



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE184801

RANCANG BANGUN TROLI BELANJA PENGIKUT MANUSIA BERBASIS VISI KOMPUTER

Alvin Abraham
NRP 07111640000056

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EE184801

**RANCANG BANGUN TROLI BELANJA PENGIKUT
MANUSIA BERBASIS VISI KOMPUTER**

Alvin Abraham
NRP 0711164000056

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EE184801

**DESIGN OF HUMAN FOLLOWER SHOPPING
TROLLEY BASED ON COMPUTER VISION**

Alvin Abraham
NRP 07111640000056

Supervisor
Dr. Ir. Djoko Purwanto, M.Eng.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical and Intelligent Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan dari tugas akhir saya dengan judul :

Rancang Bangun Troli Belanja Pengikut Manusia Berbasis Visi Komputer

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap dalam daftar pustaka. Apabila ternyata pernyataan ini tidak benar, maka saya bersedia untuk menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 20 Januari 2020



Alvin Abraham

NRP. 07111640000056

----- *Halaman ini sengaja dikosongkan* -----

RANCANG BANGUN TROLI BELANJA PENGIKUT MANUSIA BERBASIS VISI KOMPUTER

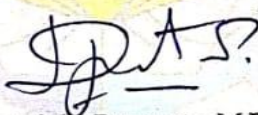
TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Dr. Ir. Djoko Purwanto, M.Eng.
NIP. 196512111990021002



----- *Halaman ini sengaja dikosongkan* -----

RANCANG BANGUN TROLI BELANJA PENGIKUT MANUSIA BERBASIS VISI KOMPUTER

Nama : Alvin Abraham
Pembimbing : Dr. Ir. Djoko Purwanto, M.Eng.

ABSTRAK

Troli belanja konvensional mempunyai keterbatasan, yaitu memerlukan tenaga manusia untuk mengendalikan gerakannya. Keterbatasan tersebut dapat menimbulkan ketidaknyamanan bagi beberapa pelanggan ketika berbelanja. Oleh karena itu, diperlukan adanya suatu inovasi pada troli belanja sehingga dapat memberikan kenyamanan bagi pelanggan ketika berbelanja. Dalam tugas akhir ini, telah dirancang dan dibangun sebuah troli belanja pengikut manusia berbasis visi komputer. Troli belanja ini dilengkapi dengan sebuah USB *webcam* serta sebuah komputer papan tunggal Raspberry Pi. USB *webcam* digunakan untuk mengakuisisi citra dalam bentuk digital sehingga dapat diproses menggunakan Raspberry Pi. Raspberry Pi digunakan untuk menjalankan proses visi komputer dan algoritma kontrol gerak. Proses visi komputer yang digunakan meliputi deteksi warna dan ekstraksi kontur. Nilai dari absis piksel minimum dan maksimum penyusun kontur digunakan untuk mengindikasikan posisi manusia relatif terhadap troli belanja. Kemudian kedua nilai ini diproses menggunakan algoritma kontrol PID untuk mengontrol kecepatan rotasi dari kedua motor yang terpasang pada kedua roda belakang troli belanja. Perbedaan kecepatan rotasi antara roda kanan dan roda kiri digunakan untuk mengendalikan gerakan troli belanja. Hasil yang didapatkan dari pengujian sistem menunjukkan bahwa rancang bangun troli belanja ini dapat bergerak mengikuti manusia secara otomatis sesuai dengan warna aksesoris yang dipakai oleh manusia.

Kata kunci : troli belanja, visi komputer, kontrol PID

----- *Halaman ini sengaja dikosongkan* -----

DESIGN OF HUMAN FOLLOWER SHOPPING TROLLEY BASED ON COMPUTER VISION

Name : Alvin Abraham
Supervisor : Dr. Ir. Djoko Purwanto, M.Eng.

ABSTRACT

Conventional shopping trolleys have a limitation, which require human energy to control the movement. The limitation can cause inconvenience for certain customers during shopping. Therefore, the innovation on the shopping trolleys is required so that they can provide conveniences for the customers during shopping. In this final project, the human follower shopping trolley based on computer vision has been designed and built. This shopping trolley is equipped with single USB webcam and a single board computer Raspberry Pi. USB webcam is used to acquire images in the digital form so that they can be processed using the Raspberry Pi. Raspberry Pi is used to run the computer vision process and movement control algorithm. The computer vision process used here includes color detection and the contour extraction. The minimum and maximum pixel abscissa value which compose the contour, both are used to indicate the human position relative to the shopping trolley. Then, these values are processed using PID control algorithm to control the rotational velocity of both motors installed on both rear wheels of the shopping trolley. The difference in the rotational velocity between the right wheel and the left wheel is used to control the movement of the shopping trolley. The result obtained from system testing shows that the design of shopping trolley is able to follow human automatically according to the color of accessory worn by the human.

Keywords : shopping trolley, computer vision, PID control

----- *Halaman ini sengaja dikosongkan* -----

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul : “**Rancang Bangun Troli Belanja Pengikut Manusia Berbasis Visi Komputer**” yaitu sebagai salah satu persyaratan dalam menyelesaikan pendidikan program sarjana dengan bidang studi Elektronika di Departemen Teknik Elektro ITS.

Laporan tugas akhir ini dapat diselesaikan penulis dengan bantuan dan dukungan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih sebesar-besarnya kepada :

1. Ayah, ibu, dan kakak yang senantiasa memberikan nasihat, kasih sayang, dan doa untuk keberhasilan penulis.
2. Bapak Dr. Ir. Djoko Purwanto, M.Eng. selaku dosen pembimbing yang telah memberikan gagasan, arahan, serta bimbingan kepada penulis selama proses pengerjaan tugas akhir.
3. Seluruh dosen dari bidang studi Elektronika di Departemen Teknik Elektro ITS yang telah membagikan banyak ilmu pengetahuan dan keterampilan, pengalaman, serta motivasi kepada penulis.
4. Teman-teman dari bidang studi Elektronika yang telah membantu dan memberikan dukungan bagi penulis selama penulis menjalani perkuliahan di Departemen Teknik Elektro ITS.

Penulis menyadari bahwa masih banyak pengembangan yang dapat dilakukan dalam tugas akhir ini. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca yang membangun. Penulis juga berharap laporan tugas akhir ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya.

Surabaya, 20 Januari 2020

Alvin Abraham

----- *Halaman ini sengaja dikosongkan* -----

DAFTAR ISI

PERNYATAAN KEASLIAN.....	i
LEMBAR PENGESAHAN	iii
ABSTRAK.....	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penelitian	2
1.5. Metodologi Penelitian.....	3
1.5.1. Studi Literatur	3
1.5.2. Perancangan Perangkat Keras (<i>Hardware</i>).....	3
1.5.3. Perancangan Perangkat Lunak (<i>Software</i>)	3
1.5.4. Pengujian dan Perbaikan Sistem	3
1.5.5. Penyusunan Laporan Tugas Akhir	4
1.6. Sistematika Penulisan	4
1.7. Relevansi atau Manfaat.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Citra Digital	7
2.1.1. Konsep Dasar Citra Digital	7
2.1.2. Representasi Citra Digital	7
2.1.3. Elemen-Elemen Citra Digital	8
2.1.4. Citra Biner.....	9
2.1.5. Citra Keabuan	9
2.1.6. Citra Warna	10
2.2. Visi Komputer	10
2.2.1. Pengaburan Citra.....	11
2.2.2. Pengembangan Citra	11
2.2.3. Segmentasi Citra	12
2.2.4. Segmentasi Warna.....	12
2.2.5. Analisis Blob.....	13
2.2.6. Ruang Warna HSV (<i>Hue-Saturation-Value</i>).....	14
2.2.7. OpenCV (<i>Open Source Computer Vision Library</i>).....	15

2.3. Sistem Kontrol PID (<i>Proportional-Integral-Derivative</i>).....	16
2.3.1. Konsep Dasar Kontroler PID	16
2.3.2. Kontrol Proporsional (P).....	17
2.3.3. Kontrol Integral (I).....	17
2.3.4. Kontrol Derivatif (D)	17
2.3.5. Kontroler PID Digital	18
2.4. PWM (<i>Pulse Width Modulation</i>).....	19
2.5. Sistem <i>Differential Drive</i> pada <i>Mobile Robot</i>	20
2.6. Raspberry Pi	21
2.7. <i>Webcam (Web Camera)</i>	22
BAB III PERANCANGAN SISTEM	23
3.1. Diagram Blok Sistem	23
3.2. Perancangan Troli Belanja	24
3.3. Perancangan Antarmuka Pengguna	28
3.4. Pengolahan Input Pengguna	31
3.5. Proses Akuisisi Citra	31
3.6. Proses Visi Komputer.....	32
3.6.1. Proses Deteksi Warna	32
3.6.2. Proses Ekstraksi Kontur.....	36
3.7. Proses Kontrol PID.....	38
3.8. Proses <i>Differential Drive</i>	39
BAB IV PENGUJIAN DAN ANALISIS	41
4.1. Pengujian Visi Troli Belanja	41
4.2. Pengujian Kecepatan Troli Belanja	42
4.3. Pengujian Jarak Troli Belanja.....	44
4.4. Pengujian Orientasi Troli Belanja	45
4.5. Pengujian Kontrol Gerak Troli Belanja	46
BAB V KESIMPULAN DAN SARAN	49
1.1. Kesimpulan.....	49
1.2. Saran.....	49
DAFTAR PUSTAKA	51
LAMPIRAN	53
BIODATA PENULIS	65

DAFTAR GAMBAR

Gambar 2.1 Sistem koordinat pada citra digital	8
Gambar 2.2 Contoh citra biner	9
Gambar 2.3 Contoh citra keabuan	10
Gambar 2.4 Diagram ruang warna HSV	15
Gambar 2.5 Logo OpenCV	15
Gambar 2.6 Diagram blok sistem kontrol PID	16
Gambar 2.7 Sinyal PWM dan tegangan rata-rata yang dihasilkan	19
Gambar 2.8 Sinyal PWM dengan <i>duty cycle</i> yang berbeda-beda	20
Gambar 2.9 Sistem <i>differential drive</i> pada <i>mobile robot</i>	21
Gambar 2.10 Logo Raspbian	21
Gambar 2.11 Pin GPIO pada Raspberry Pi 3 model B+	22
Gambar 3.1 Diagram blok sistem secara keseluruhan	23
Gambar 3.2 Perancangan mekanik troli belanja	24
Gambar 3.3 Perancangan elektronik troli belanja	28
Gambar 3.4 Tampilan halaman koneksi pada aplikasi <i>smartphone</i>	29
Gambar 3.5 Tampilan halaman perintah pada aplikasi <i>smartphone</i>	30
Gambar 3.6 Hasil proses akuisisi citra	32
Gambar 3.7 Diagram blok proses visi komputer	32
Gambar 3.8 Hasil proses pengubahan ukuran citra	33
Gambar 3.9 Hasil proses pengaburan citra	34
Gambar 3.10 Hasil proses konversi ruang warna citra	34
Gambar 3.11 Hasil proses pengembangan citra	35
Gambar 3.12 Hasil proses transformasi morfologi citra	36
Gambar 3.13 Hasil proses pencarian kontur	36
Gambar 3.14 Hasil proses seleksi luas kontur	37
Gambar 3.15 Hasil proses seleksi <i>centroid</i> kontur	38
Gambar 3.16 Hasil proses pengambilan nilai x_l dan x_r	38
Gambar 4.1 Hasil pengujian jarak troli belanja terhadap (a) luas kontur terdeteksi dan (b) selisih nilai x_r dan x_l	45
Gambar 4.2 Hasil pengujian kontrol gerak troli belanja pada lintasan lurus	47
Gambar 4.3 Hasil pengujian kontrol gerak troli belanja pada lintasan berbelok ke kiri	47
Gambar 4.4 Hasil pengujian kontrol gerak troli belanja pada lintasan berbelok ke kanan	48

----- *Halaman ini sengaja dikosongkan* -----

DAFTAR TABEL

Tabel 2.1 Efek kenaikan nilai parameter PID secara independen	18
Tabel 3.1 Spesifikasi USB <i>webcam</i> Logitech C170.....	25
Tabel 3.2 Spesifikasi Raspberry Pi 3 model B+.....	26
Tabel 3.3 Konfigurasi pin modul Bluetooth HC-05.....	27
Tabel 3.4 Konfigurasi pin modul <i>motor driver</i> IBT-2	27
Tabel 4.1 Hasil pengujian visi troli belanja.....	41
Tabel 4.2 Hasil pengujian kecepatan troli belanja	43
Tabel 4.3 Hasil pengujian jarak troli belanja	44
Tabel 4.4 Hasil pengujian orientasi troli belanja.....	46

----- *Halaman ini sengaja dikosongkan* -----

BAB I

PENDAHULUAN

Tugas akhir adalah sebuah penelitian yang dilakukan sebagai salah satu persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS). Topik yang akan dibahas dalam tugas akhir ini mengenai rancang bangun troli belanja pengikut manusia berbasis visi komputer.

Pada Bab Pendahuluan dibahas mengenai hal-hal yang mendahului pelaksanaan tugas akhir. Hal-hal ini meliputi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan, serta relevansi atau manfaat.

1.1. Latar Belakang

Dewasa ini, pasar swalayan menjadi pilihan utama sebagai tempat untuk berbelanja dikarenakan ketersediaan barang kebutuhan sehari-hari manusia yang lengkap di dalamnya [1]. Sebagian besar pasar swalayan ini menyediakan alat bantu yang dapat mempermudah pelanggan dalam membawa barang belanjaan mereka. Alat bantu dapat berupa keranjang belanja atau troli belanja, dengan berbagai model dan ukuran kapasitas [2]. Troli belanja adalah alat angkut beroda yang disediakan oleh pasar swalayan kepada pelanggannya sebagai tempat untuk membawa barang belanjaan mereka menuju meja kasir [3].

Keberadaan troli belanja sebagai alternatif dari keranjang belanja dapat mempermudah pelanggan ketika berbelanja. Namun, troli belanja ini memiliki keterbatasan. Troli belanja yang cukup sering digunakan ini merupakan troli belanja konvensional yang tentunya memerlukan tenaga manusia untuk mengendalikan gerakannya [4]. Keterbatasan ini kadang dapat menimbulkan ketidaknyamanan bagi beberapa pelanggan tertentu. Selama berbelanja, pelanggan perlu untuk mendorong troli belanja yang digunakan dari rak ke rak untuk mengumpulkan berbagai barang yang ingin dibeli [5]. Aktivitas ini membutuhkan tenaga yang tidak sedikit, apalagi jika roda troli belanja yang digunakan bermasalah, sehingga troli belanja tersebut menjadi cukup sulit untuk dikendalikan [6]. Selain itu, aktivitas ini juga dapat membatasi aktivitas tangan kita untuk melakukan aktivitas yang lain, misalnya menggunakan ponsel atau sebagainya [7]. Pengendalian troli belanja ini juga sulit untuk dilakukan oleh pelanggan yang telah lanjut usia atau yang memiliki disabilitas fisik tertentu karena

dapat menimbulkan risiko cedera [8]. Selain itu, pengawasan orangtua terhadap anak ketika berbelanja dapat terabaikan karena orangtua akan sibuk dengan mengurus troli belanja yang digunakan [9]. Kecerobohan pelanggan dalam menggunakan troli belanja juga sering terjadi, seperti meninggalkan troli belanja di sembarang tempat karena sibuk memilih barang sehingga dapat menghalangi jalan pelanggan yang lain [10].

Berdasarkan beberapa permasalahan di atas, maka diperlukan suatu inovasi pada troli belanja sehingga dapat memberikan kemudahan serta kenyamanan bagi pelanggan ketika berbelanja. Oleh karena itu, di dalam tugas akhir ini dirancang dan dibangun troli belanja pengikut manusia berbasis visi komputer. Troli belanja ini dapat bergerak untuk mengikuti manusia secara otomatis sesuai dengan warna aksesoris yang digunakan oleh manusia. Dengan dibuatnya rancang bangun troli belanja pengikut manusia ini, diharapkan dapat memberikan solusi dalam pengembangan troli belanja konvensional sehingga dapat memberikan kemudahan serta kenyamanan bagi pelanggan ketika berbelanja.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah :

1. Bagaimana visi komputer dapat diimplementasikan pada troli belanja untuk mendeteksi manusia?
2. Bagaimana visi komputer dapat diimplementasikan pada troli belanja untuk mengetahui posisi manusia?
3. Bagaimana algoritma kontrol gerak dapat diimplementasikan untuk mengendalikan gerakan troli belanja?
4. Bagaimana integrasi dari visi komputer dan algoritma kontrol gerak sehingga troli belanja dapat bergerak mengikuti manusia?

