PROTOTYPE DAN IMPLEMENTASI PENYORTIR TELUR DENGAN LOGIKA FUZZY PADA MANIPULATOR 6-*Degree Of Freedom*(DOF)

PROYEK AKHIR

Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Ahli Madya

OLEH:
HAIKAL FIRDIAWAN ZAKY
NIM. 14507134013

JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2017

LEMBAR PERSETUJUAN
PROYEK AKHIR

PROTOTIPE DAN IMPLEMENTASI PENYORTIR TELUR DENGAN LOGIKA FUZZY PADA MANIPULATOR 6-*Degree Of Freedom*(DOF)

Oleh

Haikal Firdiawan Zaky

14507134013

Telah diperiksa dan disetujui Pembimbing

untuk diuji

Yogyakarta, 05 Juli 2017

Mengetahui,
Kaprodi Teknik Elektronika

Drs. Sri Waluyanti, M.Pd
NIP.19581218 198603 2 001

Menyetujui,
Dosen Pembimbing

Dr. Fatchul Arifin, M.T.
NIP. 19720508 198802 1 002

LEMBAR PENGESAHAN

PROYEK AKHIR

PROTOTYPE DAN IMPLEMENTASI PENYORTIR TELUR DENGAN LOGIKA FUZZY PADA MANIPULATOR 6-DEGREE OF FREEDOM(DOF)

Dipersiapkan dan Disusun oleh :

HAIKAL FIRDIAWAN ZAKY

14507134013

Telah Dipertahankan di depan Dewan Penguji Proyek Akhir
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA
Pada tanggal 05 Juli 2017
Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar

Ahli Madya Teknik

Susunan Dewan Penguji

Jabatan	Nama Lengkap Penguji	Tanda Tangan	Tanggal
Ketua Penguji	Dr. Fatchul Arifin	.....	18 Agustus 2017
Sekretaris Penguji	Dessy Irmawati, M.T	.....	21 Agustus 2017
Penguji Utama	Adi Dewanto, M.Kom	.....	28 Agustus 2017

Yogyakarta, 05 Juli 2017
Dekan Fakultas Teknik UNY

Dr. Widarto M.Pd.
NIP. 19631230 198812 1 001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama	: Haikal Firdiawan Zaky
NIM	: 14507134013
Program Studi	: Teknik Elektronika-D3
Judul Proyek Akhir	: Prototype dan Implementasi Penyortir Telur dengan Logika Fuzzy pada Manipulator 6-Degree of Freedom(DOF)

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lain, kecuali bagian-bagian tertentu saya ambil sebagai acuan dengan mengikuti kaidah penulisan karya ilmiah yang benar. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 05 Juli 2017

Yang Menyatakan,

Haikal Firdiawan Zaky

NIM. 14507134013

PROTOTYPE DAN IMPLEMENTASI PENYORTIR TELUR DENGAN LOGIKA FUZZY PADA MANIPULATOR 6-*Degree of Freedom*(DOF)

Oleh : Haikal Firdiawan Zaky

NIM : 14507134013

ABSTRAK

Pengkategorikan ukuran telur secara manual kurang efektif karena bersifat *judgement expert* yang mana tiap perorangan memiliki perbedaan dalam justifikasinya. Untuk mengatasi masalah dalam mengkategorikan ukuran telur, Prototype dan Implementasi Penyortir Telur Dengan Menggunakan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom* (DOF) merupakan salah satu solusi untuk dapat mengatasi permasalahan mengenai perekaan ulang ukuran telur dengan proses sortasinya.

Proyek akhir ini dibuat berbasis pengolahan citra yang diintegrasikan dengan sistem cerdas logika fuzzy dengan metode mamdani dengan operator AND dalam fuzzyfikasinya, dan metode *center of area* untuk menetukan defuzzifikasinya. Ukuran tersebut diklasifikasikan kedalam 5 jenis ukuran telur yakni sangat kecil, kecil, sedang, besar dan sangat besar. Minimum sistem berbasis ATmega 2560 digunakan sebagai kontroler utama dari manipulator robot yang berbasis 6-*Degree of Freedom(DOF)* sebagai media penyortir telur yang akan meletakkan telur kedalam tatakan telur secara berurutan sesuai dengan klasifikasi ukuran telur.

Berdasarkan hasil pengujian pada tiap klasifikasinya dapat disimpulkan bahwa sistem pengolahan citra dengan logika fuzzy memiliki persentase rata-rata keberhasilan 84%. Sedangkan untuk penempatan telur dengan robot mengacu pada sistem Logika Fuzzy sebesar 92%, penempatan telur dengan robot mengacu pada *judgement expert* memiliki keberhasilan sebesar 76%.

