# Skripsi

**SEGMENTASI DAN IDENTIFIKASI COIN DENGAN MENGGUNAKAN CAMERA PADA CONVEYOR**



Diajukan Oleh:

**Juheri Valentino Silitonga**

**147030356**

**USULAN PENELITIAN**

**Diajukan untuk menyusun skripsi**

**Jurusan Teknik Elektro**

**PROGRAMSTUDITEKNIK ELEKTRO**

**FAKULTASTEKNOLOGIINDUSTRI**

**UNIVERSITASBALIKPAPAN**

**BALIKPAPAN**

**Tahun2019**

# PERSETUJUAN

Proposal Skripsi Mahasiswa

**Juheri Valentino Silitonga**

147030356

Dengan Judul

**SEGMENTASI DAN IDENTIFIKASI COIN DENGAN MENGGUNAKAN CAMERA PADA CONVEYOR**

Telah disetujui untuk diseminarkan pada tanggal,

**Persetujuan Pembimbing**

Pembimbing I Pembimbing II

**Mayda Waruni K,S.T.,M.T. Dr. Ir. Charles Pangaribuan. DEA. CES**

Mengetahui,

Ketua Program Studi: S1 Teknik Elektro

**A. Asni B, S.T., M.Eng.**

NIK. 006 003 003

# PERNYATAANKEASLIANTULISAN

Sayayangbertandatangandi bawahini:

Nama :**Juheri Valentino Silitonga**

NIM : 147030356

ProgramStudi : Teknik Elektro

Fakultas :Teknologi Industri

Menyatakan dengansebenarnya bahwa skripsi/tugas akhir yangsaya tulis inibenar-benar merupakanhasilkaryasayasendiri;bukanmerupakanpengambilalihan tulisanataupikiran oranglainyangsayaakusebagaihasiltulisanataupikiransayasendiri.

Apabiladi kemudianhariterbuktiataudapatdibuktikanskripsi/tugasakhirinihasiljiplakan, makasayabersediamenerimasanksiatasperbuatantersebut.

Balikpapan,25 Juni 2019

Yangmembuatpernyataan,

**JUHERI VALENTINO S**

# PRAKATA

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan barokahnya sehingga penulis dapat menyelesaikan skripsi dengan judul “**SEGMENTASI DAN IDENTIFIKASI COIN DENGAN MENGGUNAKAN CAMERA PADA CONVEYOR**”. Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T.) pada Program Studi Teknik Elektro/Teknik Mesin Fakultas Teknologi Industri Universitas Balikpapan.

Dalam melakukan penelitian dan penyusunan laporan skripsi ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada:

1. Mayda Waruni,ST,MT selaku dosen pembimbing I, danDr. Ir. Charles Pangaribuan selaku dosen pembimbing II, yang telah dengan penuh kesabaran dan ketulusan memberikan ilmu dan bimbingan terbaik kepada penulis.

2. A. Asni B, S.T, M.Eng selaku Ketua Program Studi Teknik Elektro/Teknik Mesin Fakultas Teknologi Industri Universitas Balikpapan yang memberikan izin kepada penulis untuk belajar.

3. Para Dosen Program Studi Teknik Elektro/Teknik Mesin Fakultas Teknologi Industri Universitas Balikpapan yang telah memberikan bekal ilmu kepada penulis.

4. Para Karyawan/wati Program Studi Teknik Elektro/Teknik Mesin Fakultas Teknologi Industri Universitas Balikpapan yang telah membantu penulis dalam proses belajar serta dan Teman-teman Teknik Elektro semuanya.

Penulis menyadari sepenuhnya bahwa laporan skripsi ini masih jauh dari sempurna, untuk itu semua jenis saran, kritik dan masukan yang bersifat membangun sangat penulis harapkan. Akhir kata, semoga tulisan ini dapat memberikan manfaat dan memberikan wawasan tambahan bagi para pembaca dan khususnya bagi penulis sendiri.

Balikpapan, 25 Juni 2019

Juheri Velentino Silitonga

# ABSTRAK

*A conveyor system is a common piece of mechanical handling equipment that moves materials from one location to another. Conveyors are especially useful in applications involving the transportation of heavy or bulky materials that allow quick and efficient transportation for a wide variety of materialscontinuously. Purposeofthis thesiswas to design a system which is able to automatically detect coin using image processing and analysis methods. The state information is used as an input that is selected into the memory. The output of the solving algorithm is a list of necessary movements to segmentation for cion. The software is used Visual C++ 2017 Community thats supports for making image processing with camera and the hardware is used servo motor to be controlled by minimum system Arduino. From the result of motor movement system, detection dan segmentation coin obtained with the success of 95%.*

***Keywords:****Coin, image processing, camera, minimum system Arduino.*

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc12466593)

[PERSETUJUAN ii](#_Toc12466594)

[PERNYATAANKEASLIANTULISAN iii](#_Toc12466595)

[PRAKATA iv](#_Toc12466596)

ABSTRAK v

[DAFTAR ISI v](#_Toc12466597)i

[DAFTAR GAMBAR vii](#_Toc12466598)i

[DAFTAR TABEL](#_Toc12466598) x

[ARTI LAMBANG DAN SINGKATAN i](#_Toc12466599)x

[BAB I PENDAHULUAN 1](#_Toc12466600)

[1.1 Latar Belakang Masalah 1](#_Toc12466601)

[1.2 Rumusan Masalah 2](#_Toc12466602)

[1.3 Tujuan Penelitian 2](#_Toc12466603)

[1.4 Batasan Masalah 2](#_Toc12466604)

[1.5 Manfaat Penelitian 2](#_Toc12466605)

[1.6 Sistematika Penulisan 3](#_Toc12466606)

[BAB IITINJAUAN PUSTAKA 5](#_Toc12466607)

[2.1 Penelitian yang Relevan 5](#_Toc12466608)

[2.2 Citra Digital 5](#_Toc12466609)

[2.3 Pengolahan Citra Digital 6](#_Toc12466610)

[2.3.1 Greyscaling 7](#_Toc12466611)

[2.3.2 Smoothing 8](#_Toc12466612)

[2.3.3 Laplace 10](#_Toc12466613)

[2.4 Pengenalan Pola Citra Digital 11](#_Toc12466614)

[2.4.1 Blob Detection 12](#_Toc12466615)

[2.4.2 Find Contour 14](#_Toc12466616)

[2.5 Motor Servo 15](#_Toc12466617)

[2.6 Belt Conveyor 16](#_Toc12466618)

[BAB III METODE PENELITIAN 17](#_Toc12466620)

[3.1 Waktu dan Tempat Penelitian 17](#_Toc12466621)

[3.2 Metode Pengumpulan Data 17](#_Toc12466622)

[3.3 Instrumen Penelitian 1](#_Toc12466623)7

[3.4 Jalannya Penelitian 18](#_Toc12466624)

[3.5 Metode Analisis Data 19](#_Toc12466625)

[BAB IVPEMBAHASAN DAN HASIL 20](#_Toc12466607)

[4.1 Kamera Sebagai Sensor Visual 21](#_Toc12466608)

[4.2 Central Processing Unit 22](#_Toc12466609)

[4.3 Percangan Perangkat Keras 22](#_Toc12466610)

[4.3.1 Pengontrolan Motor Servo 23](#_Toc12466611)

[4.3.2 Minimum Sistem Arduino 24](#_Toc12466612)

[4.4Pengolahan dan Pengenalan Pola Citra Digital 27](#_Toc12466617)

[4.5Proses Segmentasi Coin 32](#_Toc12466618)

[4.6Pengujian Gerak Motor Servo 34](#_Toc12466618)

[4.7Pengujian Pergankat Lunak 35](#_Toc12466614)

[4.7.1 Pengujian Deteksi Coin 35](#_Toc12466615)

[4.7.2 Pengujian Segmentasi Coin 37](#_Toc12466616)

[4.7Pembahasan Permasalahan 37](#_Toc12466614)

[BAB VKESIMPULAN DAN SARAN 39](#_Toc12466620)

[3.1 Waktu dan Tempat Penelitian 39](#_Toc12466621)

[3.2 Metode Pengumpulan Data 39](#_Toc12466622)

[DAFTAR PUSTAKA 21](#_Toc12466626)

# DAFTAR GAMBAR

* 1. Citra Lena dan Citra Rubik
  2. Blok Sistem Pengolahan Citra Digital
  3. Citra Lena yang Mengandung Derau dan Hasil dari Penapisan Citra
  4. Gambar Berwarna dan Gambar *Greyscale*
  5. Kernel Operator *Gaussian* dengan = 1,4
  6. Sebelum dan Sesudah
  7. Efek *Laplcae Filter* yang Kedua (-1,8,-1)
  8. Blok Sistem Pengenalan Pola Citra Digital
  9. Citra Karakter ‘A’ yang Digunakan Sebagai Masukan untuk Pengenalan Huruf
  10. Representasi Hough Transform
  11. Hough Transform dalam Koordinat Polar
  12. Gambar *Cone* yang Merepentasikan Nilai Skala Warna HSV
  13. Teknik PWM Untuk Mengatur Motor Servo
  14. *Pin Out* Kabel Motor Servo
  15. Bagian Tepi
  16. Bagian Sudut
  17. Bagian Tengah Setiap Langkah Dipresentasikan dengan Huruf

Huruf dengan ‘i’ adalah Inverse dari Pergerakan Tersebut

Gambar 3.1 Diagram Alir Penelitian

Gambar 4.1 Diagram Blok Sistem

Gambar 4.2 Logitech HD Webcam C525

Gambar 4.3 Posisi kamera sebagai sensor visual terhadap coin.

Gambar 4.4 Laptop HP 14-bw005au

Gambar 4.5 Perangkat keras berupa dua motor servo, minimum sistem Arduino, power supply dan dua motor driver.

Gambar 4.6 Blok diagram sistem pergerakan motor servo

Gambar 4.7 Servo motor pada saat dalam posisi di atas pulsa 0.25 ms.

Gambar 4.8 Servo motor pada saat dalam posisi di bawah pulsa 1 ms.

Gambar 4.9 Arduino Main Board berbasis ATmega328 dengan bootloader[14]

Gambar 4.10 Contoh program Arduino menggunakan Java

Gambar 4.11 Breadboard rangkaian motor servo menggunakan Arduino

Gambar 4.12 Schematic rangkaian microcontroller

Gambar 4.13 Citra yang diambil dengan kamera

Gambar 4.14 Citra threshold invert

Gambar 4.15 Citra FilterBlob

Gambar 4.16 Citra Filter Canny

Gambar 4.17 Design dan implementasi

Gambar 4.18 Tepi lingkaran yang terdeteksi Canny Filter setelah menggunakan find contour

Gambar 4.19 Contoh dua garis yang saling berpotongan dan tidak berpotongan

Gambar 4.20 Dua garis yang saling berpotongan pada sebuah titik

Gambar 4.21 Hasil citra setelah dicari persamaan garis

Gambar 4.22 Hasil citra setelah dicari persamaan garis

Gambar 4.23 Contoh motor servo error

Gambar 4.24 Contoh coin yang jatuh bersamaan

**DAFTAR TABEL**

Tabel 4. 1 Hasil pengujian pergerakan motor servo tutup

Tabel 4. 2 Hasil pengujian pergerakan motor servo buka

Tabel 4. 3 Hasil pengujian pergerakan motor servo buka tidak terdapat coin sama sekali.

Tabel 4. 4 Hasil pengujian deteksi coin pada siang hari

Tabel 4. 5 Hasil pengujian deteksi coin pada malam hari

Tabel 4. 6 Hasil pengujian segmentasi coin

# ARTI LAMBANG DAN SINGKATAN

SMS (Short Message Service)

MMS (Multimedia Message Service).