1.3. Batasan Masalah

Batasan masalah yang ditetapkan dalam tugas akhir ini adalah :

1. Rancang bangun berupa *prototype* (prototipe) troli belanja.
2. Sistem bekerja dalam kondisi permukaan lantai yang datar.
3. Sistem bekerja dalam kondisi pencahayaan ruang yang stabil.
4. Manusia yang diikuti menggunakan aksesoris berwarna tunggal.

1.4. Tujuan Penelitian

Tujuan penelitian yang ingin dicapai dalam tugas akhir ini adalah :

1. Mengimplementasikan deteksi warna untuk mendeteksi manusia dari warna aksesoris yang dipakai.

2. Mengimplementasikan ekstraksi kontur untuk mengetahui posisi dari manusia yang diikuti relatif terhadap troli belanja.
3. Mengimplementasikan algoritma kontrol PID pada sistem penggerak dua roda untuk mengendalikan gerakan troli belanja.
4. Menggunakan nilai absis piksel minimum dan maksimum penyusun kontur sebagai variabel yang dikontrol oleh kontroler PID.

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam tugas akhir ini adalah :

1.5.1. Studi Literatur

Pada tahap ini, penulis melakukan pengumpulan dan pembelajaran beberapa literatur yang dapat menunjang dalam pelaksanaan tugas akhir. Literatur ini diperoleh dari buku-buku, jurnal-jurnal, artikel-artikel, dan forum-forum diskusi di internet. Literatur yang dipelajari ini mengenai metode deteksi warna dan ekstraksi kontur dalam visi komputer, sistem kontrol PID, sistem *differential drive*, teknologi komunikasi Bluetooth, metode PWM, dan fitur-fitur yang terdapat dalam Raspberry Pi.

1.5.2. Perancangan Perangkat Keras (*Hardware*)

Pada tahap ini, penulis melakukan perancangan *hardware* dimana terdiri dari desain dan realisasi alat. Desain alat ini meliputi beberapa komponen, yaitu troli belanja (terdiri dari troli dan keranjang belanja), Raspberry Pi, USB *webcam*, *push button*, *buzzer*, *motor driver*, DC *gear motor*, modul Bluetooth, dan juga baterai. Realisasi alat ini merupakan kegiatan menyusun komponen-komponen ini menjadi sebuah *prototype* troli belanja pengikut manusia yang sesuai.

1.5.3. Perancangan Perangkat Lunak (*Software*)

Pada tahap ini, penulis melakukan perancangan *software* dimana terdiri dari program yang diimplementasikan pada Raspberry Pi dan juga program untuk aplikasi pada *smartphone*. Program pada Raspberry Pi meliputi bagian visi komputer untuk deteksi warna dan ekstraksi kontur, bagian algoritma kontrol gerak untuk pengendalian gerak troli belanja. Program untuk aplikasi *smartphone* sebagai antarmuka pengguna (*user interface*) untuk sarana interaksi antara manusia dan troli belanja.

1.5.4. Pengujian dan Perbaikan Sistem

Pada tahap ini, penulis melakukan pengujian terhadap sistem untuk mengetahui kinerjanya dari penggabungan *software* dan *hardware* yang

telah dibuat. Hasil pengujian ditentukan berdasarkan kinerja troli belanja dalam mengendalikan gerak ketika mengikuti manusia. Hasil pengujian dianalisis sehingga dapat diperoleh karakteristik dari sistem yang dapat digunakan sebagai acuan dalam melakukan perbaikan sistem. Apabila karakteristik dari sistem belum sesuai dengan kehendak, maka dilakukan perbaikan sistem sehingga kinerja sistem dapat dioptimalkan.

1.5.5. Penyusunan Laporan Tugas Akhir

Pada tahap ini, penulis melakukan penyusunan laporan tugas akhir sesuai dengan struktur dan format yang telah ditentukan. Laporan tugas akhir ini terdiri dari tiga bagian utama, yaitu bagian awal yang memuat bahan-bahan preliminer, bagian inti/pokok yang memuat naskah utama dari tugas akhir, serta bagian akhir yang memuat bahan-bahan referensi. Penyusunan laporan tugas akhir ini dilakukan secara bersamaan dengan studi literatur, perancangan, pengujian, dan perbaikan alat.

1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam tugas akhir ini adalah :

1. Bab 1 : Pendahuluan
Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi atau manfaat dari penelitian yang dilakukan dalam tugas akhir.
2. Bab 2 : Teori Penunjang
Bab ini berisi tentang beberapa teori dasar yang berhubungan dengan permasalahan yang dibahas serta dapat menunjang dalam penelitian yang dilakukan dalam tugas akhir.
3. Bab 3 : Perancangan Sistem
Bab ini berisi tentang beberapa penjelasan dalam perancangan sistem yang meliputi perancangan *hardware* serta *software* dari penelitian yang dilakukan dalam tugas akhir.
4. Bab 4 : Pengujian dan Analisis
Bab ini berisi tentang beberapa penjelasan dalam pengujian sistem yang meliputi proses akuisisi data serta analisis data dari penelitian yang dilakukan dalam tugas akhir.
5. Bab 5 : Kesimpulan dan Saran
Bab ini berisi tentang sebuah kesimpulan akhir dari penelitian yang dilakukan dalam tugas akhir serta saran-saran untuk pengembangan penelitian lebih lanjut.

1.7. Relevansi atau Manfaat

Hasil dari penelitian yang dilakukan dalam tugas akhir diharapkan dapat memberikan manfaat sebagai berikut :

1. Dapat memberikan solusi untuk kegiatan pengembangan troli belanja konvensional yang ada sehingga dapat memberikan kemudahan dan kenyamanan bagi pelanggan ketika berbelanja.
2. Dapat menjadi referensi belajar mengenai aplikasi dari visi komputer dan dapat memberikan inspirasi bagi mahasiswa yang lainnya untuk mengembangkan penelitian lebih lanjut.

----- *Halaman ini sengaja dikosongkan* -----

BAB II

TEORI PENUNJANG

Pada Bab Teori Penunjang dibahas mengenai teori-teori dasar yang berhubungan dengan permasalahan yang dibahas serta dapat menunjang penelitian yang dilakukan dalam tugas akhir.

2.1. Citra Digital

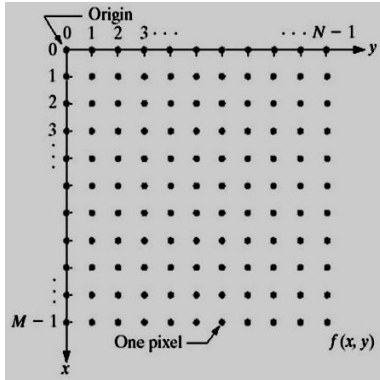
Citra adalah suatu kombinasi antara titik, garis, bidang, dan warna untuk menciptakan imitasi dari suatu objek. Citra terbagi ke dalam dua jenis, yaitu citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu. Citra analog tidak dapat direpresentasikan di komputer sehingga tidak dapat diproses di komputer secara langsung. Oleh karena itu, agar citra analog dapat diproses di komputer, perlu dilakukan proses konversi analog ke digital terlebih dahulu. Proses digitalisasi citra dapat dilakukan dengan menggunakan kamera digital.

2.1.1. Konsep Dasar Citra Digital

Citra digital adalah representasi dari citra yang diambil oleh mesin dengan bentuk pendekatan yang berdasarkan sampling dan kuantisasi. Sampling pada citra ini menyatakan besarnya ukuran piksel (titik) pada citra yang tersusun sebagai baris dan kolom. Kuantisasi ini menyatakan besarnya tingkat intensitas yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit yang dipakai oleh mesin. Dengan kata lain, kuantisasi pada citra menyatakan jumlah warna yang terdapat dalam citra tersebut. Semakin banyak jumlah bit yang digunakan untuk merepresentasikan suatu citra analog, maka semakin bagus juga kualitas dari citra digital yang dihasilkan. Hal tersebut dikarenakan jumlah warna yang dapat direpresentasikan semakin banyak.

2.1.2. Representasi Citra Digital

Sebuah citra digital dapat direpresentasikan dengan sebuah matriks yang terdiri M baris dan N kolom. Perpotongan antara baris dan kolom tersebut disebut dengan piksel (*picture element*), adalah elemen terkecil dalam suatu citra. Suatu piksel memiliki dua parameter, yaitu koordinat dan intensitas warna. Nilai yang terdapat dalam koordinat (x, y) adalah $f(x, y)$, yaitu besar intensitas warna dari piksel dalam koordinat tersebut. Gambar 2.1 menunjukkan sistem koordinat pada citra digital.



Gambar 2.1 Sistem koordinat pada citra digital

Berdasarkan gambaran di atas, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x, y)$, dimana harga x (baris) serta y (kolom) merupakan koordinat sebuah piksel, sedangkan harga $f(x, y)$ merupakan nilai fungsi pada setiap koordinat (x, y) ini yang menyatakan besar intensitas warna dari piksel pada koordinat tersebut.

$$f(x, y) = \begin{bmatrix} f(0,0) & \dots & f(0, N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \dots & f(M-1, N-1) \end{bmatrix} \dots\dots\dots (2.1)$$

2.1.3. Elemen-Elemen Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar ini dimanipulasi dalam pengolahan citra serta dieksploitasi lebih lanjut di dalam analisis citra. Elemen-elemen dasar ini diantaranya adalah kecerahan, kontras, warna, kontur, bentuk, dan tekstur.

- Kecerahan adalah intensitas cahaya yang dipancarkan piksel dalam suatu citra dan dapat ditangkap oleh sistem penglihatan. Kecerahan dari sebuah piksel dalam suatu citra merupakan intensitas rata-rata dari suatu area yang melingkupi piksel tersebut.
- Kontras menyatakan sebaran terang dan gelap di dalam suatu citra. Pada citra yang baik, komponen dari terang dan gelap ini tersebar secara merata sehingga rincian citra semakin bertambah.
- Warna merupakan persepsi yang ditangkap oleh sistem penglihatan terhadap panjang gelombang cahaya yang dipantulkan oleh objek.

- Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas sekitar piksel-piksel yang bertetangga. Karena perubahan intensitas, sistem penglihatan mampu mendeteksi tepi-tepi objek dalam citra.
- Bentuk merupakan properti intrinsik dari objek 3 dimensi. Bentuk adalah properti intrinsik utama untuk sistem penglihatan manusia.
- Tekstur dapat dicirikan sebagai suatu distribusi spasial dari derajat keabuan pada sekumpulan piksel yang bertetangga. Tekstur adalah keteraturan suatu pola tertentu yang terbentuk dari susunan piksel-piksel yang terdapat dalam suatu citra.

2.1.4. Citra Biner

Citra biner merupakan jenis citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner disebut juga sebagai citra B&W (*Black and White*) atau citra monokrom dan hanya diperlukan satu bit untuk mewakili nilai di setiap pikselnya. Citra biner ini sering kali muncul sebagai hasil dari proses pengolahan citra, seperti segmentasi citra, pengambangan citra, morfologi citra, dan sebagainya. Gambar 2.2 menunjukkan contoh citra biner.



Gambar 2.2 Contoh citra biner

2.1.5. Citra Keabuan

Citra keabuan merupakan jenis citra digital yang hanya memiliki 1 nilai kanal di setiap pikselnya. Nilai ini digunakan untuk menunjukkan tingkat intensitas. Citra ini memiliki warna dari hitam, keabuan, hingga putih. Citra ini memberikan kemungkinan warna lebih banyak daripada citra biner karena terdapat warna-warna lain di antara hitam dan putih. Banyaknya dari kemungkinan warna bergantung pada jumlah bit yang digunakan. Sebagai contohnya, apabila digunakan skala keabuan 8-bit,

maka ini memberikan kemungkinan warna sebanyak 256 warna dengan nilai intensitas yang mulai dari 0 (hitam) hingga 255 (putih). Gambar 2.3 menunjukkan contoh citra keabuan.



Gambar 2.3 Contoh citra keabuan

2.1.6. Citra Warna

Setiap piksel pada citra warna memiliki warna yang spesifik yang merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru. Format citra warna sering disebut sebagai citra RGB (*Red-Green-Blue*). Tiap warna dasar memiliki intensitas sendiri yang menunjukkan tingkat keabuan warna dasar tersebut (umumnya 8-bit). Sebagai contoh, warna kuning adalah kombinasi dari warna merah dan warna hijau, sehingga nilai RGB-nya adalah (255 255 0). Dengan demikian, setiap piksel pada citra warna ini membutuhkan media penyimpanan data 3 byte. Jumlah kombinasi warna yang mungkin untuk citra ini adalah lebih dari 16 juta warna. Dengan demikian, citra warna dapat dianggap mencakup semua warna yang ada dan inilah sebabnya citra ini dinamakan *true color*.

2.2. Visi Komputer

Visi komputer merupakan ilmu dan metode dalam menggunakan komputer untuk memperoleh, mengolah, menganalisis, dan memahami data visual seperti gambar atau video. Visi komputer merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, pengenalan pola, dan juga pembuatan keputusan. Terdapat beberapa proses yang menunjang tugas dari visi komputer, yaitu proses penangkapan citra (*image acquisition*), proses pengolahan citra (*image processing*), proses analisis citra (*image analysis*), dan proses pemahaman citra (*image understanding*).

Pengolahan citra adalah manipulasi citra dengan teknik dan metode tertentu, khususnya dengan menggunakan komputer, menjadi citra lain yang memiliki kualitas lebih baik dan sesuai dengan keinginan pemakai. Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Pada dasarnya, tujuan dari pengolahan citra adalah mentransformasikan citra sehingga informasi yang terdapat di dalamnya dapat dibuat lebih jelas.

2.2.1. Pengaburan Citra

Konvolusi merupakan operasi mendasar pada citra dimana operasi matematis diterapkan pada setiap piksel untuk memperoleh hasil yang diinginkan. Untuk tujuan ini, matriks lain yang disebut kernel digunakan, dimana ukuran lebih kecil dari citra yang ingin dioperasikan. Ini disebut juga dengan filter. Filter ini diterapkan pada setiap piksel citra dan nilai baru yang telah diperoleh menjadi nilai pada piksel tersebut. Setiap sel di dalam kernel memiliki suatu nilai. Kernel ini diletakkan di atas citra dan nilai-nilai piksel yang bersangkutan dikalikan dan juga dijumlahkan menghasilkan suatu nilai piksel yang baru.

Filter yang digunakan untuk pengaburan disebut dengan *low pass filter*, karena memungkinkan frekuensi rendah masuk dan menghentikan frekuensi tinggi. Frekuensi disini artinya perubahan nilai piksel. Nilai piksel di sekitar tepi berubah dengan cepat. Oleh karena itu, frekuensi tinggi perlu difilter untuk dapat menghasilkan citra yang halus. Operasi *blur* bertujuan untuk mengaburkan citra, membuat citra menjadi kurang jelas atau berbeda. Operasi ini dilakukan dengan bantuan dari berbagai kernel *low pass filter*. Operasi ini dapat membantu menghilangkan *noise*. Karena *noise* ini dianggap sebagai *high pass signal*, maka diaplikasikan kernel *low pass filter* untuk membatasi *noise*. Operasi *blur* juga dapat membantu menghaluskan citra, menghilangkan tepi-tepian berintensitas rendah, serta mengurangi detail-detail di dalam citra. Beberapa jenis dari operasi *blur* yang sering digunakan adalah Gaussian *blur*, Median *blur*, dan Bilateral *blur*.

2.2.2. Pengambilan Citra

Thresholding adalah proses pembatasan nilai intensitas dari piksel-piksel di dalam sebuah citra. Tujuan dari pembatasan ini adalah untuk memisahkan daerah antara daerah yang termasuk objek dan daerah yang termasuk *background* pada citra dengan jelas. Citra hasil *thresholding* ini kegunaan biasanya digunakan lebih lanjut pada proses ekstraksi fitur.

Thresholding atau operasi ambang batas akan memetakan nilai intensitas piksel yang memenuhi syarat ambang batas tertentu ke nilai intensitas piksel yang dikehendaki. Dengan pemilihan nilai *threshold* yang tepat, maka piksel-piksel objek dan *background* dapat dipisahkan dengan jelas. Salah satu metode *thresholding* yang sering digunakan adalah *binary thresholding*, yaitu memetakan sebuah citra menjadi citra biner dimana objek akan dipetakan sebagai warna putih sedangkan *background* akan dipetakan sebagai warna hitam.