Kata Kunci : Manipulator Robot 6-DOF, Pengolahan citra, Logika Fuzzy

MOTTO

"*Bersujud dan mengakui semua kelemahan adalah bagian*

dari usaha untuk mencapai keberhasilan"

"Berpikir tanpa bertindak adalah kosong, dan

bertindak tanpa berfikir tidak lebih dari kosong pula"

"Berapa banyak orang yang berusaha lalu gagal itu karena mereka tidak bekerja dengan efektif dengan hanya mengandalkan fisik.

Letakkan semua masalahmu, duduklah dan berisitrahatlah lalu berjanjilah, mulai gunakan fikiranmu untuk mengendalikan fisikmu"

PERSEMBAHAN

Proyek akhir ini ku persembahkan kepada :

Orang Tua atas segala doa, dukungan dan segala keikhlasan

yang selalu menyertai saya

Adik-adik yang selalu menghibur disaat jenuh

Teman Hidup saya yang selalu setia menemani dan memberi semangat

untuk pengerjaan proyek akhir

Tim Mastro Evo 2015 – 2016 yang menopang jerih payah dalam perjuangan untuk memajukan robotika UNY

Teman-teman Kelas B Teknik Elektronika 2014 atas kebersamaanya yang telah dilewati bersama

Seluruh Dosen yang dengan ikhlas selalu membimbing dan mendidik saya

Semua pihak yang telah membantu kelancaran pengerjaan proyek akhir

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga dapat terselesaikannya Proyek Akhir dengan judul

"Prototype dan Implementasi Penyortir Telur Dengan Logika Fuzzy Pada Manipulator 6-*Degree of Freedom*(DOF)"

Proyek akhir ini dapat terselesaikan tidak lepas dari berbagai pihak yang telah memberikan bantuan guna menunjang pengerjaan proyek akhir ini. Oleh karena itu, pada kesempatan ini saya ingin menyampaikan ucapan banyak terima kasih kepada:

1. Bapak Dr. Fatchul Arifin selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta dan juga selaku Dosen Pembimbing Proyek Akhir yang telah memberikan bimbingan dan arahan untuk penyelesaian proyek akhir ini.
2. Bapak Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
3. Ibu Dr. Sri Waluyanti, M.Pd. selaku Ketua Program Studi Teknik Elektronika D3, Fakultas Teknik, Universitas Negeri Yogyakarta.
4. Seluruh Dosen Pengajar Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada saya.
5. Teman-teman Teknik Elektronika kelas B 2014 yang telah memberikan bantuan sehingga pembuatan proyek akhir ini dapat terselesaikan.
6. Tim KRAI-Maestro Evo UNY yang telah berjuang untuk pengabdian dan menjunjung nama Universitas dalam ajang Kontes Robot Indonesia 2015-2016.
7. Kedua Orang tua dan Adik-adik yang selalu mendukung dan mendoakan dengan keihklasannya untuk kelangsungan penyelesaian proyek akhir ini.

8. Semua pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu pengerjaan proyek akhir ini.

Saya menyadari bahwa laporan ini masih memiliki bayak kelemahan dan jauh dari kesempurnaan. Oleh karena itu saran dan kritik akan senantiasa saya harapkan dengan maksud untuk penyempurnaan pengerjaan proyek akhir ini.

Akhir kata, Semoga proyek akhir ini dapat bermanfaat untuk kelangsungan proses belajar bagi mahasiswa Universitas Negeri Yogyakarta.

Yogyakarta, 05 Juli 2017

Haikal Firdiawan Zaky

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Perkembangan teknologi industri sejauh ini telah mengalami peningkatan dalam mutu kerja dan efektifitas untuk mencapai hasil yang optimal. Hasil optimal tersebut dilatarbelakangi atas permintaan investor industri untuk melakukan pembenahan setiap kinerja mesin baik itu kinerja mesin *sorter*, *labelling*, *production*, *palletizing* maupun *packaging*. Kelima jenis mesin yang dibedakan berdasarkan cara kerja mesin tersebut ditarget mampu melakukan tugasnya sebagai alat bantu manusia dengan efektif dan optimal untuk mengurangi biaya produksi dari suatu industri tanpa mengurangi kualitas hasil produksi itu sendiri. Ketika proses produksi berlangsung kelima jenis mesin tersebut akan lebih ditekankan unjuk kerja dari masing-masing jenis mesin tersebut salah satunya adalah mesin penyortir. Mesin penyortir merupakan alat bantu untuk memilah hasil produksi utama dengan hasil produksi pengotor(cacat produksi).

