengan Metode Mobilenet-SSD Menggunakan Movidius Neural Compute Stick pada Raspberry PI,” 2019, Accessed: Jul. 25, 2022. [Online]. Available: <https://repositori.usu.ac.id/handle/123456789/25830>
- [15] M. Zunaidi, M. Ramadhan, and H. Winata, “Faktor Penentu Tingkat Keberhasilan Sistem Deteksi Wajah pada Citra Digital,” no. 2, pp. 121–130, 2015.
- [16] “Pengaruh Perbedaan Intesitas Pencahayaan Terhadap Identifikasi Objek Gerak | Seminar Nasional Sistem Informasi (SENASIF),” Nov. 2019, Accessed: Jul. 24, 2022. [Online]. Available: <https://jurnalfti.unmer.ac.id/index.php/senasif/article/view/263>

LAMPIRAN – LAMPIRAN

- Tabel Data hasil pengujian

Percobaan – 1								
No.	Objek	Jarak (m)	Jumlah Aktual	Jumlah Terdeteksi	TP	TN	FP	FN
1	Manusia	1	4	2	2			2
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					6	-	-	2
2	Manusia	2	4	3	3			1
	Sepeda		1	0	-			1
	Kursi		3	3	3			
Total					6	-	-	2
3	Manusia	3	4	2	2			2
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					6	-	-	2
4	Manusia	4	4	3	3			1
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					7	-	-	1
5	Manusia	5	4	4	4			
	Sepeda		1	2	1		1	

	Kursi		3	3	3			
Total					8	-	1	-

Percobaan – 2								
No.	Objek	Jarak (m)	Jumlah Aktual	Jumlah Terdeteksi	TP	TN	FP	FN
1	Manusia	1	4	3	3			1
	Sepeda		1	2	1		1	
	Kursi		3	3	3			
Total					7	-	1	1
2	Manusia	2	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8	-	-	-
3	Manusia	3	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	4	3		1	
Total					8	-	1	-
4	Manusia	4	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8	-	-	-
5	Manusia	5	4	3	3			1
	Sepeda		1	1	1			

	Kursi		3	4	3		1	
Total					7	-	1	1

Percobaan - 3								
No.	Objek	Jarak (m)	Jumlah Aktual	Jumlah Terdeteksi	TP	TN	FP	FN
1	Manusia	1	4	3	3			1
	Sepeda		1	2	1		1	
	Kursi		3	3	3			
Total					7			
2	Manusia	2	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8			
3	Manusia	3	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8	-	-	-
4	Manusia	4	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8	-	-	-
5	Manusia	5	4	5	4		1	
	Sepeda		1	1	1			

	Kursi		3	3	3			
Total					8	-	1	-

Percobaan - 4								
No.	Objek	Jarak (m)	Jumlah Aktual	Jumlah Terdeteksi	TP	TN	FP	FN
1	Manusia	1	4	3	3			1
	Sepeda		1	1	1			
	Kursi		3	5	3		2	
Total					7	-	2	1
2	Manusia	2	4	4	3			1
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					7	-	-	-
3	Manusia	3	4	4	3			1
	Sepeda		1	2	1		1	
	Kursi		3	3	3			
Total					7	-	1	-
4	Manusia	4	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	3			
Total					8	-	-	-
5	Manusia	5	4	5	4		1	
	Sepeda		1	1	1			

	Kursi		3	3	3			
Total					8	-	1	-







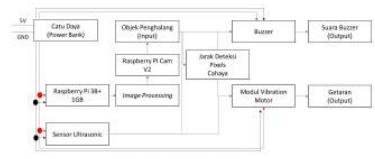

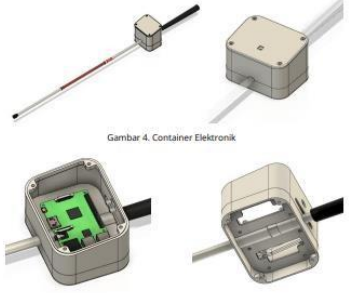
Percobaan - 5								
No.	Objek	Jarak (m)	Jumlah Aktual	Jumlah Terdeteksi	TP	TN	FP	FN
1	Manusia	1	4	3	3			1
	Sepeda		1	3	1		2	
	Kursi		3	4	3		1	
Total					7	-	3	1
2	Manusia	2	4	4	4			
	Sepeda		1	0	-			1
	Kursi		3	4	3		1	
Total					7	-	1	1
3	Manusia	3	4	4	4			
	Sepeda		1	1	1			
	Kursi		3	3	2			1
Total					6	-	-	1
4	Manusia	4	4	4	4			
	Sepeda		1	2	1		1	
	Kursi		3	4	3		1	
Total					8	-	2	-
5	Manusia	5	4	4	4			
	Sepeda		1	2	1		1	