2.2.3. Segmentasi Citra

Segmentasi citra akan membagi suatu citra menjadi daerah-daerah atau objek-objek yang dimilikinya. Segmentasi citra merupakan sebuah proses memecah sebuah citra digital ke dalam banyak segmen/bagian daerah dimana tidak saling bertabrakan. Dalam konteks citra digital ini, daerah hasil dari segmentasi tersebut merupakan kelompok piksel yang bertetangga atau berhubungan. Segmentasi citra dapat dilakukan melalui beberapa pendekatan, diantaranya adalah :

- Pendekatan batas (*boundary approach*), pendekatan ini dilakukan untuk memperoleh batas yang ada antar daerah.
- Pendekatan tepi (*edge approach*), pendekatan ini dilakukan untuk mengidentifikasi piksel-piksel pada tepi serta menghubungkannya menjadi suatu batas yang diinginkan.
- Pendekatan daerah (*region approach*), pendekatan ini dilakukan untuk membagi citra ke dalam daerah-daerah sehingga didapatkan suatu daerah sesuai dengan kriteria yang diinginkan.

Proses segmentasi digunakan dalam berbagai penerapan. Meskipun metode yang digunakan ini sangat bervariasi, semuanya memiliki tujuan sama, adalah untuk mendapatkan representasi sederhana yang berguna dari sebuah citra. Terdapat beberapa macam metode dalam melakukan segmentasi, cukup sulit untuk menentukan metode yang komprehensif. Oleh karena itu, pemilihan metode ini bergantung pada pendekatan yang akan digunakan dan fitur yang ingin diperoleh dari suatu citra.

2.2.4. Segmentasi Warna

Segmentasi warna adalah proses segmentasi dengan pendekatan daerah yang bekerja dengan menganalisis nilai warna dari setiap piksel pada sebuah citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Salah satu model warna yang umum untuk digunakan dalam segmentasi warna adalah model warna HSV. Segmentasi warna dengan

deteksi warna HSV ini menggunakan dasar seleksi warna dalam model warna HSV dengan nilai toleransi tertentu. Dalam metode segmentasi dengan deteksi warna HSV, dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang dikehendaki. Citra digital ini menggunakan model warna RGB sebagai standar acuan warna. Oleh karena itu, proses awal pada metode ini memerlukan konversi model warna RGB ke HSV. Untuk membentuk segmen sesuai dengan warna yang dikehendaki, maka ditentukan nilai toleransi dalam setiap dimensi warna HSV, kemudian nilai toleransi tersebut dapat digunakan dalam perhitungan proses *thresholding*. Hasil dari proses *thresholding* tersebut akan membentuk segmen daerah dengan warna sesuai dengan toleransi yang dikehendaki.

Secara garis besar, cara kerja dari proses segmentasi warna adalah dengan membandingkan nilai warna di dalam setiap piksel citra terhadap nilai warna sampel. Apabila nilai warna pikselnya berada dalam rentang toleransi nilai warna sampel, maka nilai warna piksel tersebut dibiarkan tetap. Apabila nilai warna piksel tidak berada di dalam rentang toleransi nilai warna sampel, maka nilai warna piksel tersebut digantikan menjadi nilai warna hitam. Proses pada segmentasi warna dengan menggunakan deteksi warna HSV ini menghasilkan segmen warna yang akurat sesuai dengan warna sampel dan nilai toleransi yang diberikan ke proses. Hasil dari segmentasi tersebut menghasilkan segmen citra yang membentuk sebuah *blob*, yaitu sekumpulan piksel bertetangga yang memiliki nilai tertentu. Proses segmentasi warna suatu citra akan menghasilkan citra objek tertutup dengan warna tertentu dan berlatar belakang warna hitam. Dengan demikian, hasil proses segmentasi ini akan sangat berpengaruh pada proses analisis objek.

2.2.5. Analisis Blob

Blob adalah sekumpulan piksel yang memiliki hubungan tetangga. Proses perhitungan *blob* ini dapat dilakukan dengan melakukan analisis piksel yang bertetangga. Piksel bertetangga pada sebuah citra ditentukan sebagai piksel yang berjarak 1 dari piksel asal. Proses perhitungan *blob* ini menggunakan relasi piksel *4-neighbors* atau *8-neighbors*. Pada relasi *4-neighbors*, piksel diagonal tidak dianggap sebagai piksel bertetangga. Pada relasi *8-neighbors*, piksel diagonal dianggap sebagai piksel yang bertetangga sehingga bisa memberikan hasil pemetaan yang lebih akurat daripada relasi *4-neighbors*. Perhitungan *blob* ini terdiri dari beberapa proses, yaitu pemetaan objek, pengumpulan informasi objek, serta filter

objek berdasarkan ukuran. Langkah-langkah perhitungan *blob* ini adalah sebagai berikut :

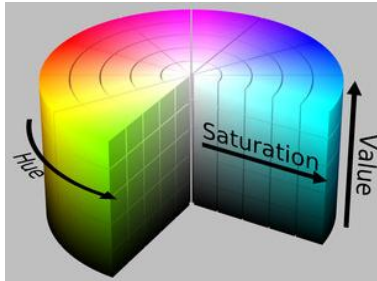
- Proses pemetaan objek akan menelusuri piksel-piksel pada setiap baris yang ada serta memberikan label pada piksel yang memiliki nilai warna selain hitam. Setiap piksel yang memiliki relasi piksel *8-neighbors* akan diberikan label yang sama.
- Proses pengumpulan informasi objek akan mengumpulkan beserta mengolah informasi setiap pikselnya yang bertetangga berdasarkan letak dan label yang dihasilkan oleh proses pemetaan objek. Letak dan label suatu piksel yang bertetangga tersebut digunakan untuk membentuk suatu *blob* dan informasi pendukungnya seperti luas area, tingkat kepenuhan, dan titik pusat *blob*.
- Proses filter objek akan melakukan filter berdasarkan tinggi dan lebar *blob*. *Blob* dengan tinggi atau lebar di bawah nilai minimum akan dihilangkan dari peta objek. Setelah *blob* tersebut dihilangkan, dilakukan *update* label *blob* yang terdeteksi.

2.2.6. Ruang Warna HSV (*Hue-Saturation-Value*)

Hue merupakan salah satu elemen dalam ruang warna HSV yang mewakili nilai suatu warna sehingga nilai dari toleransi *hue* juga akan mempengaruhi nilai warna yang terseleksi di dalam proses segmentasi. Nilai *hue* ini direpresentasikan dalam bentuk lingkaran yang memiliki rentang berupa sudut antara 0° - 360° . Oleh karena elemen warna *hue* berupa lingkaran dan dituliskan dalam sudut, maka setiap operasi yang berkaitan dengan elemen warna *hue* merupakan operasi sudut.

Saturation merupakan salah satu elemen dalam ruang warna HSV yang mewakili tingkat intensitas warna. Pada tingkat kecerahan (*value*) yang sama, nilai *saturation* ini akan menggambarkan kedekatan suatu warna terhadap warna abu-abu. Nilai *saturation* memiliki rentang antara 0 (minimum) - 1 (maksimum). Nilai toleransi *saturation* mempengaruhi tingkat kemurnian warna yang ikut terseleksi dalam proses segmentasi.

Value merupakan salah satu elemen dalam ruang warna HSV yang mewakili tingkat kecerahan warna. Ketika nilai *value* maksimum (1), warna yang dihasilkan adalah warna putih, sedangkan pada nilai *value* minimum (0), warna yang dihasilkan adalah warna hitam. Nilai toleransi *value* mempengaruhi tingkat kecerahan warna objek yang ikut terseleksi dalam proses segmentasi warna. Gambar 2.4 menunjukkan diagram dari ruang warna HSV.



Gambar 2.4 Diagram ruang warna HSV

Pada suatu citra, yang terdapat suatu objek dengan warna tertentu, warna tersebut terkadang tidak bernilai mutlak, akan tetapi dipengaruhi oleh intensitas cahaya yang dipantulkan oleh objek tersebut. Bentuk dan tekstur objek akan mempengaruhi nilai warna ini pada bagian tertentu sehingga objek di dalam citra akan memiliki intensitas warna dan gelap terang yang berbeda pada bagian tertentu. Proses segmentasi warna dan analisis objek yang bersangkutan dapat dioptimalkan dengan mengubah nilai toleransi pada setiap elemen warna *hue*, *saturation*, dan *value*.

2.2.7. OpenCV (*Open Source Computer Vision Library*)

OpenCV ini merupakan sebuah *open source library software API (Application Programming Interface)* dengan berlisensi BSD (*Berkeley Software Distribution*) yang digunakan untuk mengolah gambar maupun video secara *real-time* hingga mampu untuk mengekstraksi informasi di dalamnya. OpenCV *library* ini dapat dijalankan dalam berbagai bahasa pemrograman, seperti C, C++, Java, Python, serta mendukung berbagai *platform* seperti Windows, Linux, Mac OS, iOS, serta Android. Gambar 2.5 menunjukkan logo dari OpenCV.



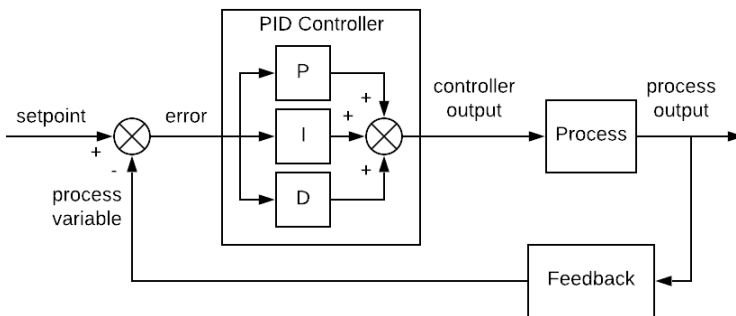
Gambar 2.5 Logo OpenCV

OpenCV ini didesain untuk komputasi yang efisien dan fokus pada aplikasi yang *real-time*. Selain memiliki komputasi yang efisien, OpenCV ini juga memiliki banyak fungsi untuk pemrosesan citra digital sehingga pemrosesan citra digital dapat menjadi lebih mudah. OpenCV memiliki sejumlah algoritma yang dioptimalkan dimana mencakup serangkaian algoritma visi komputer (*computer vision*) dan algoritma pembelajaran mesin (*machine learning*). Algoritma-algoritma dapat digunakan untuk mendeteksi dan mengenali wajah, mengklasifikasikan tindakan manusia, mengidentifikasi suatu objek, melacak objek bergerak, menggabungkan gambar, membandingkan gambar dengan *database* gambar yang dibuat, mengekstrak model objek tiga dimensi dari kamera stereo, dan lain-lain.

2.3. Sistem Kontrol PID (*Proportional-Integral-Derivative*)

2.3.1. Konsep Dasar Kontroler PID

Kontroler PID adalah kontroler yang digunakan untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik terdapat umpan balik (*feedback*) dalam sistem tersebut. Kontroler PID secara kontinu menghitung nilai kesalahan (*error*) sebagai selisih antara nilai referensi (*setpoint*) yang diinginkan dengan variabel proses terukur. Kontroler ini akan mencoba untuk meminimalkan nilai kesalahan setiap waktu dengan penyetelan variabel kontrol. Karena kontroler PID hanya mengandalkan variabel proses terukur dan bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyetelan (*tuning*) ketiga parameter (K_p , K_i , dan K_d), kontroler PID dapat memenuhi kebutuhan proses. Beberapa kriteria ini yang perlu diperhatikan dalam mendesain kontroler PID adalah *rise time*, *overshoot*, dan *steady-state error*.



Gambar 2.6 Diagram blok sistem kontrol PID

2.3.2. Kontrol Proporsional (P)

Kontrol proporsional (P) berlaku sebagai *gain* (penguat) saja tanpa memberikan sebuah efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian, di dalam aplikasi-aplikasi dasar yang sederhana, kontrol P cukup mampu untuk memperbaiki respon transien, khususnya *rise time* dan *settling time*. Kontrol P memiliki keluaran yang sebanding atau proporsional dengan besarnya sinyal kesalahan atau *error*. Nilai K_p yang besar akan menghasilkan perubahan yang besar pula pada keluaran untuk suatu nilai kesalahan yang tertentu. Namun jika nilai K_p terlalu besar, sistem akan membutuhkan waktu yang cukup lama untuk mencapai kondisi stabil (*steady-state*). Sebaliknya, nilai K_p yang terlalu kecil akan menghasilkan keluaran yang juga kecil sehingga kontroler ini menjadi kurang responsif/sensitif. Hal ini dapat mengakibatkan respon kontroler menjadi lebih lambat jika mendapatkan gangguan.

2.3.3. Kontrol Integral (I)

Kontrol integral (I) berfungsi untuk menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*steady-state error* = 0). Keluaran kontrol I ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Apabila sinyal kesalahan tidak mengalami perubahan, maka keluaran sistem akan menjaga keadaan seperti sebelum terjadinya perubahan pada masukan. Jika suatu kontroler tidak memiliki unsur integrator, kontrol proporsional tidak mampu menjamin keluaran sistem yang memiliki kesalahan keadaan mantap nol. Kontrol I dapat mempercepat perpindahan proses menuju *setpoint* beserta memperbaiki sekaligus menghilangkan *steady-state error* yang muncul pada kontrol P. Namun karena kontrol I merespon terhadap *error* yang terakumulasi dari sebelumnya, maka ini dapat menyebabkan *overshoot*. Pemilihan K_i yang tidak tepat menyebabkan respon transien yang tinggi dan menyebabkan ketidakstabilan sistem. Dan pemilihan K_i yang sangat tinggi justru dapat menyebabkan keluaran sistem berosilasi karena menambah orde sistem.

2.3.4. Kontrol Derivatif (D)

Kontrol derivatif (D) memiliki karakteristik yaitu perubahan yang mendadak pada masukan akan mengakibatkan perubahan yang sangat besar dan cepat pada keluaran. Apabila masukannya tidak mengalami perubahan, maka keluarannya juga tidak mengalami perubahan. Apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*),

maka akan menghasilkan sinyal keluaran bentuk impuls. Apabila sinyal masukan berubah naik secara perlahan (berbentuk fungsi *ramp*), maka keluarannya justru merupakan fungsi *step* yang magnitudonya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan faktor konstanta K_d . Dengan karakteristik tersebut, kontrol D ini dapat digunakan untuk memperbaiki respon dengan cara memprediksi *error* yang akan terjadi. Kontrol D dapat mencegah terjadinya *overshoot* yang diakibatkan oleh kontrol I. Kontrol D ini hanya berubah ketika terdapat perubahan *error* sehingga saat *error* statis, kontrol ini tidak akan bereaksi. Hal ini jugalah yang menyebabkan kontrol D tidak dapat digunakan sendiri.

Tabel 2.1 Efek kenaikan nilai parameter PID secara independen

Parameter	K_p	K_i	K_d
Rise Time	decrease	decrease	small change
Overshoot	increase	increase	decrease
Settling Time	small change	increase	decrease
Steady-State Error	decrease	eliminate	no effect in theory
Stability	degrade	degrade	improve if K_d small

2.3.5. Kontroler PID Digital

Pada awalnya, kontroler PID umumnya diimplementasikan dengan menggunakan rangkaian elektronika analog seperti op-amp (*operational amplifier*). Namun, seiring dengan berkembangnya dunia digital, seperti mikroprosesor dan mikrokontroler, maka kontroler PID tersebut dapat direalisasikan dalam bentuk persamaan PID digital. Implementasi dari persamaan ini berupa program yang diimplementasikan ke dalam sistem tertanam (*embedded system*) seperti mikroprosesor atau mikrokontroler. Persamaan PID digital secara umum adalah sebagai berikut :

$$u(n) = K_p e(n) + K_i \sum_{i=0}^n e(i) T + K_d \frac{e(n) - e(n-1)}{T} \dots\dots\dots (2.2)$$

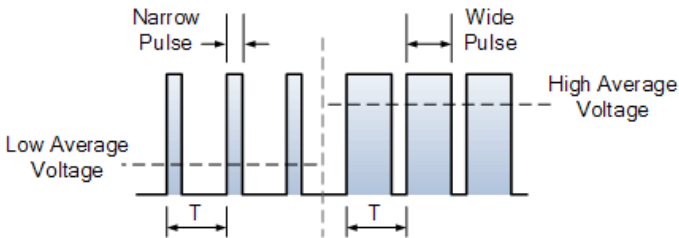
dimana u adalah nilai kontrol yang merupakan output dari kontroler PID digital, e adalah nilai *error* yang merupakan input dari kontroler PID digital, dan T adalah periode sampling *error*, dan K_p , K_i , K_d adalah nilai konstanta parameter kontroler PID digital yang akan disetel (*tuning*).