PWM (Pulse Width Modulation)

DIY (Do It Yourself)

LED (Light Emitting Diode)

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Pada saat ini, perkembangan ilmu pengetahuan dan teknologi terutama dalam bidang elektronika semakin berkembang pesat. Perkembangan ini sedikit banyaknya telah meningkatkan kualitas kehidupan manusia, seiring berjalannya waktu dengan banyaknya tuntutan masyarakat akan produk-produk yang memiliki kualitas baik dan bermutu yang dihasilkan oleh industri. Hal ini telah membuat banyak proses industri beralih dari sebuah sistem manual ke sistem otomatis yang lebih mudah dikendalikan dengan peran manusia yang semakin lama semakin berkurang.

Salah satunya topik yang saat ini sedang dikembangkan dan banyak sekali yang bisa ditemukan aplikasinya di dunia industri adalah *conveyor*dan *image processing*. *Conveyor* merupakan alat bantu yang umum dijumpai pada pada industri-industri pengolahan, alat ini digunakan untuk memindahkan satu produk ketempat lain secara berurutan. Selain itu, penggunaan *image processing* akan sangatmemudahkan manusia untuk mengetahui jenis suatu benda secara jelas. Di era teknologi yang semakin canggih ini, mengganti sensor yang masih manual dengan mengunakan sensor kamera dapat menurunkan biaya produksi suatu industri. Untuk itu diperlukan suatu alat yang pintar yang mampu mendeteksi benda secara cepat dan akurat, sehingga mengurangi kesalahan penyortiran pada sistem *conveyor*.

Prinsip kerja sistem ini, yaitu input berupa *coin* ditangkap oleh kamera dan citranya berupa objek yang berbentuk lingkaran yang akan diletakan di atas *conveyor* berjalan. Proses pengenalanbentuk tersebut akan dihitung jarak diameter *coin-coin* yang melintas menggunakan kamera yang fungsinya menggantikan mata manusia untuk membedakan *coin-coin* tersebut, karena pada setiap *coin* memiliki diameter yang berbeda-beda. Untuk pemogramannya sendiri menggunakan Visual C++ dan *image processing* menggunakan OpenCV, sedangkan untuk kontrolernya menggunakan mikrokontroler Arduino. *Coin* yang berhasil terdeteksi akan disortir sesuai wadahnya, sedangkan untuk *coin* yang cacat produksi tidak akan terdeteksi dan dilewati begitu saja sampai masuk ke wadah cacat produksi.

Oleh karena itu penulis ingin menggunakan teknologi mikrokontroller yang diaplikasikan untuk penyortiran benda pada conveyor dengan judul segmentasi dan identifikasi coin dengan menggunakan camera pada conveyor

## 1.2 Rumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana cara mendeteksi dan *tracking* objek pada *coin* menggunakan kamera pada OpenCV ?
2. Bagaimana cara mengintegrasikan Visual C++ dengan minimum sistem Arduino pada *conveyor* ?
3. Bagaimana mengatur kontroler pada penyortiran *coin* ?

## 1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan untuk membuat sebuah alat yaitu *conveyor*untuk jenisyang mampuberkerja secara otomatis menyortir benda dan diletakan pada suatu wadah berdasarkan bentukmenggunakan camera sebagai alat bantu pengidentifikasi bendasecara *realtime*.

## 1.4 Batasan Masalah

Dalam pengerjaan tugas akhir, permasalahan di atas dibatasi dengan asumsi sebagai berikut :

1. Kamera yang digunakan adalah kamera webcam yang mempunyai kecepatan minimal 30 frame per second video dan 1.3 megapixel.
2. Pengambilan data diperlukan pencahayaan yang cukup untuk dapat mendeteksi coin.
3. *Coin* yang dipakai hanya 3 saja yaitu 100, 500 dan 200 Rupiah.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Sebagai alat bantu memudahkan pekerjaan manusia khususnya pada industri-industri sekala besar, baik dari segi efesiensi waktu, tenaga, kualitas serta hasil yang diperoleh.
2. Pengenalan benda menggunakan sensor kamera akan menggantikan sensor-sensor yang masih manual sehingga mengurangi jumlah sensor-sensor yang dipakai begitu banyak pada sistem *conveyor*.
3. Sebagai bahan untuk penelitian selanjutnya. Contohnya penyortiran buah jeruk berdasarkan warna dan bentuk, penyortiran benda-benda kecil seperti baut dan mur yang berdasarkan bentuk ukuran dan masih banyak lagi aplikasi yang sensor-sensor manual sulit lakukan.

## 1.6 Sistematika Penulisan

Hasil penelitian ini, selanjutnya disusun dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Berisikan latar belakang masalah, perumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan di dalam penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA

Pada bagian ini berisi tentang landasan teori yang digunakan dalam pelaksanaan Tugas akhir yang meliputi Citra Digital, Pengolahan Citra Digital, Segmentasi Citra menggunakan Thresholding, Konfigurasi *conveyor* dan mikrokontroler Arduino. Bagian ini memaparkan tentang beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

BAB III METODE PENELITIAN

Berisi tentang Waktu dan Tempat Penelitian, Objek Penelitian, Teknik Pengumpulan Data, Alat dan Bahan, Prosedur Penelitian, Variabel Penelitian dan Diagram Alir Penelitian, Langkah-langkah ini harus disesuaikan dengan fokus permasalahan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil-hasil yang diperoleh dan cara pencapaiannya. Uraian harus komprehensif namun tetap ringkas dan padu. Pembahasan hasil penelitian meliputi kelebihan dan kekurangan, termasuk pengujian.

BAB V PENUTUP

Bagian ini berisi kesimpulan dan saran, kesimpulan merupakan rangkuman hasil yang dicapai dan merupakan jawaban rumusan masalah, sedangakn saran Bagian ini menguraikan saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi yang dibuat, termasuk saran untuk pengembangan lebih lanjut.

# BAB II

# TINJAUAN PUSTAKA

## 2.1 Citra Digital

Data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio (bunyi, suara, musik), dan video. Keempat macam data atau informasi ini sering disebut multimedia. Era teknologi informasi saat ini tidak dapat dipisahkan dari multimedia. Situs web (website) di Internet dibuat semenarik mungkin dengan menyertakan visualisasi berupa gambar atau video yang dapat diputar. Beberapa waktu lalu istilah SMS *(Short Message Service)* begitu populer bagi pengguna telepon genggam (*handphone* atau HP). Tetapi, saat ini orang tidak hanya dapat mengirim pesan dalam bentuk teks, tetapi juga dapat mengirim pesan berupa gambar maupun video, yang dikenal dengan layanan MMS (*Multimedia Message Service*).

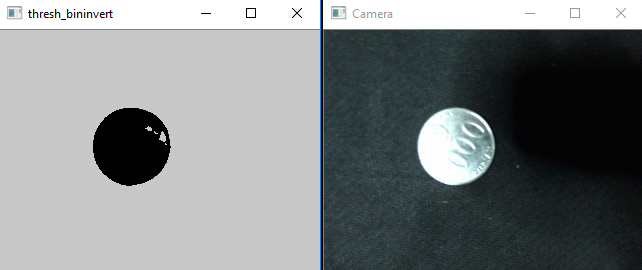
Citra (*image*) sebagai salah satu komponen multimedia memegang peranan sangaat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi “sebuah gambar bermakna lebih dari seribu kata” (*a picture is more than a thousand words*). Maksudanya tentu sebuah gambar dapat memberikan informasi yang lebih banyak daripada informasi tersebut disajikan dalam bentuk kata-kata (tekstual).

Citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Gambar 2.1 adalah citra seorang gadis model yang bernama Lena, dan gambar disebelah kanannya adalah citra sebuah rubik. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

Citra sebagai keluaran dari sistem perekaman data dapat bersifat :

1. Optik berupa foto,
2. Analog berupa sinyal video seperti gambar pada monitor televisi,
3. Digital yang dapat langsung disimpan pada suatu pita magnetik.

Citra yang dimaksudkan di dalam keseluruhan isi buku ini adalah “citra diam” (*still images*). Citra diam adalah citra tunggal yang tidak bergerak. Gambar 2.1 adalah dua buah citra diam. Untuk selanjutnya, citra diam kita sebut citra saja.



* + 1. (b)

**Gambar 2.1** (a) Citra Lena dan (b) Citra *Coin* [1]

Citra bergerak (*moving images*) adalah rangkaian citra diam yang ditampilkan secara (sekuensial) sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri atas ratusan sampai ribuan *frame*.

## 2.2 Pengolahan Citra Digital

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, maksudnya adalah citra dan keluaranya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk kedalam bidang ini juga adalah pemampatan citra (image compression).

Citra

citra

citra

**Gambar 2.2** Blok sistem pengolahan citra digital [1]

Pengubahan kontras citra seperti pada Gambar 2.2 adalah contoh operasi pengolahan citra. Contoh operasi pengolahan citra lainnya adalah penghilangan derau (*noise*) pada citra Lena (Gambar 2.3). Citra Lena yang disebelah kiri mengandung derau berupa bintik-bintik putih (derau). Operasi penapisan *(filtering)* yang akan dibahas adalah *grayscaling*, *smoothing*, dan *Laplace*.



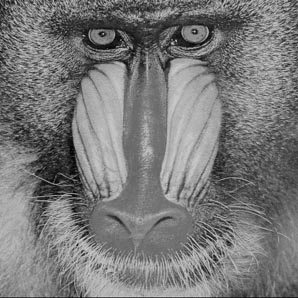
(a) (b)

**Gambar 2.3** (a) Citra Lena yang mengandung derau, (b) Hasil dari penapisan derau[1]

### 2.2.1 Greyscaling

*Greyscaling* adalah teknik yang digunakan untuk mengubah citra berwarna menjadi bentuk citra monokrom (hitam - putih). Pada citra berwarna yang terdiri dari 3 komponen warna R,G,B sedangkan pada citra monokrom hanya terdiri dari 1 komponen warna. Pengubahan citra berwarna menjadi citra *greyscale* mengikuti kaidah berikut :

|  |  |
| --- | --- |
|  | (2.1) |



(a) (b)

**Gambar 2.4** (a) Gambar berwarna dan (b) Gambar *Grayscale*[1]

### 2.2.2 Smoothing

Perbaikan citra dengan *smoothing filter* bertujuan untuk menghilangkan noise, memperjelas sisi (*sharp edge*) dalam sebuah image. *Smoothing filter* dibagi lagi menjadi empat jenis tipe : *blur, gaussian, median,* dan *bilateral*. Metode yang saya gunakan adalah *gaussian filter*. *Gaussian Filtering* adalah filter yang ideal yang mampu mengurangi besarnya frekuensi spasial yang tinggi dalam foto sebanding dengan frekuensi mereka. Artinya, metode ini mampu mengurangi besarnya frekuensi spasial yang tinggi lagi. *Gaussian Filtering* mampu meluas hingga ke segala arah, tetapi karena mendekati nol secara eksponensial, dapat dipotong tiga atau empat standar deviasi dari pusat tanpa mempengaruhi hasilnya.

*Gaussian filtering* mampu bekerja lebih cepat dari metode mean dan median dengan memisahkan sebuah *Gaussian* 2-D menjadi dua *Gaussian* 1-D, G (x, y) = G (x) G (y), dan melakukan *Filtering* dalam 1-D, baris demi baris dan kemudian kolom ke kolom. Dasar dari *Gaussian filtering* dapat dirumuskan sebagai berikut:

|  |  |
| --- | --- |
|  | (2.2) |

dimana,

h(x, y) = Gaussian filter 2-D pixel (x, y)

g2D (x, y) = Gaussian filter 2-D pixel (x, y)

 = *Bandwith Kernel*, Standar Populasi Kernel

g1D (x) = Gaussian filter 1-D line (x)

g1D (y) = Gaussian filter 1-D line (y)

Ada beberapa cara yang berbeda untuk mengimplementasikan *Gaussian filtering*, salah satunya menggunakan *Spatial Filter*, konvulusi ini menggunakan Operator *Gaussian Smoothing* 2-D yang mirip dengan filter 1-D, tetapi menggunakan kernel yang berbeda yang mewakili bentuk *Gaussian* 1-D untuk nilai X dan nilai *Gaussian* 1-D untuk nilai Y. Sehingga konvulusi ini termasuk tipe non linear kernel. Dalam *Spatial Filter* digunakan persamaan :

|  |  |
| --- | --- |
|  | (2.3) |

Ide *Gaussian smoothing* ini adalah dengan menggunakan distribusi ini 2-D sebagai fungsi ‘titik-menyebar’ atau lebih dikenal dengan ‘*point-spread*’, dan ini dicapai dengan konvolusi. Gambar disimpan sebagai koleksi *discrete pixels*, sehingga perlu dihasilkan pendekatan diskrit ke fungsi *Gaussian* sebelum kita dapat melakukan konvolusi tersebut. Secara teori, distribusi *Gaussian* adalah nol-nol di mana-mana, yang akan membutuhkan sebuah *kernel* konvolusi besar tak berhingga, tetapi dalam prakteknya nol efektif lebih dari sekitar tiga standar deviasi dari mean, dan sehingga dapat dilakukan pemotongan *kernel*.