	Kursi		3	3	3			
Total					8	-	1	-

- *Logbook* Kegiatan Selama Proses Tugas Akhir 2

Hari, Tanggal	Deskripsi Kegiatan
Selasa, 1 Maret 2022	Diskusi mengenai pemrosesan citra
Sabtu, 5 Maret 2022	Mempelajari Model YolotinyV4
Senin, 7 Maret 2022	Mempelajari Model MobileNet SSD
Selasa, 15 Maret 2022	Pembelian Komponen
Rabu, 13 April 2022	Pembuatan Desain <i>Case</i> pada Solidwork
Senin, 18 April 2022	Bimbingan dengan dosen pembimbing 1
Selasa, 19 April 2022	Diskusi mengenai pendeteksian objek
Selasa, 17 Mei 2022	Pembuatan TA201
Kamis, 19 Mei 2022	Bimbingan dengan dosen pembimbing 1 mengenai <i>real-time</i> deteksi
Jumat, 20 Mei 2022	Pengumpulan TA201
Selasa, 24 Mei 2022	Pembuatan Laporan Desain Eksperimen
Sabtu, 28 Mei 2022	Pengambilan <i>Case</i> 3D print
Minggu, 29 Mei 2022	Memperbaiki sistem agar pendeteksian tidak delay
Rabu, 1 Juni 2022	Pengintegrasian komponen pada <i>case</i> ke tongkat
Senin, 13 Juni 2022	Diskusi parameter pengujian bersama dosen pembimbing 1
Rabu, 15 Juni 2022	Pembuatan Laporan TRP 202
Jumat, 24 Juni 2022	Pengambilan Data Objek
Senin, 27 Juni 2022	Diskusi dengan dosen pembimbing 1 mengenai metode pengujian metric machine learning dan SUS
Rabu, 29 Juni 2022	Pengambilan Data Objek
Selasa, 5 Juli 2022	Pengambilan Data SUS di Karangnom
Senin, 18 Juli 2022	Pengerjaan laporan tugas akhir

- Dokumen TA201 dan TA202

<p style="text-align: center;">  Capstone Design 2021 - 2022 </p> <p style="text-align: center;">TECHNICAL REPORT</p> <p>IDENTITAS</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Dokumentasi Proses</td> <td>TA201</td> </tr> <tr> <td>Topik / Judul Capstone Design</td> <td>ADOPTUKAM (Alat Deteksi Objek pada Tongkat Bantuan Tunanetra Berbasis Kamera)</td> </tr> <tr> <td>Nama Lengkap</td> <td>Bayu Arif Ramadhani Reyhan Pramadhana</td> </tr> <tr> <td>No. Induk Mahasiswa (NIM)</td> <td>18524108 18524076</td> </tr> <tr> <td>Dosen Pembimbing 1</td> <td style="text-align: center;">  Suatmi Murnika, S.T., M.Eng </td> </tr> <tr> <td>Dosen Pembimbing 2</td> <td style="text-align: center;">Dwi Ana Ratna Wati, S.T., M.Eng.</td> </tr> </table>	Dokumentasi Proses	TA201	Topik / Judul Capstone Design	ADOPTUKAM (Alat Deteksi Objek pada Tongkat Bantuan Tunanetra Berbasis Kamera)	Nama Lengkap	Bayu Arif Ramadhani Reyhan Pramadhana	No. Induk Mahasiswa (NIM)	18524108 18524076	Dosen Pembimbing 1	 Suatmi Murnika, S.T., M.Eng	Dosen Pembimbing 2	Dwi Ana Ratna Wati, S.T., M.Eng.	<p style="text-align: center;">  Capstone Design 2021 - 2022 </p> <p>Spesifikasi Sistem</p> <ol style="list-style-type: none"> Tongkat yang digunakan menggunakan tongkat tunanetra dengan panjang 100 cm jarak uji pendeteksian : 1 - 5 m dengan pengujian pada mini market Sumber listrik atau catu daya yang digunakan menggunakan power bank dengan dengan input 5V objek yang dijadikan pendeteksian berupa pinta, rak besi, dan kardus pada mini market Cover komponen elektronik memiliki dimensi panjang 9 cm, lebar 12 cm, tinggi 11 cm dan berat sekitar 2,5 kg Raspberry Pi Model sebagai mini PC agar menjadi sistem yang portable kamera raspberry pi dengan resolusi sebesar 5MP sebagai sensor pendeteksian sensor ultrasonic sebagai backup memberikan notifikasi pendeteksian penghalang atau objek yang telah dilakukan training image berupa suara melalui buzzer dan getaran dari modul vibration <p>Desain rancangan awal</p> <p>a. Diagram Alir</p> <pre> graph TD Start([START]) --> Step1[Instansi Cat Kamera sensor dan modul] Step1 --> Step2[Kamera menangkap Citra / Objek] Step2 --> Step3[Pemrosesan Citra] Step3 --> Step4[Labeling Objek] Step4 --> Step5[Deteksi Objek] Step5 --> Step6[Output Vibration & Buzzer] </pre>
Dokumentasi Proses	TA201												
Topik / Judul Capstone Design	ADOPTUKAM (Alat Deteksi Objek pada Tongkat Bantuan Tunanetra Berbasis Kamera)												
Nama Lengkap	Bayu Arif Ramadhani Reyhan Pramadhana												
No. Induk Mahasiswa (NIM)	18524108 18524076												
Dosen Pembimbing 1	 Suatmi Murnika, S.T., M.Eng												
Dosen Pembimbing 2	Dwi Ana Ratna Wati, S.T., M.Eng.												
<p style="text-align: center;">  Capstone Design 2021 - 2022 </p> <p>b. Desain Elektronik</p>  <p style="text-align: center;">Gambar 3. Diagram Blok Sistem</p>	<p style="text-align: center;">  Capstone Design 2021 - 2022 </p> <p>c. Desain Container dan Elektronik</p>  <p style="text-align: center;">Gambar 4. Container Elektronik</p> <p style="text-align: center;">Gambar 5. Tampak dalam Container</p> <p>Keterangan :</p> <ol style="list-style-type: none"> pada Gambar 4, cover memiliki 2 buah bagian yang seakan terpisah dengan tongkatnya dengan bagian sisi atas akan menjadi tempat dudukan raspberry pi yang terhubung dengan modul kamera yang terletak pada tutup case bagian paling atas dan catu daya terletak pada bagian sisi bawah yang dengan modul vibration motor dan buzzer terletak pada tampak luar pada gambar 5 yang berada dekat posisi pegangan tangan agar pengguna lebih merasakan getaran dan suara notifikasi pendeteksian lebih terdengar. <p>d. Sistem Detector Object</p> <p>Sample object berupa "Cardboard" dengan sistem Yolov4-tiny</p>												