2.4. PWM (Pulse Width Modulation)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode untuk memperoleh nilai tegangan rata-rata yang berbeda. Sinyal PWM ini pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun mempunyai lebar pulsa yang bervariasi. Lebar pulsa sinyal PWM ini berbanding lurus dengan amplitudo dari sinyal asli yang belum termodulasi. Artinya, sinyal PWM memiliki frekuensi gelombang yang tetap, namun siklus kerja (*duty cycle*) yang bervariasi. *Duty cycle* adalah persentase lebar pulsa *high* dalam satu periode (dari 0 % hingga 100 %) yang dapat dirumuskan sebagai berikut :

$$D = \frac{t_{on}}{T} = \frac{t_{on}}{t_{on} + t_{off}} \dots\dots\dots (2.3)$$

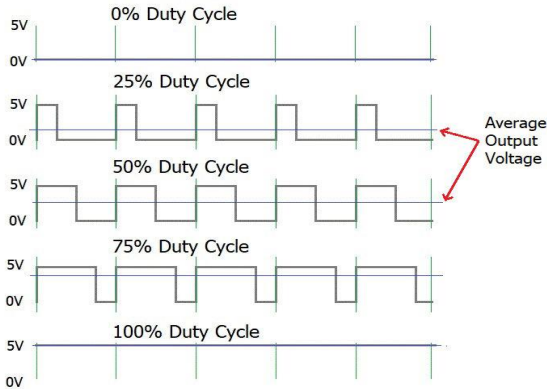
dimana D adalah *duty cycle*, T adalah periode sinyal PWM, t_{on} dan t_{off} adalah lebar pulsa *high* dan lebar pulsa *low*.



Gambar 2.7 Sinyal PWM dan tegangan rata-rata yang dihasilkan

PWM adalah salah satu teknik untuk mendapatkan sinyal analog dari sebuah piranti digital. Sebenarnya sinyal PWM dapat dibangkitkan dengan banyak cara, dapat menggunakan metode analog menggunakan rangkaian op-amp (*operational amplifier*) atau juga dapat menggunakan metode digital. Dengan metode analog, setiap perubahan sinyal PWM-nya sangat halus, sedangkan dengan metode digital, setiap perubahan sinyal PWM-nya dipengaruhi oleh resolusi dari sinyal PWM itu sendiri. Resolusi ini adalah jumlahnya variasi perubahan nilai pada sinyal PWM tersebut. Misalkan suatu sinyal PWM ini memiliki resolusi 8 bit, berarti sinyal PWM ini mempunyai variasi perubahan nilai sebanyak $2^8 = 256$

variasi, mulai dari 0 hingga 255 yang mewakili nilai *duty cycle* dari 0 % hingga 100 % pada keluaran sinyal PWM tersebut.



Gambar 2.8 Sinyal PWM dengan *duty cycle* yang berbeda-beda

Aplikasi PWM biasanya berupa pengendalian kecepatan motor DC, pengendalian posisi motor servo, pengendalian kecerahan dari LED, dan lain-lain. Pengendalian kecepatan dari motor DC dengan menggunakan sinyal PWM dapat dirumuskan sebagai berikut :

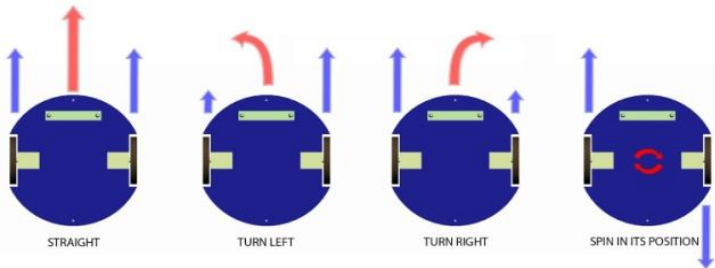
$$V_{out} = DV_{in} \dots\dots\dots (2.4)$$

dimana V_{out} adalah tegangan rata-rata yang diaplikasikan ke motor DC, D adalah nilai *duty cycle* dari sinyal PWM, dan V_{in} adalah tegangan referensi yang digunakan. Misalkan sinyal PWM memiliki *duty cycle* 75 % dan digunakan tegangan referensi 12 V, maka sesungguhnya motor DC akan berputar pada tegangan rata-rata yang bernilai 9 V.

2.5. Sistem Differential Drive pada Mobile Robot

Hal yang mendasar dari suatu *mobile robot* ialah mengenai sistem kemudi robot tersebut. Salah satu sistem kemudi yang dapat digunakan untuk pengoperasian suatu *mobile robot* dalam ruangan adalah sistem *differential drive*. Sistem *differential drive* adalah sistem dimana sebuah *mobile robot* tersusun atas dua buah *driving wheels* dengan tambahan *roller ball* sebagai penyeimbang. Sistem ini mengandalkan perbandingan kecepatan dari kedua roda untuk menentukan arah gerakan dari *mobile*

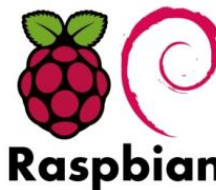
robot. Sistem tersebut sangat sensitif terhadap perbedaan kecepatan dari kedua roda karena dapat mempengaruhi secara langsung terhadap arah gerak *mobile robot*. Jika kedua roda penggerak tersebut berputar dengan kecepatan dan juga arah yang sama, maka robot tersebut akan bergerak dengan arah lurus. Jika kecepatan salah satu rodanya lebih lambat, maka robot akan bergerak dan membentuk kurva dengan arah lintasan menuju salah satu roda yang bergerak lebih lambat.



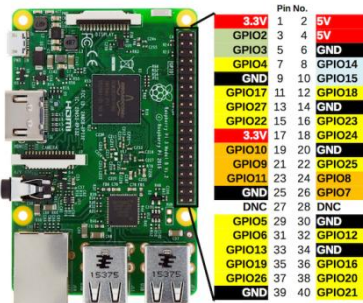
Gambar 2.9 Sistem *differential drive* pada *mobile robot*

2.6. Raspberry Pi

Raspberry Pi adalah sebuah komputer papan-tunggal (*single-board computer*) yang telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap dan menggunakan SoC (*System on a Chip*) Broadcom BCM dengan prosesor ARM yang dikemas dan terintegrasi di atas PCB. Perangkat ini menggunakan sebuah kartu SD (*SD card*) untuk *booting* dan penyimpanan jangka panjang. Raspberry Pi memiliki sejumlah fitur seperti *USB port*, *LAN port*, *CSI header*, *DSI header*, *GPIO header*, *SD card slot*, *memory chip*, dan *indicator LED*. Salah satu fitur Raspberry Pi yang kuat adalah deretan pin GPIO (*General-Purpose Input/Output*) yang berfungsi sebagai antarmuka fisik antara Raspberry Pi dan dunia luar yang dapat diakses secara *software* melalui program.



Gambar 2.10 Logo Raspbian



Gambar 2.11 Pin GPIO pada Raspberry Pi 3 model B+

Raspberry Pi bersifat *open source* berbasis Linux sehingga dapat dimodifikasi sesuai kebutuhan penggunanya. Sistem operasi utama dari Raspberry Pi adalah menggunakan Debian GNU/Linux dengan bahasa pemrograman Python. Salah satu dari pengembang sistem operasi untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian yang diklaim mampu untuk memaksimalkan perangkat Raspberry Pi ini. Raspbian dilengkapi dengan sejumlah paket atau *pre-compiled software* dalam format yang bagus untuk kemudahan instalasinya. Sistem operasi ini dibuat berbasis Debian, salah satu distribusi Linux OS.

2.7. Webcam (Web Camera)

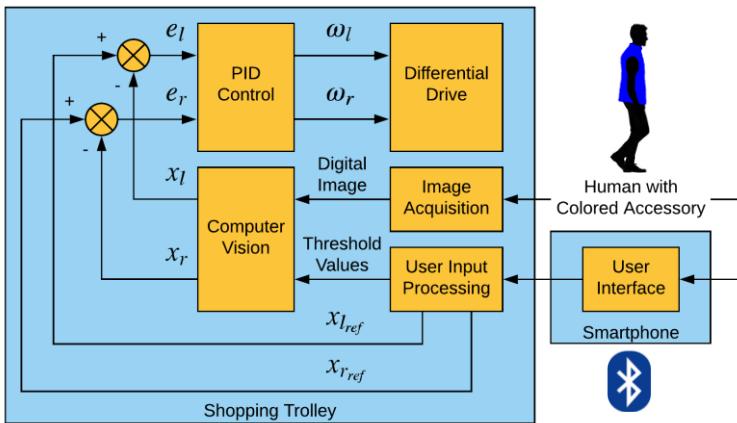
Webcam adalah jenis kamera kompak yang dapat dipasang pada komputer, baik menggunakan USB *port* atau secara *wireless*. *Webcam* digunakan untuk menangkap gambar maupun video secara *real-time* dan dapat ditampilkan secara langsung di komputer menggunakan *software* tertentu. Mirip dengan kamera digital pada umumnya, *webcam* bekerja dengan cara menangkap cahaya yang melalui lensa berukuran kecil pada bagian depan, dengan menggunakan detektor cahaya mikroskopik yang terpasang pada *microchip* penerima gambar. Detektor ini pada umumnya berteknologi CMOS (*Complementary Metal-Oxide Semiconductor*) atau CCD (*Charge-Coupled Device*). Gambar atau video yang ditangkap oleh *webcam* ini dapat langsung diproses secara digital dengan menggunakan komputer. Pada umumnya, *webcam* ini dilengkapi dengan sejumlah fitur seperti mikrofon, sensor *built-in*, lampu indikator, dan dapat disesuaikan posisinya. Beberapa jenis *webcam* diantaranya adalah *serial and parallel port webcam*, *USB webcam*, *firewire and card-based webcam*, *network and wireless webcam*.

BAB III PERANCANGAN SISTEM

Pada Bab Perancangan Sistem dibahas mengenai penjelasan dalam perancangan sistem yang meliputi perancangan *hardware* dan *software* dari penelitian yang dilakukan dalam tugas akhir.

3.1. Diagram Blok Sistem

Gambar 3.1 menunjukkan diagram blok sistem secara keseluruhan. Secara umum, sistem menerima masukan berupa citra manusia dengan aksesoris berwarna dan perintah melalui aplikasi *smartphone*. Sistem ini menghasilkan keluaran berupa gerak mengikuti manusia sesuai dengan perintah yang diberikan. Setiap blok pada sistem akan dijelaskan lebih lanjut pada subbab-subbab berikutnya.



Gambar 3.1 Diagram blok sistem secara keseluruhan

Pada sistem ini, manusia memberikan perintah kepada troli belanja melalui antarmuka pengguna berupa aplikasi pada *smartphone*. Perintah ini diproses dan diterjemahkan menjadi nilai-nilai yang berguna, yaitu nilai *threshold* warna serta nilai absis piksel minimum dan maksimum referensi. Citra manusia dengan aksesoris berwarna diakuisisi sehingga dihasilkan citra digital yang dapat diproses dengan visi komputer. Pada pemrosesan visi komputer digunakan nilai *threshold* warna yang telah

didapatkan sebelumnya. Hasil pemrosesan ini berupa nilai absis piksel minimum dan maksimum. Kedua nilai ini dikurangkan dari nilai absis piksel minimum dan maksimum referensi dan menghasilkan nilai *error* absis piksel minimum dan maksimum. Kedua nilai ini diproses dengan kontrol PID dan dihasilkan nilai kecepatan rotasi kiri dan kanan yang digunakan pada sistem penggerak dua roda atau *differential drive*.

3.2. Perancangan Troli Belanja

Perancangan troli belanja terdiri dari perancangan mekanik dan juga perancangan elektronik. Troli belanja yang dirancang ini terdiri dari troli dan keranjang belanja. Troli ini memiliki dimensi 45 x 45 x 90 cm dan keranjang belanja memiliki dimensi 35 x 23 x 21 cm. Ketinggian letak keranjang belanja dari dasar troli adalah 50 cm. Troli belanja memiliki empat roda yang menggunakan sistem penggerak dua roda (*differential drive*). Dua roda pada bagian belakang memiliki diameter 12 cm dengan menggunakan sistem *fixed wheel*, yaitu sistem dimana poros putar roda ini menempel pada kerangka troli belanja sehingga roda ini hanya dapat berputar ke arah depan atau belakang. Kedua roda ini dipasangkan ke DC *gear motor* yang berfungsi sebagai penggerak roda. Jarak di antara kedua roda belakang ini adalah 35 cm. Dua roda bagian depan memiliki diameter 10 cm dengan menggunakan sistem *orientable wheel*, yaitu sistem dimana orientasi atau arah roda ini dapat dikendalikan sehingga dapat memudahkan dalam pengendalian gerak troli belanja. Kedua roda ini difungsikan sebagai roda penyeimbang dari troli belanja. Gambar 3.2 menunjukkan perancangan mekanik troli belanja.



Gambar 3.2 Perancangan mekanik troli belanja

Troli belanja dilengkapi dengan komponen-komponen elektronik yang meliputi USB *webcam*, Raspberry Pi, *push button*, *buzzer*, modul Bluetooth, *motor driver*, DC *gear motor*, dan baterai. USB *webcam* ini diletakkan pada troli belanja pada ketinggian 90 cm dengan tujuan untuk mendapatkan jangkauan pandangan yang lebih luas. USB *webcam* ini digunakan untuk menangkap citra secara digital sehingga dapat diproses lebih lanjut dengan Raspberry Pi. Peneliti menggunakan USB *webcam* Logitech C170 dengan spesifikasi yang ditunjukkan pada tabel 3.1. USB *webcam* ini dipilih karena memiliki resolusi dan FPS yang cukup baik dan mudah dalam pemasangannya. USB *webcam* dapat dihubungkan ke Raspberry Pi secara langsung melalui USB *port*.

Tabel 3.1 Spesifikasi USB *webcam* Logitech C170

USB Type	USB 2.0
Focus Type	Fixed
Focal Length	2.3 mm
Diagonal FoV	58°
Image Capture	640 x 480, 1.3 MP, 3 MP, 5 MP
Video Capture	320 x 240, 640 x 480, 1024 x 768
Frame Rate (max)	640 x 480 @ 30
Microphone	Mono
Indicator LED	Yes
Adjustable Clip	Yes
CPU Minimum	1 GHz (1.6 GHz Recommended)
RAM Minimum	512 MB (1 GB Recommended)

Komputer papan tunggal berjenis Raspberry Pi ini digunakan untuk menjalankan proses visi komputer serta algoritma kontrol gerak. Selain itu, Raspberry Pi ini juga dimanfaatkan untuk mengontrol komponen-komponen elektronik yang lainnya melalui pin GPIO yang dimilikinya. Raspberry Pi ini diletakkan di dekat USB *webcam* untuk memudahkan pemasangannya dikarenakan panjang kabel USB *webcam* yang terbatas. Peneliti menggunakan Raspberry Pi 3 model B+ dengan spesifikasi yang ditunjukkan pada tabel 3.2. Raspberry Pi 3 dipilih dikarenakan memiliki kualitas yang lebih baik dengan fitur-fitur yang lebih lengkap daripada model-model sebelumnya.