Operator berikut akan menunjukan sebuah *kernel* konvolusi yang memiliki nilai integer sebuah *Kernel Gaussian* yang mendekati = 1,4.



**Gambar 2.5 *Kernel Operator Gaussian* dengan = 1,4**



(a) (b)

**Gambar 2.6** (a) Sebelum dan (b) Sesudah[1]

### 2.2.3 Laplace

Secara formal, fungsi 2-D dari laplacian didefinisikan jumlah dari derivatif kedua:

|  |  |
| --- | --- |
|  | (2.4) |

Dalam bentuk paling sederhana, dalam didekati dengan menggunakan kernel (3x3) berikut ini:





**Gambar 2.7** Efek Laplace Filter yang kedua (-1,8,-1)[12]

*Laplacian filtering* digunakan untuk mencari pinggir atau *outline* dari gambar dalam citra digital. Pada contoh dibawah garis mata, pinggiran baju, bentuk kepala kelihatan jelas. Gambar sebelah kanan adalah hasil laplace filter yang dinegatifkan atau *invert*, disitu terlihat seperti kita menggambar *outline* sebuah foto.

## 2.3Pengenalan Pola Citra Digital

Pengenalan pola mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnyaa karena otak manusia telah belajar mengklasifikasikan objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek didalam citra.

Citra

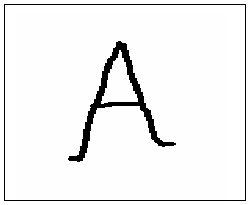
citra

deskripsi

objek

**Gambar 2.8** Blok sistem pengenalan pola citra digital

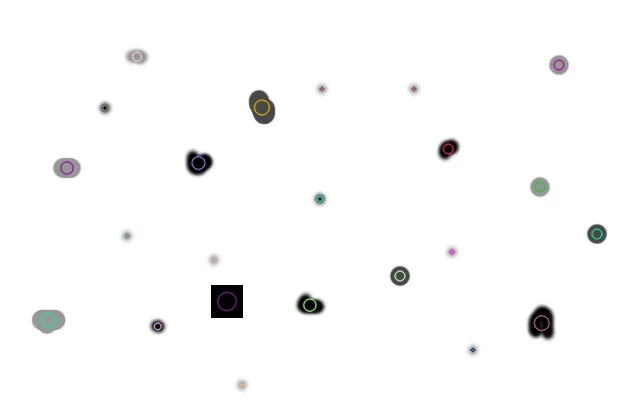
Contoh pengenalan pola misalnya citra pada Gambar 2.8 adalah tulisan tangan yang digunakan sebagai data masukan untuk mengenali karakter ‘A’. Dengan menggunakan suatu alogaritma pengenalan pola, diharapkan komputer dapat mengenali bahwa karakter tersebut adalah ‘A’.



**Gambar 2.9** Citra karakter ‘A’ yang digunakan sebagai masukan untuk pengenalan huruf

### 2.3.1 Blob Detection

*Blob* adalah kelompok titik yang saling terhubung dalam sebuah gambar yang mempunyai nilai *grayscale*. Pada gambar dibawah terlihat terdapat daerah yang gelap saling berpisah, daerah yang gelap tersebut kita sebut dengan *blob* dan tujuan dari *blob detection* adalah untuk mengidentifikasi daerah yang gelap itu dengan cara menandainya dengan *marker*.



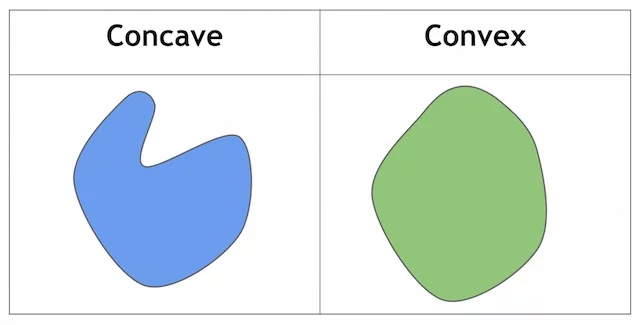
**Gambar 2.10** Citra *Blob* pada sebuah *image*

[ Sumber : https://www.learnopencv.com ]

Untuk melakukan Blob detection kita harus mengubah dulu *image* RGB menjadi *grayscale* dan setelah image menjadi *grayscale* kita harus mengubah *image* tersebut menjadi data biner dengan mengubahnya lagi kedalam *image threshold*. Dengan menggunakan *blob detection* kita bisa mendapatkan titik tengah dan radius *blob* tersebut.

Ada banyak parameter filter yang bisa kita gunakan dengan menggunakan blob detection yaitu :

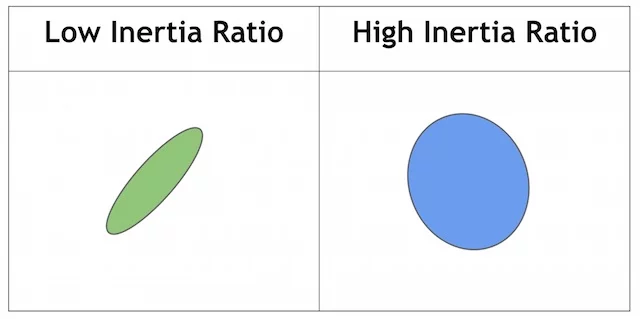
1. Dengan warna, kita bisa memilih warna blob yang ingin kita deteksi. Dari warna blob yang gelap hingga yang terang.
2. Dengan ukuran, ukuran sebuah blob dapat kita pilih juga untuk dideteksi dari yang skala kecil sampai terbesar.
3. Dengan bentuk, untuk filter bentuk kita dapat memilih 3 parameter.
4. Lingkaran.
5. Kecembungan.



**Gambar 2.11** Gambar *blob* yang cembungnya berbeda

[ Sumber : https://www.learnopencv.com ]

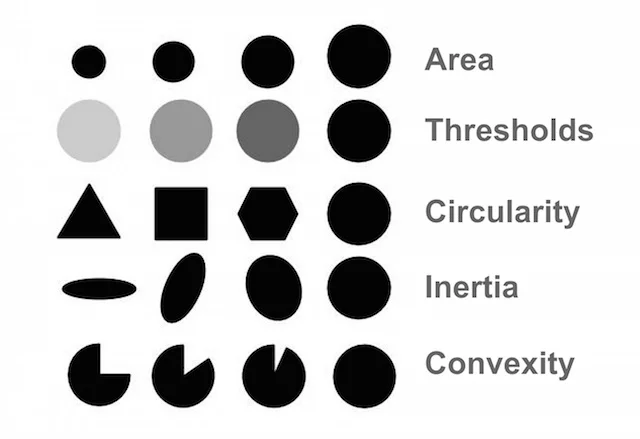
1. *Inertia Ratio*.



**Gambar 2.12** Gambar blob dengan *Inertia Ratio* yang berbeda

[ Sumber : https://www.learnopencv.com ]

Lebih rincinya *blob* detection bisa mendeteksi sebuah bentuk lingkaran dengan berbagai macam parameter yang berbeda-beda tergantung dengan kebutuh deteksi yang ingin kita dapatkan. Kelebihan menggunakan *blob detection* kita dapat mendeteksi secara cepat tetapi kita tidak dapat mengambil data titik tengah dari *blob* tersebut.



**Gambar 2.13** Gambar filter parameter *blob detection*

[ Sumber : https://www.learnopencv.com ]

### 2.3.2 Find Contour

Di SMP kita pernah belajar tentang berbagai bentuk bidang geometri. Itu masih relatif mudah untuk mencari titik tengah dari berbagai bentuk standar geometri seperti lingkaran, kotak, segitiga, elips, dan lain-lain.

Tetapi ketika kita ingin mencari bentuk bidang yang tidak beraturan polanya. Metode ini susah kita lakukan tanpa menggunakan kalkulus matematika, masalah yang sama timbul saat menemukan titik tengah ketika menggunakan *computer vision*. Kita dapat menemukan titik pusat dari *blob* menggunakan *moments* di OpenCV*.* Tetapi pertama-tama, kita harus mengetahui apakah itu *image moments*. *Image moments* adalah rata-rata titik yang terhubung antara satu dengan yang lain yang saling menyilang dari inensitas *pixel* suatu *blob*, yang mana akan kita dapatkan jarak spesifik dari suatu *blob* seperti radius, area dan titik tengah. Untuk mencari titik tengah, kita biasanya mengkonversikannya kedalam bentuk biner dan setelah itu kita dapat mencari titik tengah tersebut.

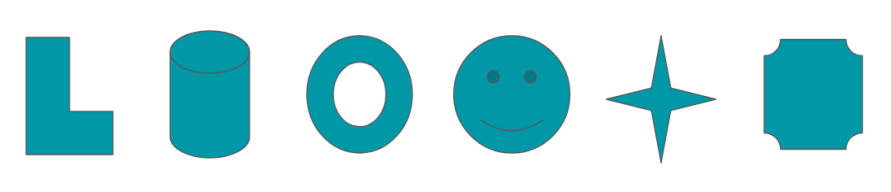
Formula titik tengah yang kita pakai :

\[C_x = \cfrac{M_{10}}{M_{00}}\]\[C_y = \cfrac{M_{01}}{M_{00}}\]

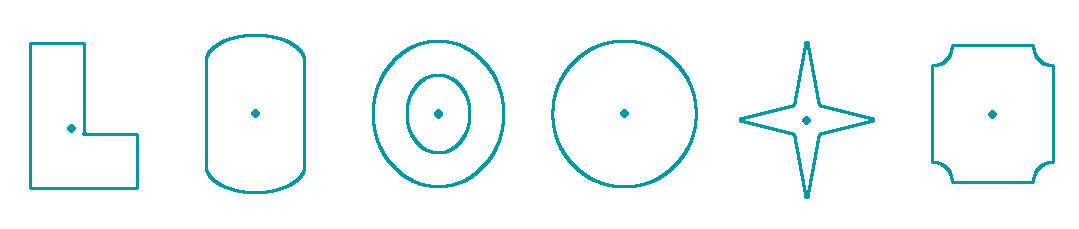
Cx adalah koordinat x dan Cy adalah koordinat y dari titik pusat dan M adalah titik yang menunjukan *moments* tersebut.

Adapun langkah-langkah untuk mencari titik tengah di blob yang kita inginkan di image processing sebagai berikut :

1. Konversi image dalam bentuk *grayscale*.
2. Lakukan Binarisasi pada *image*.
3. Mencari titik tengah pada blob setelah menhitung *moments*.



**Gambar 2.14** *Image* dengan bermacam bentuk bidang



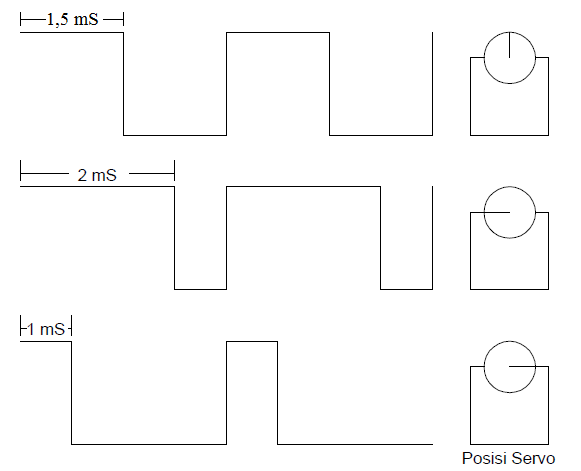
**Gambar 2.15** *Image* dengan titik tengah pada *blob*.