TECHNICAL REPORT

IDENTITAS

Dokumentasi Proses	TA202
Topik / Judul Capstone Design	ADOPTUKAM: Alat Deteksi Objek Untuk Bantuan Tuna Netra Berbasis Kamera
Nama Lengkap	Bayu Arif Ramadhan Reyhan Pramadhana
No. Induk Mahasiswa (NIM)	18524108 18524076
Dosen Pembimbing 1	(Suatmi Murnani S.T., M.Eng.)
Dosen Pembimbing 2	(Dwi Ana Ratnawati S.T., M.Eng.)

Metode / Rancangan Pengujian Sistem

Berikut adalah metode dari pengujian sistem yang dilakukan untuk menguji perangkat secara menyeluruh:

1. Pengujian dilakukan sebanyak 5 kali percobaan dengan mendeteksi ketiga objek yaitu manusia, kursi dan sepeda/motor dengan menggunakan *pre-trained model COCO* dengan metode *SSD MobileNet Deploy*. Terjadi perubahan dari *YOLOv4* akibat hasil pendeteksian di mana fps yang didapatkan kurang dari 1 fps sehingga terjadi delay selama 20 detik pada saat pendeteksian, jarak antar objek dengan tongkat sebesar 1 m - 5 m, pengambilan jarak tersebut karena kamera baru dapat menangkap atau yang terlihat masuk pada frame kamera.

2. pengujian kemudian menghitung akurasi dari sistem secara *Real - Time* menggunakan raspberry pi dengan berbagai macam kondisi seperti jumlah objek, jarak kamera dan

1. Metode Confusion Matrix

merupakan sebuah pengukuran performa untuk permasalahan model *Machine Learning (ML)* yang diterapkan pada sebuah sistem deteksi objek. Dimana pada percobaan ini confusion matrix digunakan dalam pengambilan data dengan menerapkan variabel dari setiap kondisi aktual dengan kondisi deteksi seperti pada tabel berikut :

Tabel Variabel Kondisi		Kondisi Aktual	
		Ada	Tidak Ada
Hasil Deteksi	Terdeteksi	True Positive (TP)	False Positive (FP)
	Tidak Terdeteksi	False Negative (FN)	True Negative (TN)

Keterangan :

- **True Positive (TP)** adalah kondisi dimana objek yang berhasil dideteksi oleh sistem.
- **False Positive (FP)** adalah kondisi dimana sistem menganggap objek atau kondisi lain sebagai objek yang dituju.
- **False Negative (FN)** adalah kondisi dimana ada objek yang tidak terdeteksi oleh sistem.
- **True Negative (TN)** adalah kondisi dimana sistem mampu mengabaikan objek selain manusia.

2. Dependent Variables

a. Responsive and error

Untuk mengetahui tingkat responsiveness sensor, sistem akan diuji dengan meletakkan objek dengan rentang jarak minimal 30 cm dan maksimal 100 cm guna menganalisa apakah keberadaan objek terdeteksi oleh sensor, kemudian nilai error akan dianalisa dengan membandingkan hasil baca sensor dengan uji manual menggunakan alat ukur baik penggaris maupun meteran.

b. User Experience

Pembuatan alat yang ditujukan guna membantu penyangkutan tunanetra melakukan mobilitas, maka dari itu kenyamanan dan nilai fungsionalitas dari alat membutuhkan penilaian langsung oleh pengguna

Metode Pengukuran untuk pengujian Sistem

Metode yang digunakan dalam pengukuran dan pengujian nya adalah sebagai berikut :

1. Metode Confusion Matrix

Untuk melakukan evaluasi pada percobaan, ada beberapa aspek yang akan diperhatikan dalam penelitian ini. Evaluasi menggunakan hasil dari rumus perhitungan Recall, Precision, F-Score dan akurasi yang mampu dihasilkan oleh sistem berdasarkan hasil uji coba yang telah dilakukan.

Aspek yang akan diperhatikan dalam percobaan ini dengan menggunakan hasil evaluasi dari rumus perhitungan berikut :

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{TP}{TP + FN + FP} \times 100 \%$$

Keterangan :

- **Recall** merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif dan salah negatif.
- **Precision** merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.
- **F-Score** adalah nilai yang didapat untuk menunjukkan performa algoritma yang diterapkan.
- **Akurasi** adalah rasio prediksi benar (positif) dengan keseluruhan data.