Tabel 3.2 Spesifikasi Raspberry Pi 3 model B+

SoC Type (Processor)	Broadcom BCM2837B0
Core Type	Cortex-A53 64-bit
Number of Cores	4 Cores
GPU	VideoCore IV @ 400 MHz
CPU Clock	1.4 GHz
Memory	Micro SD
RAM	1 GB LPDDR2
USB Port	4 x USB 2.0
Ethernet	Gigabit over USB 2.0 @ 300 Mbps
HDMI	1 x Full Size HDMI
WiFi	Dual-Band 2.4 GHz and 5 GHz IEEE 802.11 b/g/n/ac Wireless LAN
Bluetooth	Bluetooth 4.2
Antenna	PCB Antenna
GPIO	40 Pins
Operating System	Latest Raspbian
Power over Ethernet	Yes, with PoE HAT
Power Rating	1.13 A @ 5 V
Dimension	85 x 56 mm

Pada troli belanja ini ditambahkan *push button* untuk memudahkan pengoperasian troli belanja. *Push button* digunakan sebagai antarmuka fisik baik untuk mengaktifkan atau menonaktifkan troli belanja. Peneliti menggunakan *normally open push button* yang memiliki prinsip kerja akan terhubung jika ditekan. Pada troli belanja juga ditambahkan *buzzer* yang digunakan sebagai indikator. Bunyi dari *buzzer* tersebut digunakan untuk mengindikasikan bahwa troli belanja telah menerima perintah dari manusia. Bunyi *buzzer* juga digunakan untuk mengindikasikan jika troli belanja tidak mampu mendeteksi aksesoris berwarna yang dipakai oleh manusia. Peneliti menggunakan *active buzzer* karena memiliki osilator internal di dalamnya sehingga hanya membutuhkan tegangan DC untuk dapat menghasilkan bunyi. Modul Bluetooth digunakan untuk menerima perintah yang diberikan oleh manusia secara *wireless* dengan teknologi

komunikasi Bluetooth melalui antarmuka aplikasi *smartphone*. Peneliti menggunakan modul Bluetooth HC-05 karena modul ini mudah untuk diaplikasikan juga kualitas komunikasinya cukup baik. Konfigurasi pin modul Bluetooth HC-05 ditunjukkan pada tabel 3.3.

Tabel 3.3 Konfigurasi pin modul Bluetooth HC-05

Modul Bluetooth HC-05	Raspberry Pi 3 model B+
EN	-
VCC	3.3 V
GND	GND
TX	GPIO15
RX	GPIO14
STATE	GPIO18

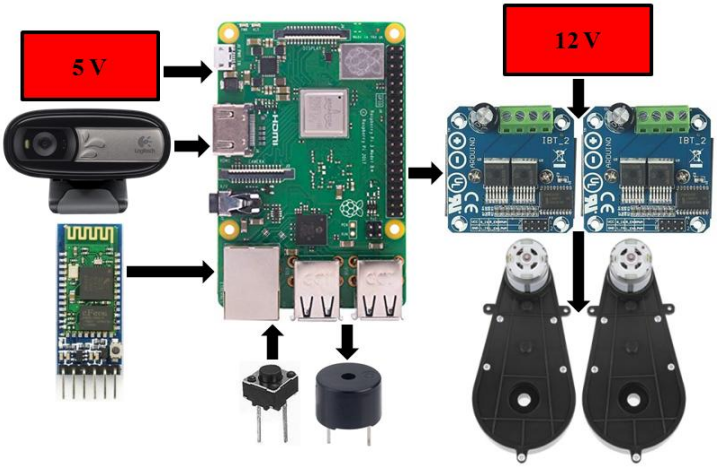
Penggerak troli belanja meliputi *motor driver* dan DC *gear motor*. *Motor driver* digunakan untuk mengontrol nilai tegangan rata-rata yang diberikan kepada DC *gear motor*. Nilai tegangan rata-rata ini bergantung pada sinyal PWM pada frekuensi dan *duty cycle* tertentu yang dihasilkan Raspberry Pi. Peneliti menggunakan modul *motor driver* IBT-2 karena dapat dikontrol melalui Raspberry Pi yang hanya mempunyai tegangan kerja pin 3.3 V. Selain itu, modul ini juga mampu menghantarkan arus hingga 43 A dan juga dilengkapi dengan proteksi arus berlebih. DC *gear motor* digunakan untuk menggerakkan kedua roda belakang troli belanja dengan kecepatan rotasi yang bergantung pada tegangan rata-rata yang diberikan. Peneliti menggunakan DC *gear motor* RS390PH spesifikasi 12 V 20000 RPM. DC *gear motor* ini dilengkapi dengan *gearbox* yang bertujuan untuk menurunkan RPM motor namun menaikkan torsi motor. Kedua DC *gear motor* ini dipasangkan pada kedua roda belakang troli belanja. Gerakan troli belanja ini dapat dikendalikan dengan mengontrol kecepatan rotasi dari masing-masing DC *gear motor* ini. Konfigurasi pin modul *motor driver* IBT-2 ditunjukkan pada tabel 3.4.

Tabel 3.4 Konfigurasi pin modul *motor driver* IBT-2

Modul <i>motor driver</i> IBT-2	Kiri	Kanan
GND	GND	GND
VCC	3.3 V	3.3 V

LIS	-	-
RIS	-	-
LEN	GPIO6	GPIO12
REN	GPIO13	GPIO16
LPWM	GPIO19	GPIO20
RPWM	GPIO26	GPIO21

Baterai digunakan untuk mensuplai Raspberry Pi dan juga DC gear motor. Peneliti menggunakan dua jenis baterai. Suplai untuk Raspberry Pi diambil dari *power bank* dengan spesifikasi 5 V 2 A 24 AH. Suplai untuk DC gear motor ini diambil dari baterai *sealed-lead-acid* 6-DFM-9 dengan spesifikasi 12 V 9 AH / 20 HR. Baterai jenis ini dipilih karena memiliki kapasitas yang cukup besar dan tegangan yang sesuai dengan tegangan DC gear motor yang digunakan. Baterai ini memiliki dimensi 15 x 6.5 x 9.5 cm dengan berat sekitar 3 kg. Gambar 3.3 menunjukkan perancangan elektronik troli belanja.

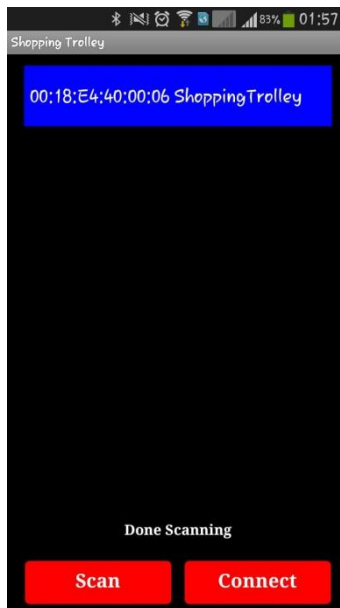


Gambar 3.3 Perancangan elektronik troli belanja

3.3. Perancangan Antarmuka Pengguna

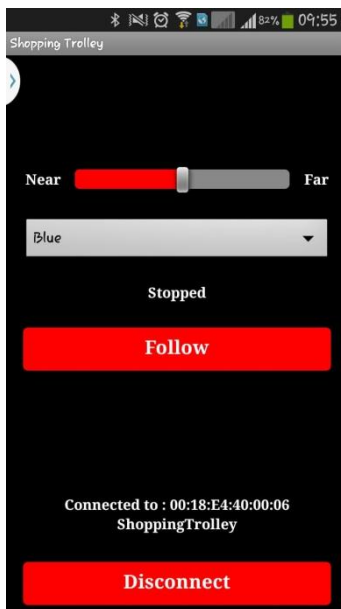
Perancangan antarmuka pengguna bertujuan untuk mempermudah pengguna dalam berinteraksi dengan troli belanja. Antarmuka pengguna

yang dirancang berupa aplikasi yang terdapat pada *smartphone* dengan sistem operasi Android. Untuk bisa menggunakan aplikasi ini, pengguna perlu mengaktifkan perangkat Bluetooth di *smartphone* terlebih dahulu serta melakukan *pairing* dengan perangkat Bluetooth pada troli belanja. Aplikasi yang dirancang ini memiliki dua halaman utama, yaitu halaman koneksi dan halaman perintah. Pada halaman koneksi ini disediakan dua tombol, yaitu tombol *scan* dan tombol *connect*. Tombol *scan* berfungsi untuk menampilkan seluruh perangkat Bluetooth yang telah ter-*pairing* dengan perangkat Bluetooth di *smartphone* pengguna. Tombol *connect* berfungsi untuk mengkoneksikan perangkat Bluetooth pada *smartphone* pengguna dengan perangkat Bluetooth di troli belanja yang digunakan. Halaman ini juga menampilkan status sebagai umpan balik untuk setiap proses yang terjadi. Peneliti merancang antarmuka pengguna ini dengan tampilan yang sederhana agar aplikasi mudah untuk digunakan. Peneliti menggunakan komunikasi Bluetooth karena sederhana sehingga mudah untuk digunakan dalam aplikasi *smartphone*. Gambar 3.4 menunjukkan tampilan halaman koneksi.



Gambar 3.4 Tampilan halaman koneksi pada aplikasi *smartphone*

Setelah koneksi antara perangkat Bluetooth *smartphone* pengguna dengan perangkat Bluetooth troli belanja berhasil, maka halaman beralih ke halaman perintah. Pada halaman perintah tersedia *trackbar* jarak, *list* warna, tombol *follow/stop*, dan tombol *disconnect*. *Trackbar* jarak ini digunakan untuk mengatur jarak pengikutan troli belanja. Troli belanja akan mempertahankan jarak ini selama mengikuti pengguna. Pengaturan jarak diperlukan mengingat aksesoris berwarna yang dipakai pengguna dapat bermacam-macam ukurannya sehingga perlu dilakukan kalibrasi agar troli belanja dapat bergerak mengikuti pengguna dengan jarak yang dikehendaki. *List* warna berisi daftar warna aksesoris yang dipakai oleh pengguna. Warna yang dipilih oleh pengguna menjadi warna yang akan terdeteksi oleh troli belanja. Karena itu, pengguna perlu untuk memilih warna sesuai dengan warna aksesoris yang dipakai. Tombol *follow/stop* berfungsi untuk menjalankan atau menghentikan troli belanja. Tombol *disconnect* berfungsi untuk memutus koneksi antara perangkat Bluetooth *smartphone* pengguna dan perangkat Bluetooth pada troli belanja yang digunakan. Gambar 3.5 menunjukkan tampilan halaman perintah.



Gambar 3.5 Tampilan halaman perintah pada aplikasi *smartphone*

Perintah yang diberikan kepada troli belanja ini dikirimkan dalam bentuk teks yang terdiri dari perintah, warna, dan jarak yang dipisahkan menggunakan tanda koma. Teks dikirimkan kepada troli belanja secara *wireless* menggunakan teknologi komunikasi Bluetooth. Teks dilakukan pengkodean karakter UTF-8, yaitu setiap karakternya dikirimkan dalam bentuk bilangan biner 8-bit. Sebagai contoh, perintah untuk mendeteksi dan mengikuti warna biru pada jarak sejauh 50% dari jarak maksimum dikirimkan dalam bentuk teks ini “Follow,Blue,50”, sedangkan perintah untuk menghentikan dikirimkan dalam bentuk teks ini “Stop”.

3.4. Pengolahan Input Pengguna

Input pengguna yang diberikan melalui aplikasi pada *smartphone* diterima oleh modul Bluetooth. Modul ini selanjutnya meneruskan input yang diberikan ke Raspberry Pi untuk diolah. Komunikasi antara modul Bluetooth dengan Raspberry Pi menggunakan UART dengan 9600 bps, tanpa bit paritas, dan dengan 1 bit stop. Untuk menjalankan komunikasi tersebut, peneliti menggunakan *serial port* ttyAMA0 pada Raspberry Pi karena memiliki kualitas komunikasi yang lebih baik daripada ttyS0.

Di subbab yang sebelumnya telah dijelaskan bahwa perintah yang dikirimkan oleh pengguna melalui aplikasi *smartphone* ini berupa teks. Teks ini merupakan input untuk sistem troli belanja dan diproses untuk menghasilkan nilai-nilai yang berguna untuk proses selanjutnya. Peneliti menggunakan metode *parsing* pada teks berdasarkan tanda koma untuk mendapatkan perintah, warna, serta jarak secara terpisah. Perintah ini menentukan apakah troli belanja akan dijalankan atau dihentikan. Warna ini diterjemahkan menjadi nilai *threshold* warna yang sesuai yang akan digunakan pada proses visi komputer. Jarak diterjemahkan menjadi nilai absis piksel minimum dan maksimum referensi ($x_{l_{ref}}$ dan $x_{r_{ref}}$) yang akan digunakan dalam kontrol gerak. Kedua nilai ini digunakan karena dapat merepresentasikan jarak antara troli belanja dan pengguna.

3.5. Proses Akuisisi Citra

Setelah troli belanja diperintahkan untuk mengikuti manusia, maka dilanjutkan dengan proses akuisisi citra. Proses akuisisi citra dilakukan menggunakan USB *webcam* untuk menghasilkan citra digital yang dapat diproses secara langsung oleh Raspberry Pi. Citra yang diakuisisi adalah citra manusia dengan aksesoris berwarna yang terdapat pada depan troli belanja. Citra digital yang dihasilkan ini berada dalam ruang warna RGB

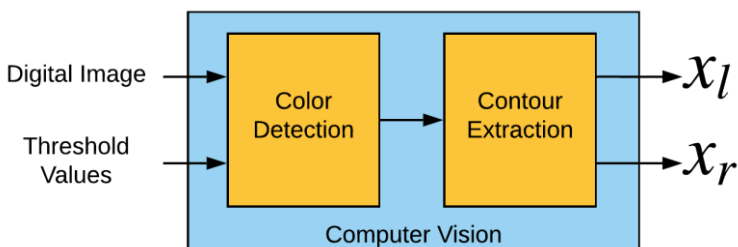
yang memiliki resolusi 640 x 480 piksel. Gambar 3.6 menunjukkan hasil proses akuisisi citra.



Gambar 3.6 Hasil proses akuisisi citra

3.6. Proses Visi Komputer

Citra digital hasil dari proses akuisisi citra diolah lebih lanjut pada proses visi komputer. Proses visi komputer yang digunakan ini meliputi deteksi warna dan ekstraksi kontur. Gambar 3.7 menunjukkan diagram blok dari proses visi komputer. Secara umum, proses visi komputer ini menerima input berupa citra digital dan nilai *threshold* warna yang telah ditentukan. Proses visi komputer menghasilkan output berupa nilai absis piksel minimum dan maksimum yaitu x_l dan x_r . Peneliti memanfaatkan *OpenCV library* dalam menjalankan proses visi komputer.



Gambar 3.7 Diagram blok proses visi komputer

3.6.1. Proses Deteksi Warna

Proses deteksi warna yang digunakan meliputi pengubahan ukuran citra, pengaburan citra, konversi ruang warna citra, pengambangan citra,

dan transformasi morfologi citra. Pengubahan ukuran citra menjadi lebih kecil dimaksudkan untuk mempersingkat waktu yang diperlukan untuk pemrosesan citra selanjutnya. Citra digital hasil dari proses akuisisi citra memiliki resolusi 640 x 480 piksel. Peneliti memperkecil ukuran citra ini menjadi citra dengan resolusi 160 x 120 piksel dimana ukuran citra ini masih memiliki *aspect ratio* yang sama seperti ukuran sebelumnya. Pemrosesan citra yang memiliki jumlah piksel yang lebih sedikit akan memiliki waktu komputasi yang lebih singkat karena jumlah iterasi yang terjadi lebih sedikit. Dengan demikian, trolis belanja mampu merespon lebih cepat terhadap gerakan manusia. Gambar 3.8 menunjukkan hasil proses pengubahan ukuran citra.



Gambar 3.8 Hasil proses pengubahan ukuran citra

Setelah proses pengubahan ukuran citra, maka dilanjutkan dengan proses pengaburan citra. Proses ini dilakukan untuk mengurangi *noise* berupa detail-detail pada citra yang tidak diperlukan. Selain itu, proses ini juga dimanfaatkan untuk memperhalus transisi nilai intensitas warna antar pikselnya sehingga dapat menghilangkan piksel-piksel *outlier* pada citra. Peneliti menggunakan metode pengaburan Gaussian karena dapat menghilangkan tepi-tepi objek yang terdapat dalam citra sehingga tidak mengganggu proses-proses selanjutnya. Dengan demikian, pengaburan Gaussian difungsikan sebagai *low pass filter* sehingga frekuensi tinggi berupa perubahan intensitas warna antar piksel citra yang terjadi secara mendadak dapat dihilangkan. Peneliti menggunakan ukuran kernel 5 x 5 sebagai parameter fungsi pengaburan Gaussian karena ukuran ini sudah cukup bagus untuk menghilangkan *noise* tanpa banyak mempengaruhi informasi penting yang terdapat dalam citra. Gambar 3.9 menunjukkan hasil proses pengaburan citra.