[ Sumber : https://www.learnopencv.com ]

## 2.4Motor Servo

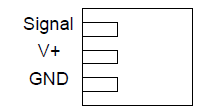
Berbeda dengan motor DC (*Direct Current*) dan motor *Stepper*, motor servo adalah motor dengan system closed feedback dimana poisi dari motor akan diinformasikan kembali ke rangkaian control yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian control.

Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 mS sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu kearah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.



**Gambar 2.16** Teknik PWM (Pulse Width Modulation) untuk mengatur motor servo

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontiyu seperti motor DC maupun motor stepper. Walaupun demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu.

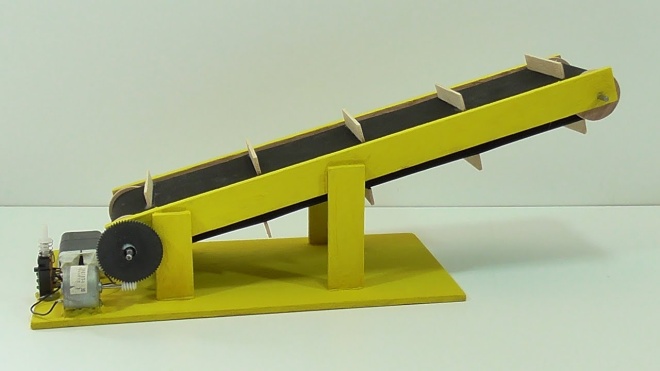


**Gambar 2.17** Pin Out kabel motor servo

Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

## 2.5 Belt Conveyor

*Belt Conveyor* adalah sebuah peralatan yang sangat sederhana dan biasanya digunakan untuk mengangkut benda-benda kecil hingga yang memiliki kapasitas besar. Alat tersebut terdiri dari sabuk yang akan berkerja sebagai pengangkut benda. Sabuk yang digunakan pada belt conveyorini dapat dibuat dari berbagai jenis bahan tergantung sifat benda yang diangkut, selain itu kita juga membutuhkan motor dc sebagai penggerak sabuk conveyor tersebut.



**Gambar 2.18** Contoh *conveyor* DIY (Do It Yourself) menggunakan 1 motor DC

[ Sumber : <https://www.watanabhand.co.th> ]

Conveyor DIY nanti akan diracang dengan beberapa buah lampu LED (Light Emitting Diode) yang akan menerangi *coin*, sehingga citra digital yang akan ditangkap kamera tidak akan terganggu

# BAB III

# METODE PENELITIAN

## 3.1 Waktu dan Tempat Penelitian

Bagian ini adalah memuat tentang waktu dan tempat anda melakukan penelitian. Penelitian ini saya buat selama 1 tahun, pembuatan alat ini hanya dilakukan dirumah yang beralamatkan di Jl. Siaga

Tabel. 3.1 Jenis Kegiatan dan Waktu

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Jenis Kegiatan** | **Bulan** | | | | | | | | | | | |
| **9** | **10** | **11** | **12** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **1** | Persiapan Penelitian | **x** | **x** |  |  |  |  |  |  |  |  |  |  |
| **2** | Studi Kepustakaan (Kajian Literatur) |  | **x** | **x** |  |  |  |  |  |  |  |  |  |
| **3** | Menyusun instrument penelitian |  |  | **x** | **x** |  |  |  |  |  |  |  |  |
| **4** | Pengumpulan data |  |  |  | **x** | **x** | **x** |  |  |  |  |  |  |
| **5** | Analisis Data |  |  |  |  |  | **x** | **x** | **x** |  |  |  |  |
| **6** | Penyusunan Laporan |  |  |  |  |  |  |  | **x** | **x** | **x** |  |  |
| **7** | Penyusunan Naskah Publikasi ke Jurnal Ilmiah |  |  |  |  |  |  |  |  |  | **x** | **x** |  |
| **8** | Evaluasi Akhir Proses Penelitian |  |  |  |  |  |  |  |  |  |  | **x** | **x** |

## 3.2 Metode Pengumpulan Data

Metode pengumpulan data berasala dari Studi Literatur yaitu

1. Mempelajari prinsip *Image Processing* dalam teknologi *Machine Vision* yaitu akuisisi citra digital dan pemrosesan citra digital.
2. Mempelajari pemrograman visual menggunakan Microsoft Visual C++ 2017*Community edition* dan *Open Source Computer Vision Library* (OpenCV).

## 3.3 Instrumen Penelitian

Instrumen pada skripsi ini menggunakan alat dan bahan sebagai berikut :

1. Conveyor
2. Kamera
3. Powersupply
4. Lampu
5. Motor servo
6. Motor dc
7. Mikrokontroler
8. Motor driver
9. Laptop

## 3.4 Jalannya Penelitian

Start

Preprocessing

Matching  
detection ?

Query Frame

Detection

Tracking

Scaning

Learn

Send to Data Serial

Push the exit button ?

Finish

Y

Y

N

N

Gambar 3.1 Diagram Alir Penelitian

## 3.5 Metode Analisis Data

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibuat. Dengan parameter keberhasilannya adalah dapat meidentifikasi bentuk coin tersebut secara benar.

Apabila hasil yang diperoleh dari tahap pengujian terjadi ketidaksesuaian pada alat yang telah dibuat atau mungkin kurang memuaskan secara bentuk dan ketelitian penyelesaian, maka perlu diadakan evaluasi pada software dan hardware serta sistem secara keseluruhan. Kemudian dilakukan pengujian ulang sampai parameter keberhasilan telah dicapai.

# BAB IV

# PEMBAHASAN DAN HASIL

Pada tugas akhir ini, sistem pendeteksian *coin* dirancang dengan menggunakan komputer sebagai pusat pengolah data citra dan aktuator pada motor servo, motor servo sebagai penutup gerbang pada *coin* yang bergerak di *conveyor*, pada proses ini coin akan disegmentasi sesuai jenisnya. Diagram blok sistem ditunjukkan pada gambar 3.1.

.



Kamera

*Input* posisi *coin*

*Coin*

Motor  
Servo



**PC**

Computer Vision

Arduino



**Gambar 4.1** Diagram Blok Sistem

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]

Cara kerja keseluruhan sistem adalah:.

1. Kamera digital sebagai sensor visual menangkap citra dari *coin* dan menampilkannya di komputer. Setelah komputer menerima citra, citra tersebut akan diolah dengan menggunakan *image processing*.
2. *Software* akan memulai pendeteksian pada lingkaran coin berwarna hitam yang mempunyai pola lingkaran. Setelah *software* dapat mengenali *coin*, selanjutnya dilakukan metode *tracking* pada *coin* untuk diambil datanya berupa titik tengah dan jarak diameter *coin*.
3. Setelah coin diambil datanya (titik tengah dan diameter) maka selanjutnya data akan di simpan ke dalam *memory* komputer dan akan dibandingkan dengan data *coin* yang lainnya.
4. Setelah selesai dibandingkan datanya, selanjutnya komputer akan mengirim data ke minimum sistemArduino melalui komunikasi serial USB untuk mengarahkan motor servo untuk menutup gerbang *conveyor* agar bisa disegmentasi.

## Kamera Sebagai Sensor Visual

Modul kamera yang digunakan sebagai sensor visual dipilih kamera digital jenis *web camera* bermerk Logitech HD tipe Webcam C525 serta memiliki fitur sebagai berikut :

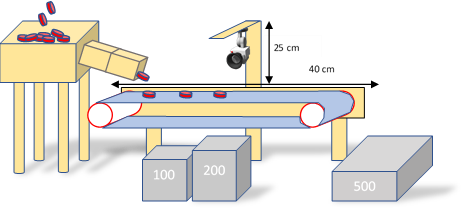
1. *Image sensor* : *True* 1.3-*megapixel sensor*
2. *Resolution* : 720 x 480 *pixels*
3. *Frame rate* : Di atas 30 *frame per second*
4. *Interface* : Hi-speed USB

.

**Gambar 4.2** Logitech HD Webcam C525

[ Sumber : <https://www.ebuyer.com/741487-logitech-hd-webcam-c525-usb-960-001064> ]

Adapun hal yang terpenting dalam pemilihan sensor visual pada sistem ini yaitu menggunakan kamera yang memiliki *image sensor* 1,3 *megapixel* agar diperoleh gambar yang tajam, serta memiliki *frame rate* yang tinggi*.* Hal tersebut dilakukan untuk membantu proses pengolahan citra agar citra yang dikirim ke komputer memiliki kualitas yang lebih presisi dan lebih cepat.



Gambar 4. 3 Posisi kamera sebagai sensor visual terhadap *coin*.

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]

Pada perancangan ini, kamera sebagai sensor visual diatur posisinya terhadap *coin*. Dimana jarak antara lensa kamera dengan coin dapat dengan jarak minimum sebesar 25 cm serta posisi sudut tangkap kamera diatur sebesar 180° terhadap garis normal horisontal. Selain itu ditambahkan 10 buah Strip LED (*Light Emitting Dioda)* yang masing-masing diletakan di sebelah kanan dan kiri kamera. Fungsi dari ketiga lampu tersebut adalah untuk mengatur intensitas cahaya sekitar kamera dan coin*.*

Selain perancangan posisi untuk kamera sebagai sensor visual, kamera tersebut juga dilakukan perancangan untuk pengaturan resolusi citra yang akan ditampilkan pada monitor. Kamera sebagai sensor visual ini diatur dengan resolusi citra output sebesar 320 x 240.

## Central Processing Unit

CPU *(Central Processing Unit*) sebagai perangkat yang bertugas untuk mengolah data. Kecepatan dalam pengolahan data ditentukan oleh spesifikasi CPU. Pada sistem robot rubik ini digunakan CPU dengan spesifikasi sebagai berikut :

1. HP 4 COMPUTE CORES 2C+2G 2.20 GHz
2. 4 GB *Memory*
3. AMD A4-9120 RADEON R3

Pemilihan spesifikasi dari CPU ini didasarkan pertimbangan bahwa pada sistem yang menggunakan teknik pengolahan citra digital harus memiliki kecepatan tinggi dan kapasitas memori yang besar.



**Gambar 4.4** Laptop HP 14-bw005au

[ Sumber : https://www.arenalaptop.com/harga/hp-14-bw005au ]

## Perancangan Perangkat Keras

Secara garis besar perancangan perangkat keras merupakan pembuatan *conveyorcoin*. *Conveyor* ini dirancang dengan spesifikasi sebagai berikut :

* 1. Conyeor ini menggunakan dua motor servo. Motor I digunakan untuk menutup jalan *coin* 200 dan motor II digunakan untuk menutup jalan *coin* 100.
  2. Conveyor ini dirancang dengan menggunakan minimum sistem Arduino. Arduino adalah pengendali mikro single-board yang bersifat open-source.
  3. Conveyor ini dilengkapi dengan power supply 12v 5A yang mana untuk memberikan tegangan pada 2 motor DC 5v. Terdapat juga adaptor universal 5v yang digunakan untuk menambah arus pada kontroller arduino.
  4. Pada motor DC juga dilengkapi dengan 2 motor driver untuk mengatur kecepatan motor.



**Gambar 4.5** Perangkat keras berupa dua motor servo, minimum sistem Arduino, power supply dan dua motor driver.

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]

### Pengontrolan Motor Servo

Sistem ini digunakan untuk mengontrol motor servo standar yang dapat diatur posisinya antara 0o sampai dengan 180o yang dipasang pada bagian samping *conveyor*. Motor servo ini digunakan untuk menutup jalan *coin* pada *conveyor* dengan sudut 30o.