2. SUS (System Usability Scale) Questionnaire

Evaluasi alat dilakukan dengan pengambilan data berupa 10 pertanyaan dengan 5 pilihan respons didasarkan pada pengalaman responden saat uji alat secara langsung, berikut lampiran tabel SUSQ :

No	Pertanyaan	Jawaban				
		STS (Sangat Tidak Setuju)	TS (Tidak Setuju)	N (Netral)	S (Setuju)	SS (Sangat Setuju)
1.	Saya berpikir akan menggunakan sistem ini lagi	1	2	3	4	5

Kursi		3	3	2		1
Total						
4	Manusia	4	4	4		
	Sepeda	1	2	1	1	
	Kursi	3	4	3	1	
Total						
5	Manusia	5	4	4	4	
	Sepeda	1	2	1	1	
	Kursi	3	3	3		
Total						

Akumulasi Hasil Percobaan Deteksi Manusia

Total Objek Manusia = 100	TP	TN	FP	FN
Jarak (m)				
1	14	-	-	6
2	18	-	-	2
3	17	-	-	3
4	19	-	-	1
5	19	-	2	1

$$\text{Akurasi} = \frac{(TP + TN)}{(TP + FN + FP)} \times 100\%$$

$$= \frac{14 + 0 + 6 + 0 + 0}{14 + 0 + 6 + 0 + 0} \times 100\%$$

$$= 70\%$$

$$\text{Presisi} = \frac{(TP)}{(TP + FP)} \times 100\%$$

$$= \frac{14}{14 + 0} \times 100\%$$

$$= 100\%$$

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \times 100\%$$

$$= \frac{14}{14 + 0} \times 100\%$$

Dokumen Desain Eksperimen untuk Penelitian ADOPTUKAM

Catatan untuk peneliti

Suatmi Murnani, S.T., M.Eng.

Dosen Pembimbing 1

Dwi Ana Ratmawati, S.T., M.Eng.

Dosen Pembimbing 2

Bayu Arif Ramadhan

Mahasiswa

Reyhan Pramadhana

Mahasiswa

1. Desain Eksperimen

1.1 Pendahuluan

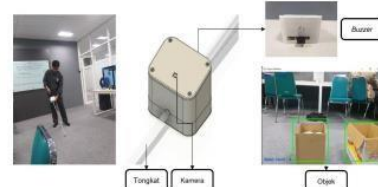
Dokumen ini berisi desain eksperimen yang digunakan untuk menguji performa tingkat tunanetra dengan sistem pendeteksian objek menggunakan kamera Raspberry Pi Cam Module Rev 1.3 bernama ADOPTUKAM, sebuah sistem yang memberikan notifikasi berupa suara Buzzer secara Real-time dengan Algoritma SSD MobileNet Lite berbasis kamera sebagai masukan untuk pendeteksian objek dengan pengukuran menggunakan Confusion Matrix.

1.2 Partisipan Eksperimen

Eksperimen ini membutuhkan kurang lebih 5 partisipan dengan kriteria sbb:

- Mahasiswa Universitas Islam Indonesia.
- Penderita Tunanetra Yayasan Karangnong Klaten, Jawa Tengah

1.3 Langkah Pengambilan Data



Gambar 1.1 Pengaturan eksperimen pengambilan data

a. Perangkat Eksperimen

Perangkat yang digunakan dalam eksperimen ini sebagai berikut:

- Raspberry Pi 4 model B 4GB

Eksperimen ini dilakukan menggunakan mini PC atau microprocessor dengan spesifikasi prosesor SoC Broadcom BCM 2711, quad-core Cortex-A72 (ARM v8) 64-bit 1.5 GHz, RAM 4 GB, sistem operasi Raspbian OS, dan Microsoft Visual Studio Code untuk menjalankan Object Detector dengan bahasa Python

- Tongkat Tunanetra

Tongkat ini digunakan untuk membantu mobilitas pengguna untuk berjalan dengan ukuran tongkat dewasa umumnya memiliki panjang sebesar 110cm serta pelatikan dari keseluruhan alat menggunakan case dengan bahan PLA yang telah di 3D-Print dengan ukuran Panjang : 12 cm, Lebar : 9 cm, Tinggi : 11 cm.

- Kamera Raspberry Pi Cam modul 1.3

Kamera ini berfungsi sebagai penangkapan gambar secara Real-Time yang akan mengirimkan langsung ke sistem pengolahan citra pada Raspberry Pi 4 dengan spesifikasi kamera yaitu SMP dengan ukuran 25mm x 23mm x 8mm.

- Buzzer

Buzzer pada sistem ini berperan sebagai aktuator untuk menghasilkan bunyi sebagai keluaran tanda kamera berhasil mendeteksi objek yang ada di depannya dengan besar tegangan sebesar 3.3V.

b. Pengaturan Eksperimen

Eksperimen dilakukan dengan pengaturan seperti ditunjukkan pada Gambar 1.1, partisipan menggunakan tongkat tunanetra yang telah dilengkapi dengan alat ADOPTUKAM dan berjalan ke arah rintangan dengan parameter jarak yang telah ditentukan yaitu 1 m – 5 m serta nilai lux yang tetap mengikuti kondisi saat pengambilan data dengan percobaan mendeteksi jumlah orang, motor, dan kursi.

c. Prosedur Eksperimen

i. Partisipan

Prosedur untuk partisipan pada penelitian ini ditunjukkan pada Tabel 1.2.

Tabel 1.2 Prosedur dan durasi pengambilan data dari partisipan

Kegiatan	Estimasi Waktu
1. Sesi pendahuluan	
a. Penjelasan eksperimen secara lisan	2 menit
2. Sesi eksperimen	15 menit
a. Eksperimen Metode SSD Lite	
3. Sesi evaluasi	
a. Pengisian biodata	1 menit
b. Pengisian kuesioner (SiusQ)	8 menit
c. Partisipan memberikan tanggapan/saran	2 menit
TOTAL	28 menit

ii. Observer

Tabel 1.3 berisi prosedur yang harus dilakukan oleh observer dalam eksperimen ini.