Gambar 3.9 Hasil proses pengaburan citra

Setelah proses pengaburan citra, maka dilanjutkan dengan proses konversi ruang warna citra. Citra digital hasil dari proses akuisisi citra masih berada dalam ruang warna RGB. Ruang warna RGB ini diketahui kurang cocok jika digunakan dalam proses pengembangan citra karena ketidakmampuannya dalam mentolerir perubahan intensitas cahaya yang terjadi. Selain itu, menentukan nilai *threshold* warna RGB cukup sulit karena membutuhkan kalibrasi dari tiga kanal warna. Karena itu, peneliti melakukan konversi citra dalam ruang warna RGB menjadi citra dalam ruang warna HSV. Ruang warna HSV hanya memiliki satu kanal yang menentukan warna, yaitu komponen *Hue* sehingga dapat memudahkan kalibrasi untuk menentukan nilai *threshold* warna HSV. Selain itu, ruang warna ini memiliki komponen *Saturation* dan *Value* yang dapat diatur sedemikian rupa sehingga mampu mentolerir pengaruh dari perubahan intensitas cahaya yang terjadi. Gambar 3.10 menunjukkan hasil proses konversi ruang warna citra.



Gambar 3.10 Hasil proses konversi ruang warna citra

Setelah proses konversi ruang warna pada citra, maka dilanjutkan dengan proses pengambangan citra. Dalam proses, nilai intensitas warna tiap piksel dalam citra dibandingkan dengan nilai *threshold* warna yang telah ditentukan. Nilai *threshold* warna ini terdiri dari nilai *threshold* bawah dan nilai *threshold* atas. Jika nilai intensitas warna piksel berada di luar rentang nilai ini, maka intensitas warna piksel tersebut digantikan menjadi warna hitam. Sebaliknya, apabila nilai intensitas warna piksel berada di dalam rentang nilai ini, maka intensitas warna piksel tersebut digantikan menjadi warna putih. Dengan begitu, proses pengambangan citra menghasilkan citra biner yang meliputi *object* berwarna putih dan *background* berwarna hitam. *Object* ini berbentuk *blob* berwarna putih yang merepresentasikan benda-benda dengan warna yang telah dipilih, termasuk aksesoris berwarna yang dipakai oleh manusia. Gambar 3.11 menunjukkan hasil proses pengambangan citra.



Gambar 3.11 Hasil proses pengambangan citra

Setelah proses pengambangan pada citra, maka dilanjutkan dengan proses transformasi morfologi citra. Peneliti menggunakan transformasi morfologi dengan operasi *opening* dan *closing*. Operasi *opening* adalah operasi erosi dilanjutkan dengan operasi dilatasi. Operasi *closing* adalah operasi dilatasi dilanjutkan dengan operasi erosi. Kedua operasi tersebut digunakan untuk menghilangkan *noise* yang berupa *blob outlier* dimana *blob* ini bukan bagian dari *blob* utama dan untuk menyatukan *blob* yang terpisah. Peneliti menggunakan kedua operasi ini dengan ukuran kernel 5×5 yang berbentuk persegi. Proses transformasi morfologi citra dapat menghasilkan citra biner yang memiliki *blob* yang telah disederhanakan untuk mempercepat proses ekstraksi kontur. Gambar 3.12 menunjukkan hasil proses transformasi morfologi citra.



Gambar 3.12 Hasil proses transformasi morfologi citra

3.6.2. Proses Ekstraksi Kontur

Proses ekstraksi kontur yang digunakan meliputi pencarian kontur, seleksi luas kontur, seleksi *centroid* kontur, dan pengambilan nilai absis piksel minimum dan maksimum penyusun kontur. Pencarian kontur ini dilakukan untuk mendata seluruh koordinat piksel yang mengitari *blob*. Peneliti mencari kontur eksternal *blob* menggunakan pendekatan rantai sederhana sehingga hanya mendata koordinat-koordinat piksel tertentu, dimana koordinat-koordinat ini sudah cukup digunakan untuk mewakili kontur yang dicari. Pendekatan tersebut bertujuan untuk mempersingkat waktu pencarian, mengingat tidak seluruh koordinat piksel perlu untuk didata. Koordinat-koordinat piksel yang telah diperoleh digunakan untuk menghitung luas kontur dan koordinat *centroid* kontur pada citra. Kedua nilai ini akan digunakan lebih lanjut pada proses seleksi kontur berdasar luas kontur dan koordinat *centroid* kontur. Gambar 3.13 menunjukkan hasil proses pencarian kontur.



Gambar 3.13 Hasil proses pencarian kontur

Setelah proses pencarian kontur, maka dilanjutkan dengan proses seleksi luas kontur. Pada proses ini, luas setiap kontur yang didapatkan dihitung dan dibandingkan dengan suatu nilai tertentu. Jika luas kontur kurang dari nilai ini, maka kontur tersebut diabaikan dan tidak diproses lebih lanjut. Dan sebaliknya, jika luas kontur lebih dari nilai ini, maka kontur tersebut dipilih dan diproses lebih lanjut. Dengan adanya proses ini, maka waktu yang diperlukan untuk proses selanjutnya dapat menjadi lebih singkat karena hanya kontur dengan luas tertentu yang diproses. Peneliti menggunakan nilai 500 sebagai nilai luas kontur referensi untuk proses seleksi luas kontur. *Noise* yang berupa *blob* kecil dengan ukuran luas lebih kecil dari nilai ini diabaikan dan tidak diproses lebih lanjut. Gambar 3.14 menunjukkan hasil proses seleksi luas kontur.



Gambar 3.14 Hasil proses seleksi luas kontur

Setelah proses seleksi luas kontur, maka dilanjutkan dengan proses seleksi *centroid* kontur. Proses ini bertujuan untuk menghindari adanya *target switching* apabila terdapat dua kontur atau lebih. *Target switching* perlu dihindari karena bisa mengganggu proses selanjutnya. Pada proses seleksi *centroid* kontur ini, peneliti membandingkan koordinat *centroid* kontur yang diperoleh pada setiap citra yang berurutan. Perbandingan ini berupa jarak Euclidean antara dua koordinat *centroid* kontur pada citra yang berurutan. Peneliti memilih kontur yang memiliki jarak Euclidean minimum karena kontur tersebut masih relevan dengan kontur pada citra sebelumnya sehingga kontur tersebut dipilih untuk diproses lebih lanjut. Dengan demikian, proses seleksi *centroid* kontur ini dapat dimanfaatkan untuk mencari kontur pada citra yang masih relevan dengan kontur pada citra yang telah diperoleh sebelumnya. Gambar 3.15 menunjukkan hasil proses seleksi *centroid* kontur.



Gambar 3.15 Hasil proses seleksi *centroid* kontur

Setelah proses seleksi *centroid* kontur, maka dilakukan tahap akhir, yaitu pengambilan nilai absis piksel minimum dan maksimum penyusun kontur. Pada proses ini, setiap piksel yang menyusun kontur didapatkan dan hanya diambil nilai absis piksel minimum dan maksimum. Dengan kata lain, nilai absis piksel minimum dan maksimum ini merupakan nilai absis piksel yang terletak paling kiri (x_l) dan paling kanan (x_r). Kedua nilai ini akan digunakan lebih lanjut pada proses kontrol PID. Gambar 3.16 menunjukkan hasil proses pengambilan nilai absis piksel minimum dan maksimum penyusun kontur.



Gambar 3.16 Hasil proses pengambilan nilai x_l dan x_r .

3.7. Proses Kontrol PID

Pada proses kontrol PID, peneliti menggunakan dua kontroler PID, yaitu kontroler PID untuk mengontrol nilai x_l dan kontroler PID untuk mengontrol nilai x_r . Dua kontroler ini menerima masukan berupa nilai *error*, yaitu hasil pengurangan nilai $x_{l_{ref}}$ dan $x_{r_{ref}}$ yang telah ditentukan

dengan nilai x_l dan x_r yang telah diperoleh. Nilai *error* yang dihasilkan adalah e_l dan e_r . Kedua nilai *error* ini diproses oleh kontroler PID untuk menghasilkan nilai kontrol yang sesuai. Untuk parameter kontroler PID, peneliti menggunakan nilai 0.5 untuk k_p , nilai 0.3 untuk k_i , dan nilai 0.1 untuk k_d . Nilai-nilai parameter tersebut didapatkan oleh peneliti dengan cara melakukan *tuning* secara manual. Untuk periode sampling, peneliti menggunakan nilai sebesar 0.2 detik. Nilai ini telah disesuaikan dengan waktu maksimum yang diperlukan program dalam sekali iterasi.

Kontroler PID ini bekerja dengan membawa nilai x_l dan x_r menuju nilai $x_{l_{ref}}$ dan $x_{r_{ref}}$ dengan cara memberikan nilai kontrol pada sistem. Nilai kontrol yang digunakan oleh peneliti berupa nilai *duty cycle* sinyal PWM dengan rentang dari 0 hingga 100. Dengan demikian, keluaran ini dapat digunakan untuk mengontrol *motor driver* dan *motor driver* akan mengontrol nilai tegangan rata-rata yang diberikan ke DC *gear motor*.

3.8. Proses *Differential Drive*

Pada proses *differential drive*, nilai kontrol yang berupa *duty cycle* sinyal PWM hasil dari proses kontrol PID digunakan untuk mengontrol kecepatan rotasi kedua DC *gear motor* yang terpasang pada kedua roda belakang troli belanja. Kedua nilai tersebut digunakan sebagai parameter sinyal PWM dan dikeluarkan melalui pin GPIO pada Raspberry Pi yang terhubung ke *motor driver*. Peneliti menggunakan sinyal PWM dengan frekuensi 1 kHz. Sinyal PWM tersebut mengontrol besarnya tegangan rata-rata yang diberikan kepada DC *gear motor* melalui *motor driver*. Dengan kata lain, pada proses *differential drive* ini, kedua nilai *duty cycle* dikonversikan menjadi nilai kecepatan rotasi, yaitu ω_l untuk DC *gear motor* kiri dan ω_r untuk DC *gear motor* kanan. Perbedaan antara kedua nilai ini digunakan untuk mengendalikan gerakan troli belanja.

Peneliti menggunakan sistem penggerak dua roda atau *differential drive* karena sederhana, baik dari sisi mekanik maupun elektronik. Pada sistem ini, arah gerak troli belanja dikendalikan berdasarkan perbedaan kecepatan rotasi antara roda kiri dan roda kanan troli belanja. Jika nilai ω_l lebih besar dari nilai ω_r , maka troli belanja akan bergerak ke kanan. Dan sebaliknya, jika nilai ω_r lebih besar dari ω_l , maka troli belanja akan bergerak ke kiri. Jika nilai ω_l sama dengan nilai ω_r , maka troli belanja akan bergerak lurus.

----- *Halaman ini sengaja dikosongkan* -----





BAB IV PENGUJIAN DAN ANALISIS

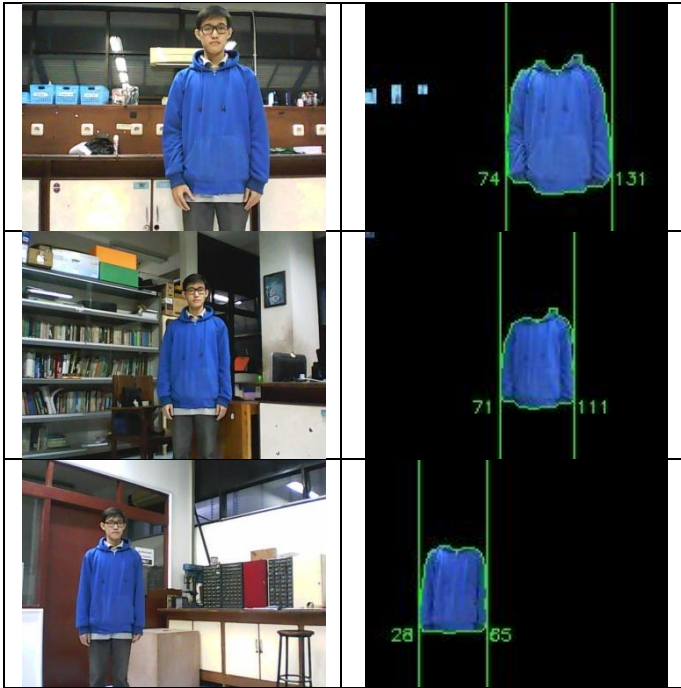
Pada Bab Pengujian dan Analisis ini dibahas mengenai penjelasan dalam pengujian sistem yang meliputi proses akuisisi data serta analisis data dari penelitian yang dilakukan dalam tugas akhir. Pengujian terdiri dari pengujian visi troli belanja, pengujian kecepatan troli belanja, juga pengujian jarak troli belanja, serta pengujian kontrol gerak troli belanja.

4.1. Pengujian Visi Troli Belanja

Pengujian visi troli belanja ini dilakukan untuk mengetahui kinerja dari proses visi komputer. Pada pengujian ini, peneliti berdiri di depan troli belanja dalam beberapa lingkungan yang berbeda. Perbedaan yang dimaksud ini adalah perbedaan pencahayaan ruang dan perbedaan jarak. Peneliti memakai aksesoris berupa jaket berwarna biru. Lebar jaket ini ketika dipakai oleh peneliti adalah sekitar 50 cm. Peneliti menggunakan aksesoris ini dengan alasan karena warna yang kontras sehingga mudah untuk dideteksi serta ukurannya yang besar sehingga mudah dibedakan dengan *noise* yang terdapat di sekitar peneliti. Hasil pengujian visi troli belanja ditunjukkan pada tabel 4.1.

Tabel 4.1 Hasil pengujian visi troli belanja

Hasil Proses Akuisisi Citra	Hasil Proses Visi Komputer
	
	



Berdasarkan hasil pengujian visi troli belanja pada tabel 4.1, dapat disimpulkan bahwa proses visi komputer dapat berjalan dengan baik di beberapa lingkungan yang berbeda. Tingkat keberhasilan deteksi warna cukup tinggi, dibuktikan dengan keutuhan dari warna aksesoris peneliti yang terdeteksi, meskipun terjadi perbedaan tingkat pencahayaan ruang dan perbedaan jarak. Tingkat keberhasilan dari ekstraksi kontur ini juga cukup tinggi, dibuktikan dengan kemampuan dalam membedakan antara aksesoris peneliti dan *noise* di sekitar peneliti.

4.2. Pengujian Kecepatan Troli Belanja

Kecepatan rata-rata troli belanja dapat diketahui dengan mengukur lama waktu yang dibutuhkan oleh troli belanja untuk menempuh suatu jarak tertentu. Pada pengujian ini, peneliti berdiri di depan troli belanja pada jarak yang sudah ditentukan. Kemudian troli belanja ini dijalankan dari keadaan diam untuk mendekati peneliti. Perhitungan waktu dimulai

dari ketika troli belanja mulai bergerak hingga ketika troli belanja telah mendekati peneliti. Sama seperti pengujian subbab sebelumnya, peneliti masih menggunakan aksesoris berupa jaket berwarna biru yang sama. Hasil pengujian kecepatan troli belanja ditunjukkan pada tabel 4.2.

Tabel 4.2 Hasil pengujian kecepatan troli belanja

Pengujian ke - n	Jarak (cm)	Waktu (detik)	Kecepatan (cm/detik)
1	100	5.06	19.76
2	100	4.62	21.65
3	100	4.31	23.20
4	100	4.89	20.45
5	100	4.35	22.99
6	200	8.43	23.72
7	200	7.93	25.22
8	200	8.32	24.04
9	200	8.18	24.45
10	200	7.86	25.45
11	300	11.37	26.39
12	300	10.84	27.68
13	300	11.28	26.60
14	300	10.59	28.33
15	300	10.43	28.76

Berdasarkan hasil pengujian kecepatan troli belanja pada tabel 4.2, didapatkan tiga nilai kecepatan rata-rata yang berbeda, bergantung pada jarak antara peneliti dan troli belanja. Pada jarak 100 cm, dapat dihitung kecepatan rata-rata troli belanja yaitu 21.61 cm/detik, pada jarak 200 cm dapat dihitung kecepatan rata-rata troli belanja yaitu 24.58 cm/detik, dan pada jarak 300 cm dapat dihitung kecepatan rata-rata troli belanja yaitu 27.55 cm/detik. Perbedaan kecepatan bisa terjadi karena ada percepatan yang ditimbulkan oleh akumulasi nilai keluaran kontroler PID. Keluaran ini sangat dipengaruhi oleh nilai *error* yang terjadi. Semakin besar jarak antara peneliti dan troli belanja, maka semakin besar *error* dan semakin besar pula kecepatan troli belanja.