Untuk pengotrolan motor servo yang bergerak 30o, prosesor kedua (Arduino) mengirimkan data/sinyal pulsa (kontinyu) yang lebarnya 1.25 ms selama interval waktu tertentu sehingga sudut yang dicapai motor servo sebesar 30o. Pada posisi ini saya namakan STATUS OFF. Setelah itu prosesor kedua akan menunggu data yang dikirim oleh CPU untuk mengetahui apakah ada *coin* yang berada di *conveyor*. Motor servo akan bergerak kebawah bila prosesor utama akan mengirim pulsa 1 ms (sebesar 0o) dan posisi ini saya namakan STATUS ON.

CPU

Adruino

Motor Servo

**Gambar 4.6** Blok diagram sistem pergerakan motor servo

30oStatus OFF

**Gambar 4.7** Servo motor pada saat dalam posisi di atas pulsa 0.25 ms.

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]

0oStatus ON

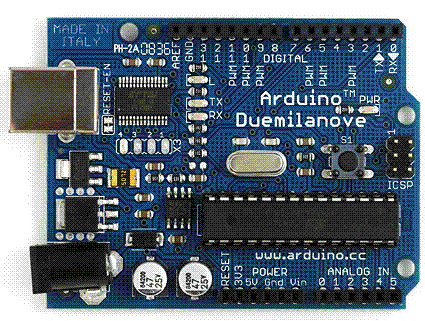
**Gambar 4.8** Servo motor pada saat dalam posisi di bawah pulsa 1 ms.

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]

### Minimum Sistem Arduino

Arduino board adalah modul yang menggunakan mikrokontroler AVR dan menggunakan seri yang lebih canggih, sehingga dapat digunakan untuk membangun sistem elektronika berukuran minimalis namun handal dan cepat. Berbagai modul dan sensor terkini dapat dipasang pada board ini dilengkapi dengan berbagai kode demo yang memuaskan.

Arduino terdiri dari beberapa board, yang dapat digunakan sesuai kebutuhan dan menggunakan software open source yang dapat dijalankan pada Windows, Mac dan Linux. Beberapa board yang terkenal ditampilkan pada gambar di bawah :



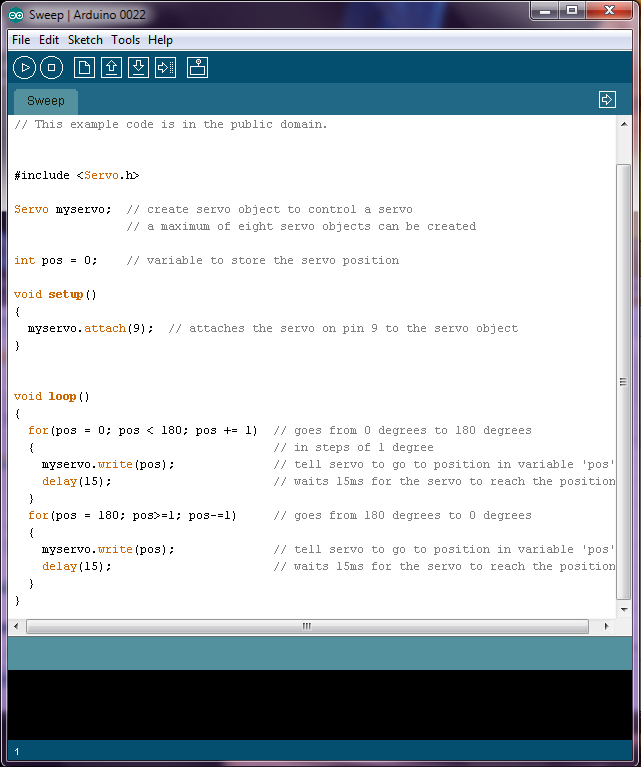
**Gambar 4.9** Arduino Main Board berbasis ATmega328 dengan bootloader[14]

[ Sumber : <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all> ]

**Spesifikasi :**

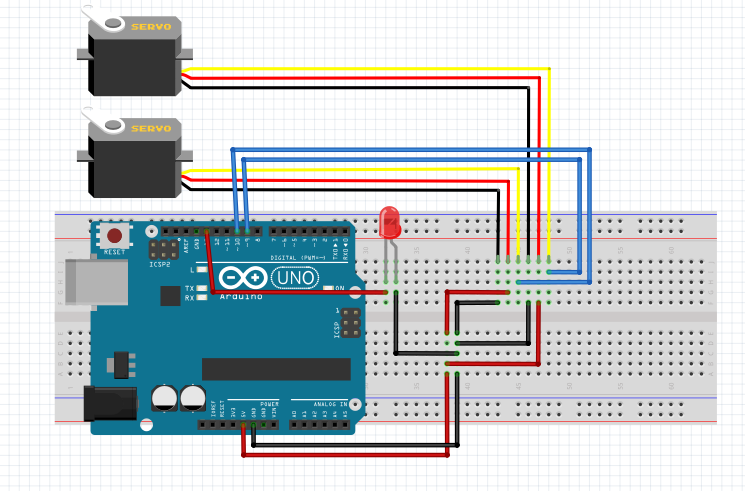
* Microcontroller : Atmega 328
* Operating Voltage : 5V
* Input Voltage (recommended) : 7-12V
* Input Voltage (limits) : 6-20V
* Digital I/O Pins : 14 (6 PWM output)
* Analog Input Pins : 6
* DC Current per I/O Pin : 40 mA
* DC Current for 3.3V Pin : 50 mA
* Flash Memory : 32 KB
* SRAM : 2 KB
* EEPROM : 1 KB
* Clock Speed : 16 MHz

Mikrokontroler Arduino ini menggunakan Arduino programming language berbasiskan Wiring dan Arduino development environment berbasiskan Processing. Arduino menggunakan koneksi USB (*Universal Serial Bus*) menggunakan chip FTDI (*Future Technology Devices International*) untuk melakukan pemrograman, dan biasanya pada chip Arduino sudah dimasukkan bootloader, sehingga dapat dilakukan pemrograman langsung ke dalam chip menggunakan software Arduino.



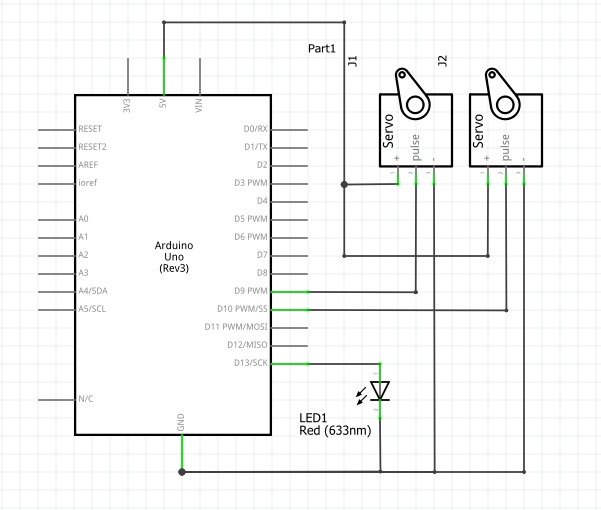
**Gambar 4.10** Contoh program Arduino menggunakan Java.

[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]



**Gambar 4.11** Breadboard rangkaian motor servo menggunakan Arduino

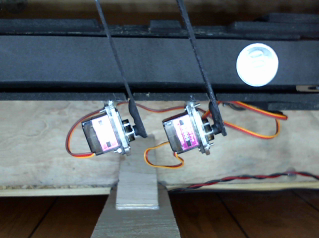
[ Sumber : [penulis](https://learn.sparkfun.com/tutorials/what-is-an-arduino/all) ]



**Gambar 4.12** Schematic rangkaian *microcontroller*

## Pengolahan dan Pengenalan Pola Citra Digital

Pada pendeteksi coin, citra akan diambil gambarnya dan diolah secara real time, yang akan kita bahas disini menggunakan bahasa Visual C++ dan OpenCV library. Untuk meningkatkan efisiensi, kita mengubah ukuran frame menjadi 320x240 pixels. Standarisasi ukuran gambar adalah ide baik yang tidak hanya untuk efesiensi saja, tetapi juga karena parameter yang kita harus miliki agar dapat bekerja lebih cepat.



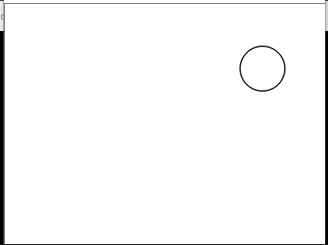
**Gambar 4.13** Citra yang diambil dengan kamera

Setelah gagal menemukan titik detektor yang baik untuk tugas akhir ini, keputusan dibuat untuk melanjutkan dengan pendeketan berbasis blob detection. Hal ini menunjukan bahwa pendekatan seharusnya bekerja dengan sangat baik kerena blob detection akan mencari bentuk lingkaran pada *coin*. Tahap pertama dari alogaritma ini adalah dengan mengubah citra dalam bentuk *threshold inverter*. Pada tahap ini *blob detection* baru bisa berkerja untuk menemukan lingkaran yang ingin kita deteksi. Di blob detection kita bisa menggunakan beberapa fitur yang bisa kita gunakan untuk membatasi filter lingkaran yang kita inginkan agar tidak terlalu banyak noise lingkaran-lingkaran kecil yang tidak kita inginkan.



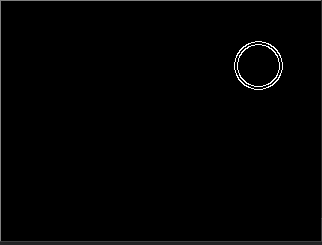
**Gambar 4.14** Citra threshold invert

Lingkaran yang kita dapatkan kita ambil dan kita taruh di citra yang kosong agar noise yang kita tidak inginkan tidak mengganggu dalam pendeteksian berikutnya.



**Gambar 4.15** Citra FilterBlob

Di blob detection kita hanya bisa mendeteksi lingkaran tetapi kita tidak bisa mendapatkan titik tengah dan sisi lingkaran luar coin. Untuk itu kita perlu menggunakan beberapa filter lagi, filter saya yang gunakan berikutanya adalah canny filter. Canny filter ini akan mencari tiap garis yang ada di citra digital, oleh sebab itu citra harus bersih dari noise yang kita tidak inginkan karena dapat memberatkan kita untuk menfilter data yang kita inginkan.



**Gambar 4.16** Citra Filter Canny

Pada citra canny kita bisa melihat lingkaran putih, sebenarnya itu bukan garis lingkaran putih tetapi garis-garis pendek yang jumlahnya ratusan sampai membentuk lingkaran. Di garis-garis tersebut lah kita bisa mendapatkan moment pada lingkaran dengan mencari garis canny yang telah kita probabilistik untuk mencari sisi-sisi panjang. Canny Filter juga mengambil parameter threshold yang menentukan kepekaannya untuk mendeteksi sebua garis lingkaran. Untuk menhindari pengambilan titik yang melebihi batas skala untuk pergantian intensitas cahaya, maka saya memberi jumlah segem tepi yang dideteksi dari gambar kurang lebih 10 sampai 200 threshold setiap iterasinya.