Tabel 1.3 Penjelasan prosedur yang harus dilakukan oleh observer

Jenis Kegiatan	Penjelasan
Penjelasan eksperimen secara lisan	Penjelasan lisan yang akan disampaikan kepada setiap partisipan pada sesi pertama eksperimen memuat hal-hal sebagai berikut: <ul style="list-style-type: none"> Gambaran umum eksperimen Gambaran umum apa yang harus dilakukan oleh partisipan (Misalnya: partisipan diminta untuk berjalan menggunakan tongkat tunanetra ke arah terdapat objek di depannya untuk men-trigger pendeteksian oleh kamera dan buzzer)

2. Evaluasi

2.1. Independent Variables

Independent variable yang digunakan pada eksperimen ini adalah dengan membandingkan nilai akurasi per jarak dengan kondisi lux yang sama untuk mengetahui jarak yang ideal.

2.2. Dependent Variables

Pada eksperimen ini, dependent variable yang digunakan sbb:

a. Metode Confusion Matrix

merupakan sebuah pengukuran performa untuk permasalahan model Machine Learning (ML) yang diterapkan pada sebuah sistem deteksi objek. Dimana pada percobaan ini confusion matrix digunakan dalam pengambilan data dengan menerapkan variabel dari setiap kondisi aktual dengan kondisi deteksi seperti pada tabel berikut :

Tabel Variabel Kondisi	Kondisi Aktual		
	Ada	Tidak Ada	
Hasil Deteksi	Terdeteksi	True Positive (TP)	False Positive (FP)
	Tidak Terdeteksi	False Negative (FN)	True Negative (TN)

Keterangan:

- **True Positive (TP)** adalah kondisi dimana objek yang berhasil dideteksi oleh sistem.
- **False Positive (FP)** adalah kondisi dimana sistem menganggap objek atau kondisi lain sebagai objek yang dituju.
- **False Negative (FN)** adalah kondisi dimana ada objek yang tidak terdeteksi oleh sistem.
- **True Negative (TN)** adalah kondisi dimana sistem mampu mengabaikan objek selain manusia.

Evaluasi menggunakan hasil dari rumus perhitungan Recall, Precision, F-Score dan akurasi yang mampu dihasilkan oleh sistem berdasarkan hasil uji coba yang telah dilakukan. Aspek

- Kode Program Deteksi Objek

```
# USAGE
# python capstone.py --prototxt MobileNetSSD_deploy.prototxt.txt --model
MobileNetSSD_deploy.caffemodel

# import the necessary packages
from imutils.video import VideoStream
from imutils.video import FPS
from multiprocessing import Process
from multiprocessing import Queue
import numpy as np
import argparse
import imutils
import time
import cv2
import os
import RPi.GPIO as GPIO
from time import sleep

def classify_frame(net, inputQueue, outputQueue):
    # keep looping
    while True:
        # check to see if there is a frame in our input queue
        if not inputQueue.empty():
            # grab the frame from the input queue, resize it, and
            # construct a blob from it
            frame = inputQueue.get()
            frame = cv2.resize(frame, (300, 300))
            blob = cv2.dnn.blobFromImage(frame, 0.007843,
                                         (300, 300), 127.5)

            # set the blob as input to our deep learning object
            # detector and obtain the detections
            net.setInput(blob)
```

```

detections = net.forward()

# write the detections to the output queue
outputQueue.put(detections)

#Disable warnings (optional)
GPIO.setwarnings(False)

#Select GPIO mode
GPIO.setmode(GPIO.BCM)

buzzer=24
GPIO.setup(buzzer,GPIO.OUT)

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
            "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
            "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
            "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")

```



```

net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

# initialize the input queue (frames), output queue (detections),
# and the list of actual detections returned by the child process
inputQueue = Queue(maxsize=1)
outputQueue = Queue(maxsize=1)
detections = None

# construct a child process *independent* from our main process of
# execution
print("[INFO] starting process...")
p = Process(target=classify_frame, args=(net, inputQueue,
    outputQueue,))
p.daemon = True
p.start()

# initialize the video stream, allow the camera sensor to warm up,
# and initialize the FPS counter
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
#vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)
fps = FPS().start()
writer = None

# Section FPS Calculation
fps_start_time = 0
fps = 0

# loop over the frames from the video stream
while True:
    # FPS Calculation
    fps_start_time = time.time()

```

```

# grab the frame from the threaded video stream, resize it, and
# grab its imensions
frame = vs.read()
frame = imutils.resize(frame, width=500)
(fH, fW) = frame.shape[:2]
# if the input queue *is* empty, give the current frame to
# classify
if inputQueue.empty():
    inputQueue.put(frame)

# if the output queue *is not* empty, grab the detections
if not outputQueue.empty():
    detections = outputQueue.get()

# initialize the people counter for each frame
object_count = 0

# check to see if our detections are not None (and if so, we'll
# draw the detections on the frame)
if detections is not None:
    # loop over the detections
    for i in np.arange(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated
        # with the prediction
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the `confidence`
        # is greater than the minimum confidence
        if confidence < args["confidence"]:
            continue

        # otherwise, extract the index of the class label from
        # the `detections`, then compute the (x, y)-coordinates
        # of the bounding box for the object

```

```

idx = int(detections[0, 0, i, 1])
dims = np.array([fW, fH, fW, fH])
box = detections[0, 0, i, 3:7] * dims
(startX, startY, endX, endY) = box.astype("int")

# draw the prediction on the frame
label = "{:}: {:.2f}%".format(CLASSES[idx],
                             confidence * 100)
cv2.rectangle(frame, (startX, startY), (endX, endY),
              COLORS[idx], 2)
y = startY - 15 if startY - 15 > 15 else startY + 15
cv2.putText(frame, label, (startX, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
# count the people
if (CLASSES[idx] == "person"):
    object_count += 1
    GPIO.output(buzzer,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(buzzer,GPIO.LOW)

if (CLASSES[idx] == "chair"):
    object_count += 1
    GPIO.output(buzzer,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(buzzer,GPIO.LOW)

if (CLASSES[idx] == "motorbike"):
    object_count += 1
    GPIO.output(buzzer,GPIO.HIGH)
    sleep(0.5)
    GPIO.output(buzzer,GPIO.LOW)