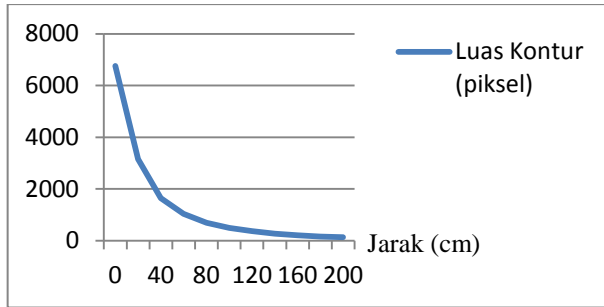
4.3. Pengujian Jarak Troli Belanja

Pengujian jarak antara peneliti dan troli belanja dilakukan dengan cara peneliti berdiri di depan troli belanja dimana orientasinya tetap 0° . Jarak antara peneliti dan troli belanja divariasikan, kemudian dilakukan akuisisi data gambar yang ditampilkan pada layar komputer. Data yang ditampilkan berupa luas kontur yang terdeteksi dan nilai x_l dan x_r yang didapatkan. Pengujian ini bertujuan untuk mengetahui hubungan antara jarak peneliti dengan troli belanja, luas kontur yang terdeteksi, dan juga selisih antara nilai x_r dan x_l . Dari pengujian ini, dapat membantu dalam menentukan nilai luas kontur referensi untuk proses seleksi luas kontur serta nilai $x_{l_{ref}}$ dan $x_{r_{ref}}$ untuk proses kontrol gerak troli belanja. Hasil pengujian jarak troli belanja ditunjukkan pada tabel 4.3.

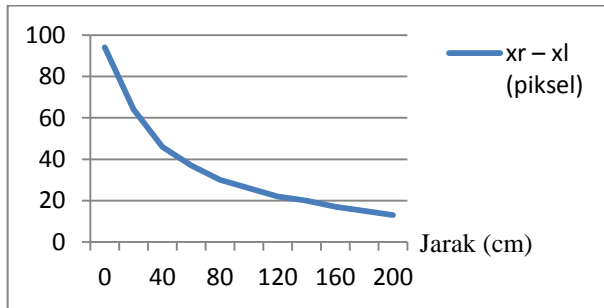
Tabel 4.3 Hasil pengujian jarak troli belanja

Jarak (cm)	Luas Kontur (piksel)	$x_r - x_l$ (piksel)
0	6763	94
20	3157	64
40	1645	46
60	1039	37
80	698	30
100	494	26
120	367	22
140	275	20
160	208	17
180	162	15
200	137	13

Untuk mempermudah pemahaman, tabel di atas dapat digambarkan dalam bentuk grafik hubungan antara jarak dan nilai luas kontur begitu juga dengan antara jarak dan selisih antara nilai x_r dan x_l . Gambar 4.1 (a) menunjukkan grafik hasil pengujian jarak terhadap luas kontur serta (b) menunjukkan grafik hasil pengujian jarak terhadap nilai selisih antar x_r dan x_l . Dengan visualisasi data dalam bentuk grafik dapat membantu peneliti untuk mengetahui pengaruh dari jarak antara peneliti dan troli belanja dengan luas kontur terdeteksi serta nilai x_l dan x_r .



(a)



(b)

Gambar 4.1 Hasil pengujian jarak troli belanja terhadap (a) luas kontur terdeteksi dan (b) selisih nilai x_r dan x_l

Berdasarkan hasil pengujian jarak terhadap luas kontur dan selisih antara nilai x_r dan x_l pada gambar 4.1, dapat disimpulkan bahwa jarak antara peneliti dan troli belanja dapat mempengaruhi luas kontur yang terdeteksi dan selisih antara nilai x_r dan x_l . Hubungan dari jarak peneliti baik dengan luas kontur yang terdeteksi maupun dengan selisih nilai dari x_r dan x_l adalah berbanding terbalik. Dengan demikian, dapat diketahui bahwa nilai x_l dan x_r ini dapat mengindikasikan jarak manusia terhadap troli belanja yang digunakan.

4.4. Pengujian Orientasi Troli Belanja

Pengujian orientasi peneliti terhadap troli belanja dilakukan dengan cara peneliti berdiri di depan troli belanja dimana jaraknya tetap 100 cm.

Orientasi peneliti terhadap troli belanja divariasikan, kemudian dilakukan akuisisi data gambar yang ditampilkan pada layar komputer. Data yang ditampilkan berupa *centroid* kontur serta nilai x_l dan x_r yang didapatkan. Pengujian ini bertujuan untuk mengetahui hubungan antara orientasi dari peneliti terhadap troli belanja, *centroid* kontur yang terdeteksi, dan juga nilai x_l dan x_r . Dari pengujian tersebut dapat membantu peneliti dalam menentukan *centroid* kontur referensi yang digunakan pada proses seleksi *centroid* kontur serta nilai $x_{l_{ref}}$ dan $x_{r_{ref}}$ yang digunakan pada proses kontrol gerak troli belanja. Hasil dari pengujian orientasi troli belanja ini ditunjukkan pada tabel 4.4.

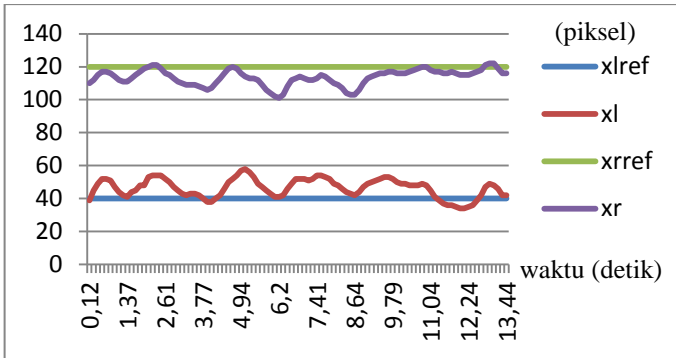
Tabel 4.4 Hasil pengujian orientasi troli belanja

Orientasi (°)	<i>Centroid</i> Kontur	x_l	x_r
-30	14	2	28
-15	49	36	63
0	80	68	95
15	119	106	134
30	150	139	160

Berdasarkan hasil pengujian orientasi terhadap *centroid* kontur dan nilai x_l dan x_r pada tabel 4.4, dapat disimpulkan bahwa orientasi peneliti terhadap troli belanja dapat mempengaruhi *centroid* kontur dan nilai dari x_l dan x_r . Hubungan dari orientasi peneliti baik dengan *centroid* kontur maupun dengan nilai x_l dan x_r adalah berbanding lurus. Dengan begitu, dapat diketahui bahwa nilai x_l dan x_r dapat mengindikasikan orientasi peneliti terhadap troli belanja.

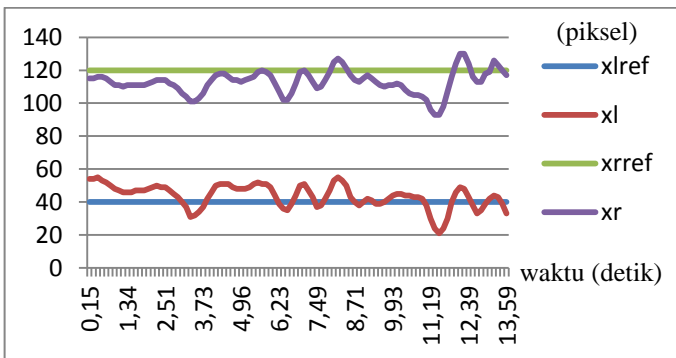
4.5. Pengujian Kontrol Gerak Troli Belanja

Pengujian kontrol gerak troli belanja dapat dilakukan dengan cara memerintahkan troli belanja bergerak mengikuti peneliti yang bergerak dengan 3 macam lintasan, yaitu lintasan lurus, lintasan berbelok ke kiri, lintasan berbelok ke kanan. Kemudian dilakukan akuisisi data variabel proses (x_l dan x_r) pada setiap waktunya. Pengujian ini bertujuan untuk mengetahui kinerja dari kontrol gerak berdasarkan nilai-nilai konstanta parameter *tuning* (k_p , k_i , dan k_d). Dari pengujian ini, dapat membantu peneliti dalam mengatur nilai-nilai parameter untuk memperoleh respon troli belanja yang dikehendaki.



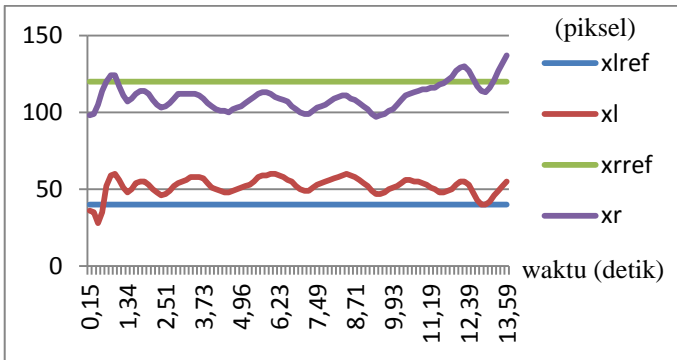
Gambar 4.2 Hasil pengujian kontrol gerak troli belanja pada lintasan lurus

Hasil pengujian dari kontrol gerak lintasan lurus pada gambar 4.2 menunjukkan bahwa ketika peneliti pergi menjauhi troli belanja pada lintasan lurus, maka nilai x_l membesar dan menjauhi x_{lref} dan nilai x_r mengecil dan menjauhi x_{rref} , sehingga selisih antara x_r dan x_l menjadi mengecil. Kemudian kontrol gerak ini berusaha mengontrol gerak troli belanja sehingga nilai x_l mengecil menuju x_{lref} dan nilai x_r membesar menuju x_{rref} , sehingga selisih antara nilai x_l dan x_r menjadi membesar. Dengan demikian, dari ini dapat disimpulkan bahwa kontrol gerak yang dirancang dapat bekerja dengan baik pada lintasan lurus.



Gambar 4.3 Hasil pengujian kontrol gerak troli belanja pada lintasan berbelok ke kiri

Hasil pengujian dari kontrol gerak lintasan berbelok ke kiri pada gambar 4.3 menunjukkan bahwa ketika peneliti ini pergi menjauhi troli belanja dengan lintasan berbelok ke kiri, maka nilai x_l mengecil dengan menjauhi $x_{l_{ref}}$ serta nilai x_r mengecil dengan menjauhi $x_{r_{ref}}$, sehingga nilai rata-rata dari x_l dan x_r juga mengecil. Kemudian kontrol gerak ini berusaha mengontrol gerak troli belanja sehingga nilai x_l membesar dan menuju $x_{l_{ref}}$ serta nilai x_r membesar dan menuju $x_{r_{ref}}$, sehingga nilai rata-rata dari x_l dan x_r juga membesar menuju ke tengah-tengah kedua referensi. Dengan demikian, ini dapat disimpulkan bahwa kontrol gerak yang dirancang dapat bekerja dengan baik pada lintasan berbelok ke kiri.



Gambar 4.4 Hasil pengujian kontrol gerak troli belanja pada lintasan berbelok ke kanan

Hasil pengujian dari kontrol gerak lintasan berbelok ke kanan pada gambar 4.4 menunjukkan bahwa ketika peneliti ini pergi menjauhi troli belanja dengan lintasan berbelok ke kanan, maka nilai x_l membesar dan menjauhi $x_{l_{ref}}$ serta nilai x_r membesar dengan menjauhi $x_{r_{ref}}$, sehingga nilai rata-rata dari x_l dan x_r juga membesar. Kemudian kontrol gerak ini berusaha mengontrol gerak troli belanja sehingga nilai x_l mengecil dan menuju $x_{l_{ref}}$ serta nilai x_r mengecil dan menuju $x_{r_{ref}}$, sehingga nilai rata-rata dari x_l dan x_r juga mengecil menuju ke tengah-tengah kedua referensi. Dengan demikian, ini dapat disimpulkan bahwa kontrol gerak yang dirancang dapat bekerja dengan baik pada lintasan berbelok ke kiri.

BAB V

KESIMPULAN DAN SARAN

1.1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dalam tugas akhir ini, peneliti memperoleh kesimpulan akhir sebagai berikut :

- Metode deteksi warna yang diimplementasikan pada troli belanja telah dapat bekerja dengan cukup baik dalam mendeteksi aksesoris berwarna yang dipakai oleh manusia.
- Metode ekstraksi kontur yang diimplementasikan pada troli belanja telah bekerja dengan cukup baik dalam menentukan posisi manusia relatif terhadap troli belanja.
- Algoritma kontrol PID yang diimplementasikan pada troli belanja telah bekerja dengan cukup baik dalam mengontrol gerak dari troli belanja untuk mengikuti manusia.

1.2. Saran

Berdasarkan hasil penelitian yang dilakukan dalam tugas akhir ini, peneliti memberikan beberapa saran untuk pengembangan penelitian ini lebih lanjut sebagai berikut :

- Menambahkan fitur lain selain fitur warna seperti fitur bentuk atau tekstur untuk meningkatkan akurasi pendeteksian manusia.
- Menggunakan kamera yang dapat mengambil informasi kedalaman citra sehingga dapat mengetahui jarak dengan lebih akurat.
- Menggunakan komponen-komponen seperti kamera, mikrokontrol atau mikroprosesor, *motor driver* yang memiliki kualitas lebih baik sehingga kinerja troli belanja dapat dioptimalkan.

----- *Halaman ini sengaja dikosongkan* -----

DAFTAR PUSTAKA

- [1] Gunarathne, Umesh. (2014). Why Customers Going Supermarkets? (Comparative Study of Motivation Factors Affecting to Visit Supermarkets).
- [2] Wagner, Udo & Ebster, Claus & Eske, Ulrike & Weitzl, Wolfgang. (2014). The Influence of Shopping Carts on Customer Behavior in Grocery Store. *Marketing : ZFP - Journal of Research and Management*.
- [3] Wikipedia contributors. Shopping cart. 23 November 2019. <URL: https://en.wikipedia.org/wiki/Shopping_cart>. [Diakses tanggal 26 November 2019].
- [4] Krishnamohan, Thebeyanthan & Mahendran, Achsuthan & Selvakumar, Ashok & Paramananthasivam, Vaikunthan. (2016). Human Following Trolley - Auto Walker.
- [5] Yewatkar, Ankush & Inamdar, Faiz & Singh, Raj & Ayushya, & Bandal, Amol. (2016). Smart Cart with Automatic Billing, Product Information, Product Recommendation Using RFID & Zigbee with Anti-Theft. *Procedia Computer Science*.
- [6] Sharma, Avadhesh. (2019). Smart Shopping Trolley. *International Journal for Research in Applied Science and Engineering Technology*.
- [7] Irawan, Denny & Misbah & Baihaqi, Jefry. 2019. Design of Human Followers Trolley Robot Based on Object Tracking (Color Detection) Method. *International Journal of Engineering Research and Advanced Technology*.
- [8] Hanwate, Aboli & Thakare, Poonam. 2012. Smart Trolley Using RFID. *International Journal of Research in Science & Engineering*.
- [9] Sales, Jorge & Marti, Jose & Marín Prades, Raul & Cervera, Enric & Sanz, Pedro. (2016). CompaRob : The Shopping Cart Assistance Robot. *International Journal of Distributed Sensor Networks*.
- [10] Ng, Yen & Lim, Cheng Siong & A. Danapalasingam, Kumeresan & Tan, Michael & Tan, Chee Wei. (2015). Automatic Human Guided Shopping Trolley with Smart Shopping System. *Jurnal Teknologi*.