Preprocessing

Tracking

Detection

Memory / Learn

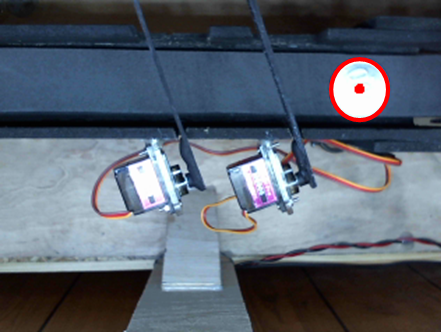
Output

update

Re-init

**Gambar 4.17** Design dan implementasi

|  |
| --- |
| // Change thresholds  params.minThreshold = 10;  params.maxThreshold = 200;  // Filter by Area.  params.filterByArea = true;  params.minArea = 900;  // Filter by Circularity  params.filterByCircularity = true;  params.minCircularity = 0.85;  // Filter by Convexity  params.filterByConvexity = true;  params.minConvexity = 0.1;  // Filter by Inertia  params.filterByInertia = true;  params.minInertiaRatio = 0.3; |



**Gambar 4.18** Tepi lingkaran yang terdeteksi Canny Filter setelah menggunakan find contour

Tujuan langkah selanjutnya adalah mengidentifikasi titik tengah. Dengan melihat posisi awal dan akhir dari semua garis, kita akan mencari dua garis yang saling terhubung atau saling berpotongan yang memiliki panjang yang hampir sama. Karena kita mencari diameter lingkaran dalam langkah ini, maka harus melakukan sedikit beberapa rumus vektor matematika dalam kasus yang terakhir. Sebenarnya sistemnya sama dengan mendeteksi garis menggunakan Hough Transform misalkan kita mempunyai sepasang garis yang telah terseleksi, garis tersebut harus mempunyai panjang kurang lebih 30% yang sama bila tidak dibuang. kita juga akan membuang sepasang garis jika dari dua sudut antara mereka yang lebih rendah dari 0.5 radians (atau kira-kira 30derajat), karena ini menunjukan garis-garis yang hampir sejajar.

**Grafik 4.1** Empat garis vertikal yang memiliki jarak berbeda-beda

Misalkan terdapat empat garis p, q, r dan t. Dimana masing-masing garis mempunyai titik:

* p0(1, 1) dan p1(1, 4)
* q0(2, 1) dan q1(2, 4)
* r0(3, 1) dan r1(3, 3)
* t0(4, 1) dan t1(4, 3.5)

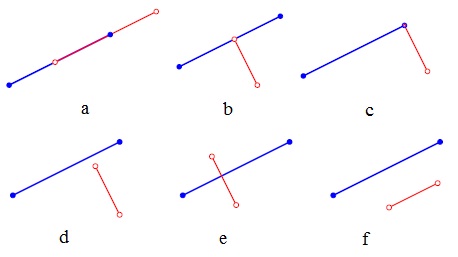
|  |  |
| --- | --- |
| Mencari jarak antara garis : | (4.1) |

Dengan menggunakan rumus 3.2 kita dapat menseleksi garis yang kita inginkan:

|  |  |
| --- | --- |
| dimana2, 3 dan 4 | (4.2) |

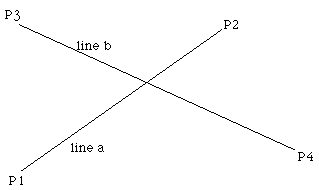
Dengan hasil jarak pada masing-masing garis , ,  dan , maka dapat kita simpulkan bahwa ,danmempunyai jarak lebih kecil dari 1,3 dan mempunyai jarak sebaliknya yang harus kita buang.

Proses selanjutnya kita mencari dua garis yang saling berpotongan untuk diseleksi (filter) lagi. Pada tahap dua garis yang saling berpotongan akan kita ambil untuk diproses lagi dan garis yang tidak saling berpotongan akan kita buang.



**Gambar 4.19** Contoh dua garis yang saling berpotongan dan tidak berpotongan

Terlihat pada gambar 3.7 bahwa a, b, c dan e mempunyai garis yang saling berpotongan sedangkan d dan f tidak saling berpotongan. Misalkan kita mempunyai dua garis yang saling berpotongan dimana garis Pa mempunyai titik P1 dan P2 dan garis Pb mempunyai titik P3 dan P4.



**Gambar 4.20** Dua garis yang saling berpotongan pada sebuah titik

Persamaan garisnnya adalah





Dimana syarat garis yang saling berpotongan **Pa = Pb** maka dapat kita peroleh

|  |  |
| --- | --- |
|  | (4.3) |

Sehingga didapatkan penyelesaian untuk persamaan ua dan ub

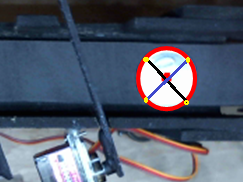
|  |  |
| --- | --- |
|  | (4.4) |

Jadi titik (x, y) yang saling berpotongan adalah

|  |  |
| --- | --- |
|  | (4.5) |

Jika dilihat dari persamaan diatas, kita dapat menentukan sebelumnya garis mana yang saling berpotongan dengan melihat bahwa pembagi antara ua dan ub memiliki persamaan yang sama, sehingga untuk den (garis yang saling berpotongan) yang lebih kecil dari 0,1 (bernilai 0) maka garis tersebut tidak saling berpotongan.

|  |  |
| --- | --- |
|  | (4.6) |
|  |  |



**Gambar 4.21**Hasil citra setelah dicari persamaan garis

Setelah menemukan koordinat coin di citra, proses selanjutnya kita tracking koordinat tersebut sehingga kita tidak perlu selalu mendeteksi posisi coin terus menerus. Hal ini memudah kita untuk mengambil data secara stabil dan membuat proses kerja lainnya lebih cepat. Metode tracking yang kita gunakan adalah menghitung iterasi pada setiap titik koordinat, apabila jumlah iterasi sudah terpenuhi maka kita tracking coin tersebut dan apabila belum memenuhi maka kita akan menndeteksi ulang, Kita juga memeriksa setiap iterasi yang kita tracking tidak mengalami kesalahan dengan terus menerus memeriksa diameter coin tersebut harus memenuhi syarat yaitu:

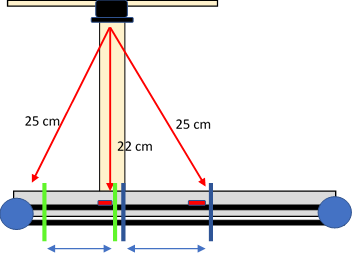
1. Jumlah yang kita tracking harus mempunyai diameter yang sama.
2. Jarak antara titik-titik yang kita tracking harus konstan.

Jika dua kondisi ini tidak terpenuhi maka kita akan kembali lagi ke proses deteksi. Karena kita membutuhkan data yang akurat.

## Proses Segmentasi Coin

Coin di Indonesia mempunyai diameter yang saling berdekatan dan tebal yang berbeda-beda, oleh sebab itu coin 1000 tidak dapat saya gunakan karena memiliki tebal yang berbeda dan diameter yang sama dengan coin 200. Kerena kita yang menggunakan 1 kamera saja dari atas maka kita hanya dapat membandingkan diameter coin dan tidak dapat membandingkan tebal coin.

Pada segmentasi coin saya memberikan batasan jarak pada pendeteksian, ini bertujuan untuk menghilangkan data noise yang mengganggu pada pengiriman serial komunikasi motor servo. Misalkan pada gambar diameter coin 200 pada wilayah hijau akan berbeda dengan jarak diameter pada wilayah biru. Ini disebabkan jarak vertikal antara kamera dan conveyor berbeda, oleh sebab itu saya memberikan batasan untuk mendeteksi coin 200 hanya pada wilayah hijau dan coin 100 pada wilayah hijau dan biru. Karena coin 100 akan terdeteksi 200 apabila coin 200 juga saya deteksi diwilayah biru, hal ini akan membuat proses alogaritma pengiriman data ke arduino menjadi kacau.



**Gambar 4.22**Hasil citra setelah dicari persamaan garis

Untuk proses *soft computing* pada *coin*, kita menggunakan metode yang hampir mirip sama dengan logika fuzzy. Sebagian besar dari kita diajarkan dari usia yang sangat muda untuk melihat dunia dalam hal hitam dan putih, A-atau-bukan-A, Boolean 1 atau 0. Banyak ilmu pengetahuan, matematika, logika, dan bahkan budaya mengasumsikan dunia 1 dan 0 itu, benar atau salah, panas atau dingin, A-atau-tidak-A. Apakah ada atau setengah setengah hilang? Apakah setengah gelas penuh atau setengah kosong? Apakah mobil akan cepat atau lambat? Masing-masing pertanyaan ini mengenai hal  grayness di dunia biasanya menjelaskan dalam warna hitam dan putih*.*Jika 2 daerah direlasikan dengan implikasi sederhana sebagai berikut :

IF x is A THEN y is B

Transfer fungsi:

y = f((x,a), B)

Grayness (data abu-abu) yang saya maksud adalah sewaktu-waktu coin 100 terdeteksi menjadi coin 200, tetapi hanya dalam beberapa iterasi pendek saja, sedangkan iterasi pendeteksian coin 100 lebih banyak daripada grayness. Maka soft computing tetap memerintahkan alogaritma untuk memilih data 100 yang tetap terdeteksi dibandingkan coin 200. Demikian untuk coin 200 apabila ada data grayness maka akan dihitung iterasi data yang didapatkan. Data-data yang saya gunakan sebagai berikut:

1. Data ON1 adalah data coin 200 terdeteksi apabila iterasi terpenuhi dan motor servo1 akan menutup jalan conveyor.
2. Data OFF1 adalah data coin 200 tidak terdeteksi dan motor servo1 akan membuka ke atas.
3. Data ON 2 adalah data coin 100 terdeteksi apabila iterasi terpenuhi dan motor servo2 akan menutup jalan conveyor.
4. Data OFF2 adalah data coin 100 tidak terdeteksi dan motor servo2 akan membuka ke atas.
5. Data OFF3 adalah data dimana tidak ada coin sama sekali yang terdeteksi pada citra, motor servo1 daan motor servo2 akan membuka ke atas bersamaan

## Pengujian Gerak Motor Servo

Pada pengujian gerak motor servo, dilakukan dengan cara 10 kali keberhasilan motor servo untuk menggerakkan gerbang tutup dan buka dalam pendeteksian coin dengan conyeor yang tidak berjalan.

**Tabel 4. 1** Hasil pengujian pergerakan motor servo tutup

|  |  |  |
| --- | --- | --- |
| Percobaan | Coin 100 | Coin 200 |
| 1 | Sukses | Sukses |
| 2 | Sukses | Sukses |
| 3 | Sukses | Sukses |
| 4 | Sukses | Sukses |
| 5 | Gagal | Sukses |
| 6 | Sukses | Sukses |
| 7 | Sukses | Sukses |
| 8 | Sukses | Gagal |
| 9 | Sukses | Sukses |
| 10 | Sukses | Sukses |

**Tabel 4. 2** Hasil pengujian pergerakan motor servo buka

|  |  |  |
| --- | --- | --- |
| Percobaan | Coin 100 | Coin 200 |
| 1 | Sukses | Sukses |
| 2 | Sukses | Sukses |
| 3 | Sukses | Sukses |
| 4 | Sukses | Sukses |
| 5 | Gagal | Sukses |
| 6 | Sukses | Sukses |
| 7 | Sukses | Sukses |
| 8 | Sukses | Sukses |
| 9 | Sukses | Sukses |
| 10 | Sukses | Sukses |

**Tabel 4. 3** Hasil pengujian pergerakan motor servo buka tidak terdapat coin sama sekali.

|  |  |
| --- | --- |
| Percobaan | Tidak ada Coin |
| 1 | Sukses |
| 2 | Sukses |
| 3 | Sukses |
| 4 | Sukses |
| 5 | Sukses |
| 6 | Sukses |
| 7 | Sukses |
| 8 | Sukses |
| 9 | Sukses |
| 10 | Sukses |

## Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan pengujian terhadap program pendeteksian coin dan motor servo dengan conveyor yang berjalan. Tujuan dari pengujian ini adalah untuk mendapatkan kondisi coin dari user agar sistem bekerja pada tingkat keberhasilan maksimal.

### Pengujian Deteksi Coin

Pengujian deteksi coin dilakukan untuk mengetahui faktor-faktor yang mempengaruhi proses deteksi coin dan pengujian ini dilakukan pada siang hari dan malam hari dengan tujuan untuk mencoba intensitas cahaya pada coin yang berbeda.