```

```

object2_count = "Object Count : {}".format(str(object_count))

```

```

cv2.putText(frame, object2_count, (10, frame.shape[0] - 25),
cv2.FONT_HERSHEY_SIMPLEX, 0.60, (255, 255, 255), 1)

```

```

# Section FPS Calculation

```

```

fps_end_time = time.time()

```

```

time_diff = fps_end_time - fps_start_time

```

```

fps = 1 / (time_diff)

```

```

fps_text = "FPS : {:.2f}".format(fps)

```

```

cv2.putText(frame, fps_text, (10,30),

```

```

cv2.FONT_HERSHEY_COMPLEX, 0.60, (255, 255, 255), 1)

```

```

# show the output frame

```

```

cv2.imshow("Frame", frame)

```

```

key = cv2.waitKey(1) & 0xFF

```

```

# if the `q` key was pressed, break from the loop

```

```

if key == ord("q"):

```

```

    break

```

```

# update the FPS counter

```

```

#fps.update()

```

```

# stop the timer and display FPS information

```

```

#fps.stop()

```

```

#print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))

```

```

#print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

```

```

# do a bit of cleanup

```

```

cv2.destroyAllWindows()

```

```

vs.stop()

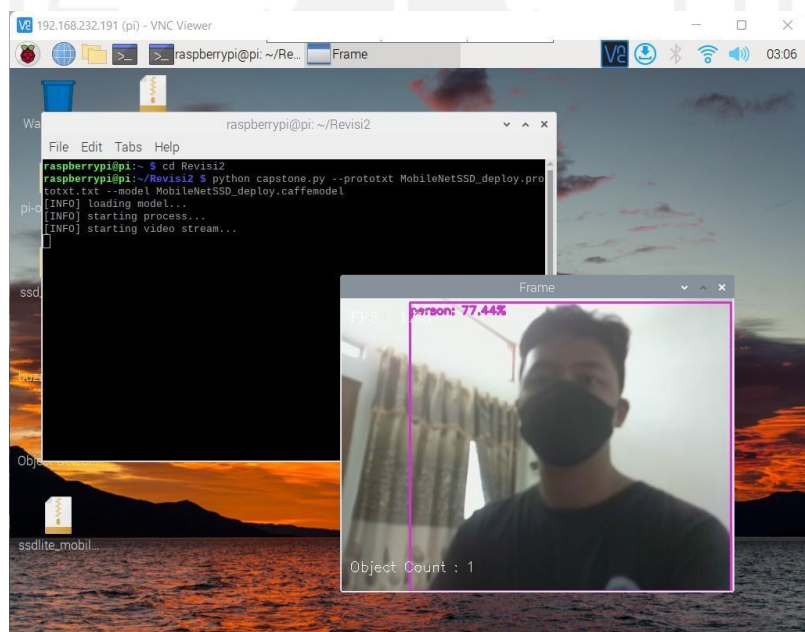
```

- Model *Deep Learning MobileNet* Metode SSD (*Single Shot Multibox Detector*) Version Deploy

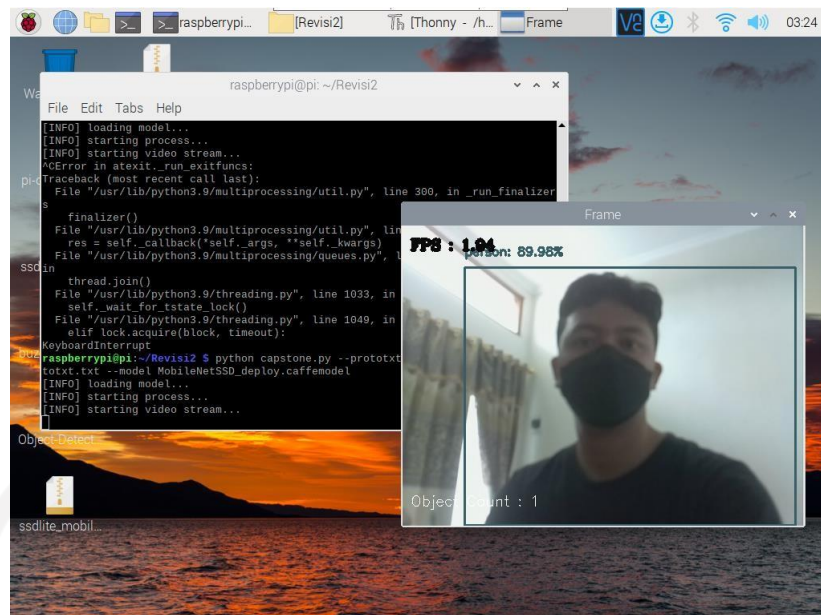
<https://github.com/PINTO0309/MobileNet-SSD-RealSense/tree/master/caffemodel/MobileNetSSD>

- Desain model/produk/sistem termasuk aplikasi jika ada
- Dokumentasi Kunjungan Ke Yayasan Karangnom, Klaten