----- *Halaman ini sengaja dikosongkan* -----

LAMPIRAN

Program untuk Raspberry Pi

```
#!/usr/bin/env python3
import cv2 as cv
import RPi.GPIO as io
from os import system
from serial import Serial
from simple_pid import PID
from numpy import array,linalg
from time import sleep,perf_counter
#-----
def stop() :
    print('STOP')
    beep(0.1,0.1,1)
    setMotorLeft(0)
    setMotorRight(0)
    while 1 :
        if not io.input(3) :
            beep(0.1,0.1,1)
            t = perf_counter()
            while not io.input(3) :
                if perf_counter()-t >= 3 :
                    WCAM.release()
                    cv.destroyAllWindows()
                    beep(0.1,0.1,3)
                    system('sudo shutdown -h now')
        if BTTY.in_waiting :
            data = BTTY.read(20).decode('UTF-8')
            data = data.replace('*',')
            data = data.split(',')
            print(data)
            if data[0] == 'Follow' :
                if data[1] == 'Blue' :
                    lth = array([80,100,0])
                    uth = array([130,255,255])
                    sp = 0.795*int(data[2])
                    LPID.setpoint = int(sp-79.5)
```

```

RPID.setpoint = int(sp+79.5)
follow(lth,uth)
#-----
def follow(lth,uth) :
    print('FOLLOW')
    LPID.reset()
    RPID.reset()
    t = perf_counter()
    while io.input(18) :
        lth = array([cv.getTrackbarPos('Hmin ','TRACKBAR'),
                    cv.getTrackbarPos('Smin ','TRACKBAR'),
                    cv.getTrackbarPos('Vmin ','TRACKBAR')])
        uth = array([cv.getTrackbarPos('Hmax ','TRACKBAR'),
                    cv.getTrackbarPos('Smax ','TRACKBAR'),
                    cv.getTrackbarPos('Vmax ','TRACKBAR')])
        frame = WCAM.read()[1]
        mask = cv.GaussianBlur(frame,(5,5),0,0)
        mask = cv.cvtColor(mask,cv.COLOR_BGR2HSV)
        mask = cv.inRange(mask,lth,uth)
        mask =
cv.morphologyEx(mask,cv.MORPH_OPEN,cv.getStructuringElement(c
v.MORPH_RECT,(5,5)))
        mask =
cv.morphologyEx(mask,cv.MORPH_CLOSE,cv.getStructuringElement(
cv.MORPH_RECT,(5,5)))
        cnts =
cv.findContours(mask,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_SI
MPLE)[0]
        mask = cv.bitwise_and(frame,frame,mask=mask)
        teud,tcnt,tctr = [],[],[]
        for cnt in cnts :
            if cv.contourArea(cnt) >= 500 :
                try :
                    m = cv.moments(cnt)
                    nctr = array([int(m['m10']/m['m00']),
                                int(m['m01']/m['m00'])])
                    eud = linalg.norm(nctr-octr)
                    teud.append(eud)
                    tcnt.append(cnt)

```

```

        tctr.append(nctr)
    except : pass
if len(teud) :
    tgt = teud.index(min(teud))
    xl,_,xr,_ = cv.boundingRect(tcnt[tgt])
    xr += xl
    octr = tctr[tgt]
    if -xl >= LPID.setpoint and xr >= RPID.setpoint :
        setMotorLeft(0)
        setMotorRight(0)
        LPID.reset()
        RPID.reset()
    else :
        setMotorLeft(LPID(-xl))
        setMotorRight(RPID(xr))
    cv.line(mask,(xl,0),(xl,120),(0,255,0),1)
    cv.line(mask,(xr,0),(xr,120),(0,255,0),1)
    cv.circle(mask,tuple(octr),1,(0,255,0),1)
    cv.drawContours(mask,[tcnt[tgt]],0,(0,255,0),1)
    with open('/home/pi/Desktop/PID.txt','a') as file :
        file.write(str(perf_counter()-t)+' '+
            str(LPID.setpoint)+' '+str(xl)+' '+
            str(RPID.setpoint)+' '+str(xr)+'\n')
else :
    beep(0.1,0.1,1)
    setMotorLeft(0)
    setMotorRight(0)
    LPID.reset()
    RPID.reset()
    octr = array([79,59])
cv.imshow('MASK',mask)
if cv.waitKey(1) == 27 : break
if not io.input(3) : break
if BTTY.in_waiting :
    data = BTTY.read(20).decode('UTF-8')
    data = data.replace('*','')
    data = data.split(',')
    print(data)
    if data[0] == 'Stop' : break

```

```

stop()
#-----
def beep(ton,toff,n) :
    for i in range(n) :
        io.output(BZZR,True)
        sleep(ton)
        io.output(BZZR,False)
        sleep(toff)
#-----
def setMotorMode(motor,mode) :
    if motor == 'LEFT' :
        if mode == 'FORWARD' :
            io.output(LLEN,True)
            io.output(LREN,False)
        elif mode == 'REVERSE' :
            io.output(LLEN,False)
            io.output(LREN,True)
        else :
            io.output(LLEN,False)
            io.output(LREN,False)
    elif motor == 'RIGHT' :
        if mode == 'FORWARD' :
            io.output(RLEN,True)
            io.output(RREN,False)
        elif mode == 'REVERSE' :
            io.output(RLEN,False)
            io.output(RREN,True)
        else :
            io.output(RLEN,False)
            io.output(RREN,False)
    else :
        io.output(LLEN,False)
        io.output(LREN,False)
        io.output(RLEN,False)
        io.output(RREN,False)
#-----
def setMotorLeft(pwm) :
    if pwm < 0 :
        pwm = -pwm

```

```

        if pwm > 100 : pwm = 100
        LLPWMPin.ChangeDutyCycle(pwm)
        LRPWMPin.ChangeDutyCycle(0)
    elif pwm > 0 :
        pwm = pwm
        if pwm > 100 : pwm = 100
        LLPWMPin.ChangeDutyCycle(0)
        LRPWMPin.ChangeDutyCycle(pwm)
    else :
        LLPWMPin.ChangeDutyCycle(0)
        LRPWMPin.ChangeDutyCycle(0)
#-----
def setMotorRight(pwm) :
    if pwm < 0 :
        pwm = -pwm
        if pwm > 100 : pwm = 100
        RLPWMPin.ChangeDutyCycle(pwm)
        RRPWMPin.ChangeDutyCycle(0)
    elif pwm > 0 :
        pwm = pwm
        if pwm > 100 : pwm = 100
        RLPWMPin.ChangeDutyCycle(0)
        RRPWMPin.ChangeDutyCycle(pwm)
    else :
        RLPWMPin.ChangeDutyCycle(0)
        RRPWMPin.ChangeDutyCycle(0)
#-----
def resetMotor() :
    io.output(LLEN,False)
    io.output(LREN,False)
    io.output(RLEN,False)
    io.output(RREN,False)
#-----
io.setmode(io.BCM)
io.setwarnings(False)
BTTY = Serial('/dev/ttyAMA0',timeout=0.1)
BTTN = 3
io.setup(BTTN,io.IN,io.PUD_UP)
BTCN = 18

```

```

io.setup(BTCN,io.IN,io.PUD_DOWN)
BZZR = 23
io.setup(BZZR,io.OUT)
io.output(BZZR,False)
#-----
LPID = PID(0.5,0.3,0.1,0,0.2,(0,100),True,False)
LLEN = 6
io.setup(LLEN,io.OUT)
io.output(LLEN,True)
LREN = 13
io.setup(LREN,io.OUT)
io.output(LREN,True)
LLPWM = 19
io.setup(LLPWM,io.OUT)
LLPWMpin = io.PWM(LLPWM,1000)
LLPWMpin.start(0)
LLPWMpin.ChangeDutyCycle(0)
LRPWM = 26
io.setup(LRPWM,io.OUT)
LRPWMpin = io.PWM(LRPWM,1000)
LRPWMpin.start(0)
LRPWMpin.ChangeDutyCycle(0)
#-----
RPID = PID(0.5,0.3,0.1,0,0.2,(0,100),True,False)
RLEN = 12
io.setup(RLEN,io.OUT)
io.output(RLEN,True)
RREN = 16
io.setup(RREN,io.OUT)
io.output(RREN,True)
RLPWM = 20
io.setup(RLPWM,io.OUT)
RLPWMpin = io.PWM(RLPWM,1000)
RLPWMpin.start(0)
RLPWMpin.ChangeDutyCycle(0)
RRPWM = 21
io.setup(RRPWM,io.OUT)
RRPWMpin = io.PWM(RRPWM,1000)
RRPWMpin.start(0)

```

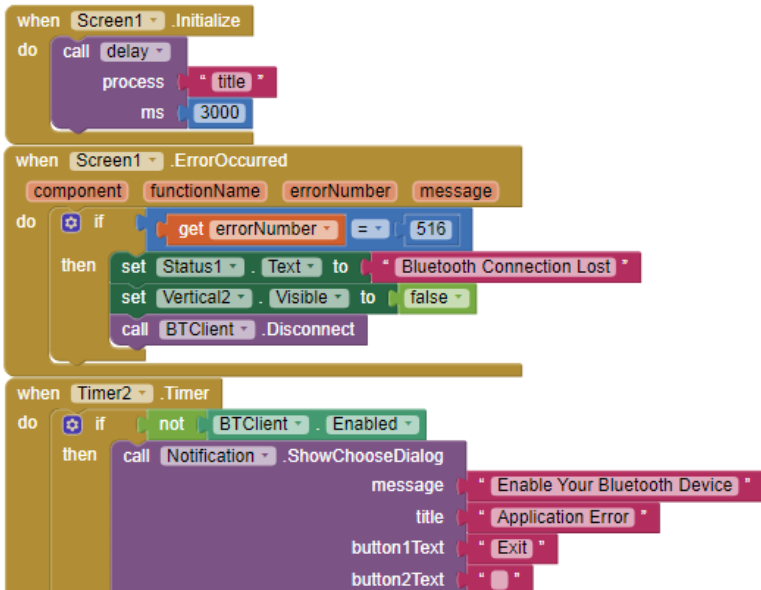


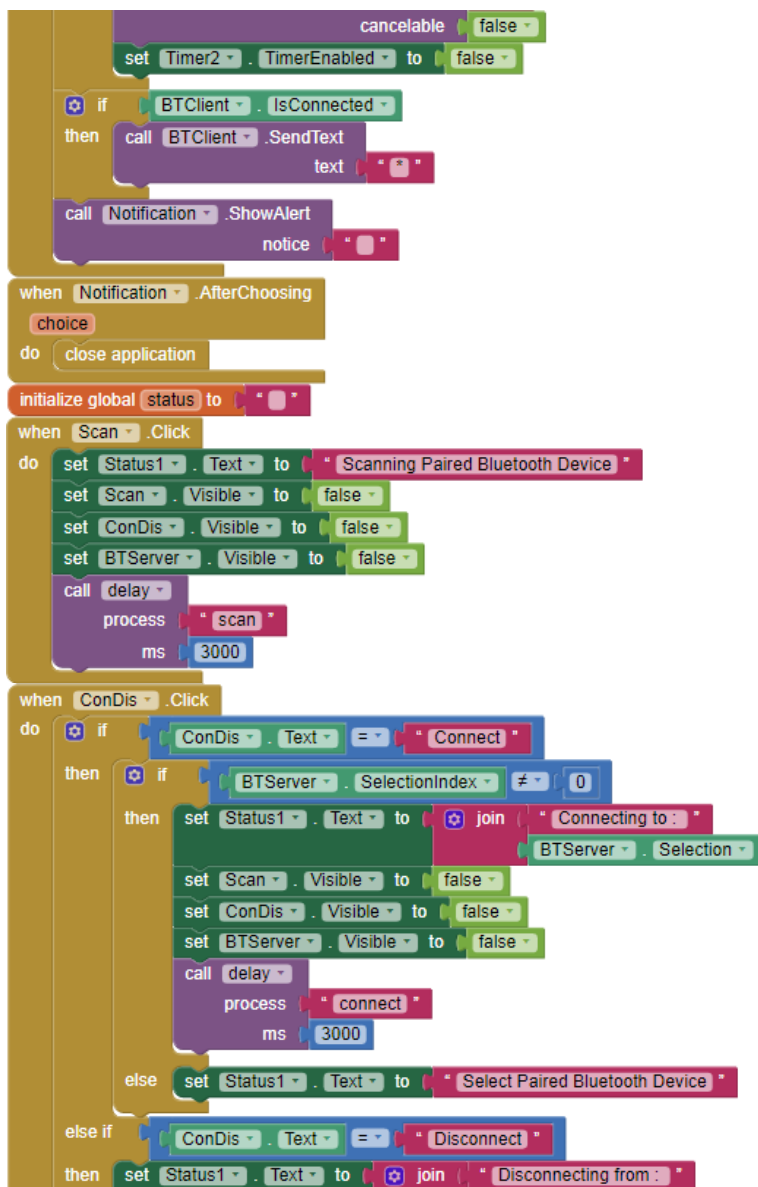
```

RRPWMpin.ChangeDutyCycle(0)
#-----
WCAM = cv.VideoCapture(cv.CAP_ANY)
WCAM.set(cv.CAP_PROP_FRAME_WIDTH,160)
WCAM.set(cv.CAP_PROP_FRAME_HEIGHT,120)
def none(n) : pass
cv.namedWindow('MASK',cv.WINDOW_NORMAL)
cv.namedWindow('TRACKBAR',cv.WINDOW_NORMAL)
cv.createTrackbar('Hmin ', 'TRACKBAR',0,179,none)
cv.createTrackbar('Hmax ', 'TRACKBAR',0,179,none)
cv.createTrackbar('Smin ', 'TRACKBAR',0,255,none)
cv.createTrackbar('Smax ', 'TRACKBAR',0,255,none)
cv.createTrackbar('Vmin ', 'TRACKBAR',0,255,none)
cv.createTrackbar('Vmax ', 'TRACKBAR',0,255,none)
beep(0.1,0.1,2)
stop()

```

Program untuk Aplikasi Smartphone





```

BTServer . Selection
set Vertical2 . Visible to false
set ConDis . Visible to false
call delay
  process "disconnect"
  ms 3000

when FollStop . Click
do
  if FollStop . Text = "Follow"
  then
    set Status2 . Text to "Following"
    call BTClient . SendText
      text join "Follow,"
      Color . Selection
      " "
      100 - get global distance
    set FollStop . Text to "Stop"
  else if FollStop . Text = "Stop"
  then
    set Status2 . Text to "Stopped"
    call BTClient . SendText
      text "Stop"
    set FollStop . Text to "Follow"

to delay process ms
do
  set global status to get process
  set Timer1 . TimerInterval to get ms
  set Timer1 . TimerEnabled to true

when Timer1 . Timer
do
  if get global status = "title"
  then
    set Title . Visible to false
    set Screen1 . AlignVertical to 3
    set Horizontal1 . Visible to true
    set Vertical1 . Visible to true
    set Space8 . Visible to true
    set Timer1 . TimerEnabled to false
  else if get global status = "scan"

```

```

then
  set BTServer . Elements to BTClient . AddressesAndNames
  set BTServer . SelectionIndex to 0
  set Status1 . Text to " Done Scanning "
  set BTServer . Visible to true
  set Scan . Visible to true
  set ConDis . Visible to true
  set Timer1 . TimerEnabled to false
else if
  get global status = " connect "
then
  if
    call BTClient . Connect
      address BTServer . Selection
  then
    set Status1 . Text to join " Connected to : "
      BTServer . Selection
    set ConDis . Text to " Disconnect "
    set ConDis . WidthPercent to 90
    set Space1 . Visible to false
    set Vertical2 . Visible to true
    set ConDis . Visible to true
  else
    set Status1 . Text to join " Unconnected to : "
      BTServer . Selection
    set BTServer . SelectionIndex to 0
    set BTServer . Visible to true
    set Scan . Visible to true
    set ConDis . Visible to true
  set Timer1 . TimerEnabled to false
else if
  get global status = " disconnect "
then
  call BTClient . Disconnect
  set Status1 . Text to join " Disconnected from : "
    BTServer . Selection
  set BTServer . SelectionIndex to 0
  set Status2 . Text to " Stopped "
  set FollowStop . Text to " Follow "
  set ConDis . Text to " Connect "
  set ConDis . WidthPercent to 43
  set BTServer . Visible to true
  set Scan . Visible to true

```

```

set ConDis . Visible to true
set Timer1 . TimerEnabled to false
else if get global status = "disconnect"
then
call BTClient . Disconnect
set Status1 . Text to join " Disconnected from : "
BTServer . Selection
set BTServer . SelectionIndex to 0
set Status2 . Text to " Stopped "
set FollStop . Text to " Follow "
set ConDis . Text to " Connect "
set ConDis . WidthPercent to 43
set BTServer . Visible to true
set Scan . Visible to true
set Space1 . Visible to true
set ConDis . Visible to true
set Timer1 . TimerEnabled to false

initialize global distance to ""
when Distance . PositionChanged
thumbPosition
do set global distance to get thumbPosition

```

----- Halaman ini sengaja dikosongkan -----

BIODATA PENULIS



Penulis bernama Alvin Abraham. Penulis dilahirkan di Kediri pada tanggal 25 November 1997. Penulis adalah anak kedua dari dua bersaudara. Penulis menempuh pendidikan dasar di SDK Santa Maria Kediri, dilanjutkan dengan pendidikan menengah pertama di SMPN 1 Kediri dan pendidikan menengah atas di SMAN 2 Kediri. Pada tahun 2016, penulis diterima sebagai mahasiswa di Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis memilih bidang studi Elektronika. Selama menjalani kegiatan perkuliahan, penulis aktif dalam berpartisipasi sebagai Asisten Dosen Matematika 1 dan 2 serta Asisten Laboratorium Elektronika Mikro dan Sistem Tertanam.

E-Mail : alvabra25@gmail.com

----- *Halaman ini sengaja dikosongkan* -----