Tabel 4. 4Hasil pengujian deteksi coin pada siang hari

|  |  |  |
| --- | --- | --- |
| Gambar | Hasil | Threshold |
|  | Sukses | 76 |
|  | Sukses | 74 |
|  | Sukses | 70 |
|  | Sukses | 68 |
|  | Sukses | 69 |
|  | Sukses | 72 |

Tabel 4. 5Hasil pengujian deteksi coin pada malam hari

|  |  |  |
| --- | --- | --- |
| Gambar | Hasil | Threshold |
|  | Sukses | 64 |
|  | Sukses | 88 |
|  | Sukses | 64 |
|  | Sukses | 75 |
|  | Sukses | 86 |
|  | Sukses | 75 |

### Pengujian Segmentasi Coin

Pengujian Segmentasi coin merupakan salah satu data yang sangat penting sebelum proses segmentasi. Dimana coin akan dipisahkan sesuai kelompoknya pada conveyor yang berjalan.

Tabel 4. 6Hasil pengujian segmentasi coin

|  |  |
| --- | --- |
| Percobaan | Identifikasi  Warna dan penyelesaian |
| 1 | Sukses |
| 2 | Sukses |
| 3 | Sukses |
| 4 | Sukses |
| 5 | Sukses |
| 6 | Gagal |
| 7 | Sukses |
| 8 | Sukses |
| 9 | Gagal |
| 10 | Sukses |

## Pembahasan Permasalahan

Setelah dilakukan pengujian, didapatkan permasalahan yang perlu diatasi dalam tugas akhir ini. Ini ditujukan untuk mendapatkan hasil yang lebih baik dalam pendeteksian dan pembacaan coin yang ada. Pendeteksian rubik ini menggunakan metode *Canny* jadi sangat bergantung sekali dengan jumlah garis yang dapat dideteksi dan membentuk pola lingkaran.

Pada proses pengujian gerak motor servo, motor servo kadang tidak dapat bergerak karena ada data yang dikirim oleh CPU ke arduino yang menumpukk bersamaan, yang membuat motor servo menjadi error dan tidak mau bergerak.



Gambar 4.23Contoh motor servo error

Kesalahan pembacaan coin terjadi bukan karena metode pembacaan yang kurang akurat, tetapi terjadi akibat posisi *tracking* coin yang bergeser pada saat melakukan berjalan diatas conveyor.



Gambar 4. 24Contoh coin yang jatuh bersamaan

Pada proses pendeteksian coin, sinar matahari bukan merupakan salah satu faktor yang mempengaruhi pendeteksian. Dalam tugas akhir ini, ada beberapa masalah teknis yang bigin coin tidak bisa disegmentasi yaitu coin yang menumpuk bersamaan dan coin yang saling berdempetan.

# BAB V

# PENUTUP

Setelah dilakukan rangkaian kegiatan perancangan sistem dan pengujian alat penulis memperoleh kesimpulan dan memberikan beberapa saran sebagai berikut



## Kesimpulan

Kesimpulan yang diperoleh dalam Tugas Akhir ini adalah:

* 1. Masalah teknis coin yang berdempetan dan coin yang menumpuk perlu dilakukan mekanik yang lebih baik lagi.
  2. Hasil pengujian pergerakan robot dalam 20 kali percobaan didapatkan hasil dengan proses keberhasilan 95%.
  3. Hasil pengujian deteksi coin pada malam dan siang hari dalam 12 kali percobaan didapatkan hasil dengan proses keberhasilan 100%.
  4. Pengujian dari penggabungan sistem pergerakan motort dan sistem pendeteksian serta segmentasi coin yang dilakukan sebanyak 10 kali percobaan didapatkan hasil dengan proses keberhasilan 85%.

## Saran

Beberapa saran yang dapat penulis berikan untuk pengembangan Tugas Akhir adalah sebagai berikut :

Mekanik yang dibuat masih sangat sederhana jadi untuk segmentasi coin yang banyak membutuhkan waktu yang lama.

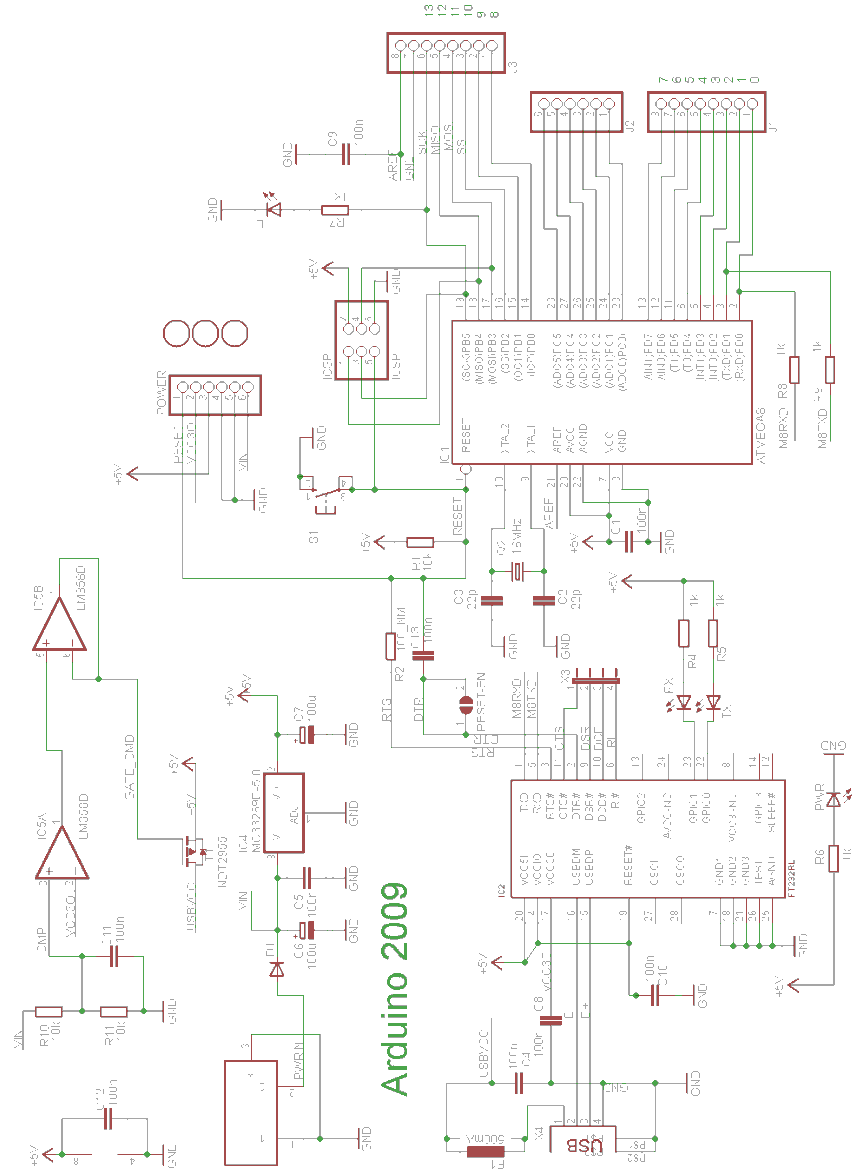
Camera perlu ditambahkan supaya penyelesain alogaritmanya bisa lebih teliti dan efesien.

# DAFTAR PUSTAKA

1. Ahmad, U. (2005). *Pengolahan Citra Digital dan Teknik Pemrogramannya.* Yogyakarta: Graha Ilmu.
2. Bourke, Paul, “Intersection of two Lines” <URL: <http://paulbourke.net/geometry/lineline2d/>>, Agustus, 2011.
3. Castleman, Kenneth R. 1996. Digital Image Processing, Prentice Hall International Inc.
4. Gardner, Sue, “Hough Transform” <URL: <http://en.wikipedia.org/wiki/Hough_transform>>, September, 2011.
5. Kaehler, G. B. (2008). *Learning OpenCV.* Sebastopol: O'Reilly Media, Inc.
6. Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik.* Bandung: Informatika.
7. OpenCV Education Center, “CV Reference Manual“ <URL: <http://www.seas.upenn.edu/~bensapp/opencvdocs/ref/opencvref_cv.htm>>Maret, 2011.
8. Ramadi, S. (2009). *7 Jam Belajar Intreaktif Visual C++ .Net 2005 untuk Orang Awam.* Palembang: Maxikom.
9. Seven Towns “Solution Guide”, London, 2008.
10. Vijn, J, “The Affine Transformation Matrix” <URL: <http://www.coranac.com/tonc/text/affine.htm>>, Maret, 2011.
11. Wahyu, Yohanes, “Rahasia Filter Digital” <URL: <http://www.fotografer.net/isi/artikel/lihat.php?id=381>>, September , 2011.
12. Wikipedia, “Aljabar Linear” <URL: <http://id.wikipedia.org/wiki/Aljabar_linear#Matriks_Balikan_.28Invers.29>>, Agustus, 2011.
13. Yulias, Zervani, “Tutorial Arduino” <URL: http://blog.famosastudio.com/2011/07/tutorial/tutorial-arduino-servo/128>, November, 2011.

# LAMPIRAN

# SCHEMATIC ARDUINO



# LISTING PROGRAM ARDUINO

|  |
| --- |
| int ledPin = 13;  #include <Servo.h>  Servo myservo;  Servo myservo2;  int pos;  void setup(){  Serial.begin(9600);  Serial.setTimeout(5);  myservo.attach(9);  myservo2.attach(10);  myservo.write(30);  myservo2.write(30);  pinMode(ledPin, OUTPUT);  }  void loop(){  if(Serial.available()>0){  String info;  info = Serial.readStringUntil('\n');  if(info.equals("OFF")){  digitalWrite(ledPin, LOW);  //servo  for (pos = 1; pos <= 30; pos += 1) {  myservo.write(pos);  delay(2);  }  }  else if(info.equals("ON")){  digitalWrite(ledPin, HIGH);  //servo  for (pos = 30; pos >= 1; pos -= 1) {  myservo.write(pos);  delay(2);  }  }  if(info.equals("OFF2")){  digitalWrite(ledPin, LOW);  //servo  for (pos = 4; pos <= 30; pos += 1) {  myservo2.write(pos);  delay(2);  }  }  else if(info.equals("ON2")){  digitalWrite(ledPin, HIGH);  //servo  for (pos = 30; pos >= 4; pos -= 1) {  myservo2.write(pos);  delay(2);  }  }  if(info.equals("OFF3")){  digitalWrite(ledPin, LOW);  //servo  for (pos = 2; pos <= 30; pos += 1) {  myservo.write(pos);  myservo2.write(pos);  delay(2);  }  }  }  } |