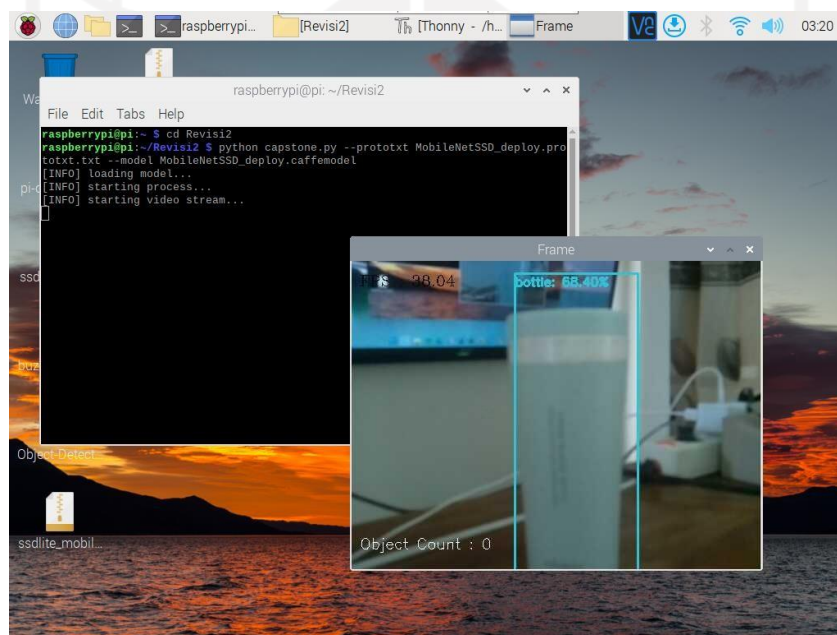




Percobaan Model *Deep Learning* MobileNET SSD + FIFO

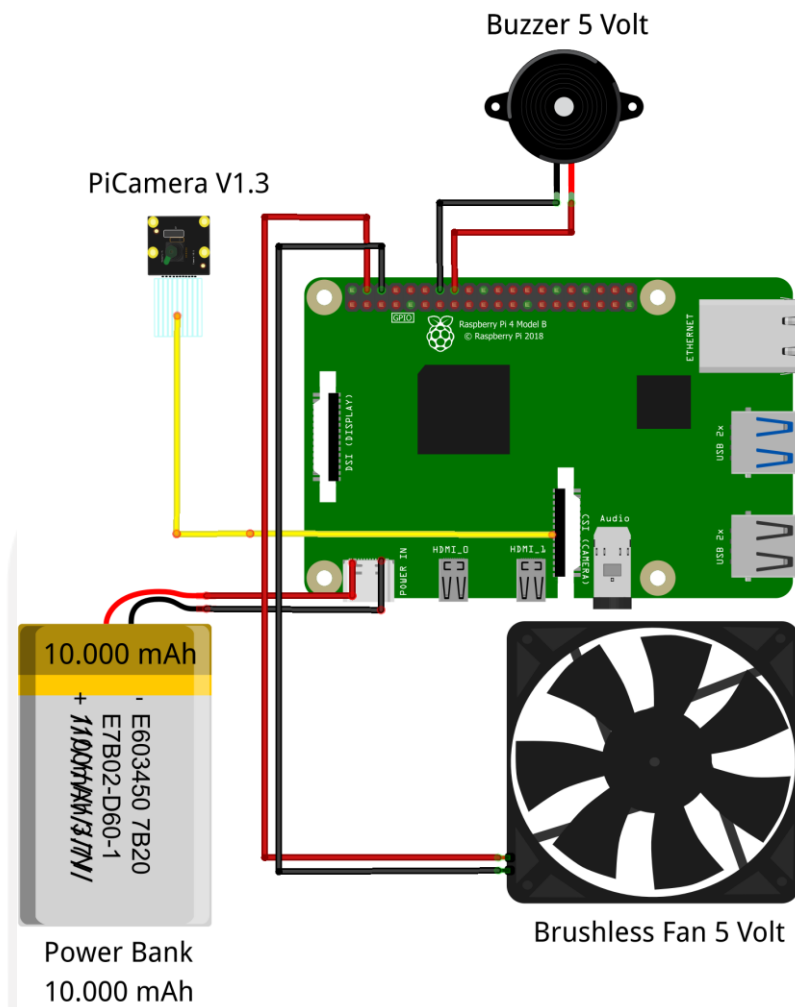


Percobaan Sistem Deteksi Objek dengan Aksi



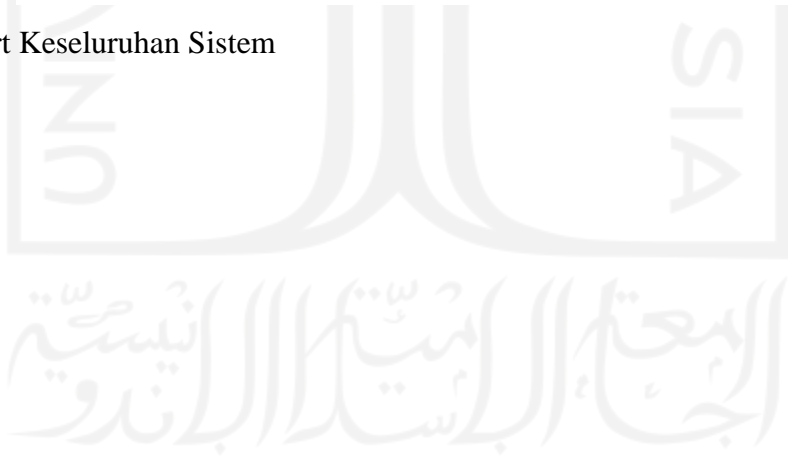
Percobaan Sistem Deteksi Objek Tanpa Aksi

- Skematik elektronik keseluruhan



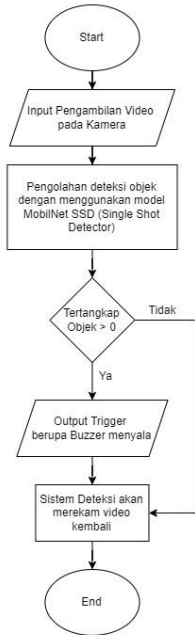
fritzing

- Flowchart Keseluruhan Sistem

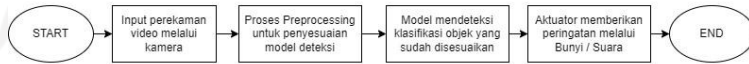
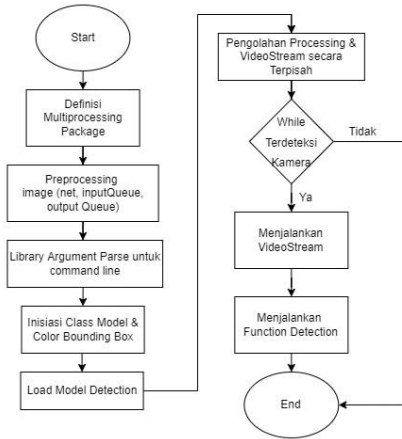


FlowChart Keseluruhan Capstone Design
ADOPTUKAM : Alat Deteksi Objek Pada Tongkat Bantu
Tunanetra Berbasis Kamera

Proses Deteksi

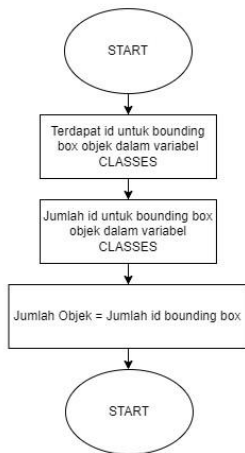


Proses Sistem Deteksi

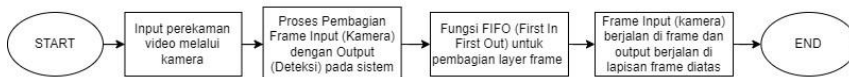
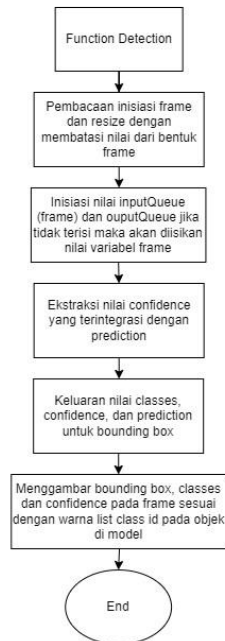


Proses Deteksi Objek Secara Umum

Proses Deteksi Jumlah Objek



Function Detection



Proses Pemisahan Perekaman Video dengan Deteksi Objek

- Dokumentasi keuangan

No	Jenis Pengeluaran	Realisasi Biaya	
		Kuantitas	Total Harga
1	Raspberry Pi Camera V1.3 5MP	1 pcs	Rp. 350.000,-
2	Raspberry Pi 4B 4GB	1 pcs	Rp. 2.790.000,-
3	Baterai Power Bank 10.000 mAh	1 pcs	Rp. 80.000,-
4	Tongkat Tunanetra	1 pcs	Rp. 73.500,-
5	Kemasan 3D print	256 gr	Rp. 400.000,-
6	Buzzer Aktif	2 pcs	Rp. 6.000,-
7	Micro SD 64 GB	1 pcs	Rp. 96.000,-
8	Kabel Jumper	4 pcs	Rp. 4.000,-
9	Kabel FPC kamera	2 pcs	Rp. 16.000,-
Total			Rp. 3.815.500,-

Lembar Saran Penguji Internal dan Eksternal:

Internal		
No.	Komentar	Capaian Perbaikan
1	Batasan Masalah ditambahkan	Menambahkan batasan masalah pada Bab I
2	Batasan realistik belum ada	Menambahkan batasan realistik pada Bab I
3	Setiap gambar, tabel, dan flowchart harus dijelaskan di dalam paragraf	Menambahkan penjelasan pada tabel, gambar, dan flowchart
4	Di jelaskan bahwa gambar 5.6, 5.7 dan 5.8 adalah visualisasi dari tabel 5.9	Hanya memamparkan gambar grafik agar tidak <i>redundant</i> .
5	Cek typo dan penulisan	Terlaksana



PERBAIKAN TUGAS AKHIR YANG DISARANKAN
PADA WAKTU UJIAN TUGAS AKHIR
Tanggal Ujian: 3 Agustus 2022

Dosen Penguji

Nama : Elvira Sukma Wahyuni, S.Pd., M.Eng.

Kelompok yang diuji

Kelompok : N1

Judul : ADOPTUKAM: Alat Deteksi Objek pada Tongkat Bantu Tunanetra
Berbasis Kamera

Saran/Komentar :

- 1/ Basasan masalah ditambahkan !
- 2/ Batasan realitis belum ada !
- 3/ Setiap gambar, tabel, dan flowchar harus dijelaskan didalam paragraf !
- 4/ di jelaskan balok gambar p.6, p.7 dan p.8 adah visualisasi dari tabel p.9, cek yg lainnya ?
- 5/ cek typo dan penulisan

Eksternal

No.	Komentar	Capaian Perbaikan
1		