# LISTING PROGRAM VISUAL CPP

|  |
| --- |
| #include <opencv\cv.hpp>  #include <opencv2\highgui.hpp>  #include <iostream>  #include <stdio.h>  #include <stdlib.h>  #include <string>  //#include <atlstr.h> //serial communication  #include "SerialPort.h"  using namespace std;  using namespace cv;  void find\_moments(Mat result\_src);  void find\_blob(Mat thr, Mat img, Mat &result\_src);  VideoCapture cap(0); // open the default camera  char output[MAX\_DATA\_LENGTH]; //Serial Port  char incomingData[MAX\_DATA\_LENGTH];  char port[] = "\\\\.\\COM19";  int cekpoint200 = 0;  int cekpoint100 = 0;  int zeropoint200 = 0;  int zeropoint100 = 0;  int zeropoint200status = 0;  int zeropoint100status = 0;  int zeronocoin = 0;  bool RESET = FALSE;  bool RESET2 = FALSE;  SerialPort arduino(port);  int main(){  Mat thr,dst,gry;  char c;  //SerialPort arduino(port);  if (arduino.isConnected()) {  cout << "Connection Arduino Terhubung" << endl << endl;  }  else {  cout << "Error in port name" << endl << endl;  }  if (!cap.isOpened()) // check if we succeeded  return -1;  for (;;)  {  //system("CLS");  Mat frame;  Mat result\_blob;  cap >> frame; // get a new frame from camera  cap.set(CV\_CAP\_PROP\_FRAME\_WIDTH, 320);  cap.set(CV\_CAP\_PROP\_FRAME\_HEIGHT, 240);  Mat img(frame.rows, frame.cols, CV\_8UC1, Scalar(255)); // make a white image  cvtColor(frame, gry, cv::COLOR\_RGB2GRAY);  blur(gry,gry,Size(3,3));  threshold(gry, thr, 127, 200, THRESH\_BINARY\_INV);    find\_blob(thr , img, result\_blob);  find\_moments(result\_blob);  //imshow("thresh\_bininvert", thr);  //imshow("Camera", frame);  waitKey(15);  }  return 0;  }  void find\_blob(Mat thr, Mat img, Mat &im\_with\_keypoints)  {  // Setup SimpleBlobDetector parameters.  SimpleBlobDetector::Params params;  // Change thresholds  params.minThreshold = 10;  params.maxThreshold = 200;  // Filter by Area.  params.filterByArea = true;  params.minArea = 900;  // Filter by Circularity  params.filterByCircularity = true;  params.minCircularity = 0.85;  // Filter by Convexity  params.filterByConvexity = true;  params.minConvexity = 0.1;  // Filter by Inertia  params.filterByInertia = true;  params.minInertiaRatio = 0.3;  // Storage for blobs  vector<KeyPoint> keypoints;  // Set up detector with params  Ptr<SimpleBlobDetector> detector = SimpleBlobDetector::create(params);  // Detect blobs  detector->detect(thr, keypoints);  drawKeypoints(img, keypoints, im\_with\_keypoints, Scalar(0), DrawMatchesFlags::DRAW\_RICH\_KEYPOINTS);  //erode(im\_with\_keypoints, im\_with\_keypoints, Mat(), Point(-1, -1), 1, 1, 1);  //dilate(im\_with\_keypoints, im\_with\_keypoints, Mat(), Point(-1, -1), 1, 1, 1);    // Show blobs  //imshow("filterblob", im\_with\_keypoints);  }  void find\_moments(Mat thr)  {  Mat frame2;  cap >> frame2;  Mat canny\_output;  Mat thr\_scd,thr\_gry;  div\_t m\_total;  vector<vector<Point>> contour;  vector<Vec4i> hierarchy;    //Init font  String text = "500";  String text2 = "200";  String text3 = "100";  int fontFace = FONT\_HERSHEY\_SCRIPT\_SIMPLEX;  double fontScale = 0.5;  int thickness = 1.8;  //detect using canny filter  cvtColor(thr, thr, CV\_BGR2GRAY);  blur(thr, thr, Size(3, 3));  Canny(thr, canny\_output, 1, 2, 3);  //imshow("canny", canny\_output); //show canny image    //find contour  findContours(canny\_output, contour, hierarchy, RETR\_TREE, CHAIN\_APPROX\_SIMPLE,Point(0, 0));  //Get the moment  vector<Moments> mu(contour.size());  for (int i = 0; i < contour.size(); i++)  {  mu[i] = moments(contour[i], false);  }  //Drawing contour  Mat drawing(canny\_output.size(), CV\_8UC3, Scalar(255, 255, 255));  Scalar color = Scalar(0, 0, 255);  circle(frame2, Point2f(28, 65), 3, Scalar(0, 255, 0), -1, 7, 0);  circle(frame2, Point2f(26, 138), 3, Scalar(0, 255, 0), -1, 7, 0);  circle(frame2, Point2f(284, 60), 3, Scalar(255, 0, 0), -1, 7, 0);  circle(frame2, Point2f(282, 148), 3, Scalar(255, 0, 0), -1, 7, 0);  // Get the centroid of figures.  vector<Point2f> mc(contour.size());  vector<Point2f> mk(contour.size());    int iterasi = 0;  int savpoint[100];  int jaraktot;  for (int i = 0; i < contour.size(); i++)  {  int k;  int j = contour.size() % 4;  //cout << j << endl;  if (j == 0) {  //cout << i << endl;  mc[i] = Point2f(mu[i].m10 / mu[i].m00, mu[i].m01 / mu[i].m00);  //cout << "data = " << mc[i].x << "," << mc[i].y << endl;  k = i + 1;  if (k % 4 == 1 || k % 4 == 0) {  double jarak1 = sqrt(pow((contour[i][1].x - mc[i].x), 2) + pow((contour[i][1].y - mc[i].y), 2));  double jarak2 = sqrt(pow((contour[i][2].x - mc[i].x), 2) + pow((contour[i][2].y - mc[i].y), 2));  double jarak3 = sqrt(pow((contour[i][3].x - mc[i].x), 2) + pow((contour[i][3].y - mc[i].y), 2));  double jarak4 = sqrt(pow((contour[i][4].x - mc[i].x), 2) + pow((contour[i][4].y - mc[i].y), 2));  double jarak5 = sqrt(pow((contour[i][5].x - mc[i].x), 2) + pow((contour[i][5].y - mc[i].y), 2));  //double jarak6 = sqrt(pow((contour[i][6].x - mc[i].x), 2) + pow((contour[i][6].y - mc[i].y), 2));  //double jarak7 = sqrt(pow((contour[i][7].x - mc[i].x), 2) + pow((contour[i][7].y - mc[i].y), 2));  //double jarak8 = sqrt(pow((contour[i][8].x - mc[i].x), 2) + pow((contour[i][8].y - mc[i].y), 2));  //double jarak9 = sqrt(pow((contour[i][9].x - mc[i].x), 2) + pow((contour[i][9].y - mc[i].y), 2));  //double jarak10 = sqrt(pow((contour[i][10].x - mc[i].x), 2) + pow((contour[i][10].y - mc[i].y), 2));  //cout << "k = " << k << endl;  jaraktot = (jarak1 + jarak2 + jarak3 + jarak4 + jarak5 ) / 5;  //cout << "jarak = " << jaraktot << endl;  //cout << "i =" << i << endl;  cout << "iterasi =" << iterasi << endl;  savpoint[iterasi] = jaraktot;  mk[iterasi] = mc[i];    iterasi++;    }    }  }  for (int i = 0; i < iterasi; i++) {  cout << "D = " << savpoint[i] << endl;  //cout << i << endl;  zeronocoin = 0;  circle(frame2, mk[i], 3, color, -1, 7, 0);  //cout << mk[i].x << endl;  //cekpoint200 = 0;  int cek = 0;  RESET = FALSE;  RESET2 = FALSE;  if (i % 2 == 0) {  if (mk[i].x <= 300 && mk[i].x >= 180) {  //coin 200  if ((savpoint[i] == 21 && savpoint[i + 1] == 19) ||  (savpoint[i] == 21 && savpoint[i + 1] == 18) ||  (savpoint[i] == 21 && savpoint[i + 1] == 17) ||  (savpoint[i] == 22 && savpoint[i + 1] == 19) ||  (savpoint[i] == 22 && savpoint[i + 1] == 18)) {  putText(frame2, text2, mk[i], fontFace, fontScale, Scalar::all(0), thickness, 2);  cekpoint200++;  }  else {  zeropoint200++;  zeropoint200status++;  RESET = TRUE;  }  }    if (mk[i].x <= 300 && mk[i].x >= 180) {  //coin 100  if ((savpoint[i] == 20 && savpoint[i + 1] == 18) ||  (savpoint[i] == 20 && savpoint[i + 1] == 17) ||  (savpoint[i] == 19 && savpoint[i + 1] == 18) ||  (savpoint[i] == 19 && savpoint[i + 1] == 17) ||  (savpoint[i] == 19 && savpoint[i + 1] == 16)) {  putText(frame2, text3, mk[i], fontFace, fontScale, Scalar::all(0), thickness, 2);  cekpoint100++;  }  else {  zeropoint100++;  zeropoint100status++;  RESET2 = TRUE;  }  }  if (mk[i].x < 180 && mk[i].x >= 110) {  //coin 100  if ((savpoint[i] == 20 && savpoint[i + 1] == 18) ||  (savpoint[i] == 21 && savpoint[i + 1] == 19) ||  (savpoint[i] == 21 && savpoint[i + 1] == 17) ||  (savpoint[i] == 21 && savpoint[i + 1] == 18) ||  (savpoint[i] == 20 && savpoint[i + 1] == 17) ||  (savpoint[i] == 19 && savpoint[i + 1] == 18) ||  (savpoint[i] == 19 && savpoint[i + 1] == 17) ||  (savpoint[i] == 19 && savpoint[i + 1] == 16)) {  putText(frame2, text3, mk[i], fontFace, fontScale, Scalar::all(0), thickness, 2);  cekpoint100++;  }  else {  zeropoint100++;  zeropoint100status++;  RESET2 = TRUE;  }  }  //coin 500  if ((savpoint[i] >= 23 && savpoint[i + 1] >= 20)) {  putText(frame2, text, mk[i], fontFace, fontScale, Scalar::all(0), thickness, 2);  }  cout << "STATUS MOTOR1 = " << RESET << endl;  cout << "STATUS MOTOR2 = " << RESET2 << endl;  //Alogaritma segmentasi coin  if (zeropoint200 >= 12) { //RP200  zeropoint200 = 0;  cekpoint200 = 0;  }  if (cekpoint200 > 0 && RESET == FALSE) zeropoint200 = 0;  //cout << "cekpoint200 =" << cekpoint200 << endl;  //cout << "zeropoint200 =" << zeropoint200 << endl;  if (zeropoint100 >= 12) { //RP100  zeropoint100 = 0;  cekpoint100 = 0;  }  if (cekpoint100 > 0 && RESET2 == FALSE) zeropoint100 = 0;  //cout << "cekpoint100 =" << cekpoint100 << endl;  //cout << "zeropoint100 =" << zeropoint100 << endl;  //DATA SERIAL Rp200  if (cekpoint200 % 2 == 0 || zeropoint200status % 2 == 0) {  if (cekpoint200 == 6){  cout << "KIRIM DATA ON1" << endl;  string data;  data = "ON";  char \*charArray = new char[data.size() + 1];  copy(data.begin(), data.end(), charArray);  charArray[data.size()] = '\n';  arduino.writeSerialPort(charArray, MAX\_DATA\_LENGTH);  delete[] charArray;  zeropoint200status = 0;  }  if (zeropoint200status == 12) {  cout << "KIRIM DATA OFF" << endl;  string data;  data = "OFF";  char \*charArray = new char[data.size() + 1];  copy(data.begin(), data.end(), charArray);  charArray[data.size()] = '\n';  arduino.writeSerialPort(charArray, MAX\_DATA\_LENGTH);  delete[] charArray;  }  }  //DATA SERIAL Rp100  if (cekpoint100 % 2 == 0 || zeropoint100status % 2 == 0) {  if (cekpoint100 == 2) {  cout << "KIRIM DATA ON2" << endl;  string data;  data = "ON2";  char \*charArray = new char[data.size() + 1];  copy(data.begin(), data.end(), charArray);  charArray[data.size()] = '\n';  arduino.writeSerialPort(charArray, MAX\_DATA\_LENGTH);  delete[] charArray;  zeropoint100status = 0;  }  if (zeropoint100status == 12) {  cout << "KIRIM DATA OFF2" << endl;  string data;  data = "OFF2";  char \*charArray = new char[data.size() + 1];  copy(data.begin(), data.end(), charArray);  charArray[data.size()] = '\n';  arduino.writeSerialPort(charArray, MAX\_DATA\_LENGTH);  delete[] charArray;  }  }  }  //cout << "zeropoint200status = " << zeropoint200status << endl;  }  //CEK STATUS COIN STANBY  if (contour.size() / 4 == 0 ) {  zeronocoin++;  if (zeronocoin == 12) {  //cout << "SEMUA DATA OFF" << endl;  cekpoint200 = 0;  cekpoint100 = 0;  cout << "OFF3" << endl;  string data;  data = "OFF3";  char \*charArray = new char[data.size() + 1];  copy(data.begin(), data.end(), charArray);  charArray[data.size()] = '\n';  arduino.writeSerialPort(charArray, MAX\_DATA\_LENGTH);  delete[] charArray;  }  }  //cout << "status zero coin = " << contour.size() / 4 << endl;  //Calculate total coin  //m\_total = div(contour.size(), 4);  //if (m\_total.rem <= 4) cout << "total coin = " << m\_total.quot << endl;  //cout << "total coin = " << m\_total.quot << endl;    imshow("results", frame2);    } |