Industri memanfaatkan jenis mesin penyortir untuk dapat memilah hasil produksinya dengan optimal berdasarkan ciri fisik dan karakteristik hasil produksi yang dinilai bagus sesuai dengan standar produksi industri dan dapat didistribusikan kepada konsumen secara langsung untuk memenuhi kebutuhan sehari-hari. Salah satu pemanfaatan mesin sortasi pada industri adalah untuk mensortir telur, dimana telur merupakan kebutuhan pokok makanan dari rumah tangga, restoran kuliner hingga industri-industri makanan seperti biskuit dan lain sebagainya. Kebutuhan

industri sortir telur sebagai distributor utama harus menyanggupi untuk memenuhi kebutuhan telur tersebut baik dari kualitas, ukuran, dan berat telur.

Berdasarkan aturan yang sesuai dengan Standar Nasional Indonesia SNI No. 3926:2008, dimana untuk bobot telur sendiri dapat dibedakan menjadi 3 kelompok yaitu kecil dengan indeks parameter berat kurang dari 50 g, indeks parameter berat golongan sedang 50 g sampai 60 g dan besar dengan indeks parameter berat lebih dari 60 g. Persyaratan mutu yang diatur mencakup tingkatan mutu fisik, meliputi kondisi kerabang, kondisi kantung udara, kondisi putih telur, kondisi kuning telur, dan Bau. Lalu bagaimanakah industri dapat mengategorikan kualitas telur yang akan dipasarkan? Mengutip dari sebuah artikel laman web (http://kulinologi.co.id/acrobat/index1.php?view&id=900 (2016, 10.15) dimana didalam artikel tersebut dipaparkan tentang cara industri mengategorikan kualitas sebuah telur, diantaranya :

1) Kualitas AA (Mutu 1)

Kondisi telur bersih, halus, licin, tidak retak, dan bentuknya normal. Kedalaman kantung udara tidak boleh lebih dari 3,2 mm (SNI : < 0,5 cm). Putih telur harus bersih, kental dan stabil, dengan konsistensi seperti gelatin, Ketika diteropong, kuning telur tidak bergerak-gerak, berbentuk bulat, terletak deitengah telur, kuning telur dan bersih dari bercak darah atau noda apapun. Bayangan batas-batas kuning dan putih telur ketika di teropong tidak terlihat jelas.

2) Kualitas A (Mutu 2)

Cangkang telur bersih, halus, licin, tidak retak, dan bentuknya normal. Kedalaman rongga udara tidak boleh lebih dari 4,8 mm (SNI : 0,5-0,9 cm). Putih telur harus bersih, dan kental. Bayangan batas-batas kuning dan putih telur ketika diteropong mulai terlihat agak jelas. Kuning telur berbentuk bulat, posisinya di tengah, harus bersih, dan tidak ada bercak atau noda.

3) Kualitas B (Mutu 3)

Cangkang bersih, tidak boleh retak, agak kasar, dan mungkin bentuknya abnormal. Kantung udara lebih dari 1,6 mm
(SNI : > 1 cm). Putih telur encer, sehingga kuning telur bebas bergerak saat diteropong. Ada noda sedikit, tetapi tidak boleh ada benda asing lainnya dan bagian kuning belum tercampur dengan putih. Kuning telur terlihat gepeng (pipih)

bentuknya, agak melebar, bintik atau noda darah mungkin ada, tetapi diameternya tidak boleh lebih dari 3,2 mm.

Dari paparan tersebut terlihat bahwa dalam pengkategorian telur tersebut telah melewatkan hal yang paling penting dari sebuah telur, yakni ukuran dari telur tersebut. Mengukur ukuran sebuah telur dapat mempercepat proses sortasi, karena ukuran daripada sebuah telur dapat memprediksi berat dari telur tersebut, dari berat telur tersebut dapat dikategorikan ulang kualitas mutu telur untuk mengestimasi kandungan gizi dari tiap ukuran telur, kita ambil contoh misalkan diketahui kualitas telur dikategorikan kualitas AA, dan dapat diketahui bahwa ukuran besar dan kecil disejajarkan (diabaikan), bagaimana jika ukuran kecil dari jenis sebuah telur yang sama tersebut ternyata memiliki kandungan gizi yang jauh lebih banyak dan lebih baik daripada ukuran besar kualitas AA dari telur tersebut? Tentu pada pengaplikasiannya perekaan ulang dalam mengategorikan telur harus dibenahi pada sistemnya secara utuh agar hasil dari pemasaran dapat meningkatkan omzet dari perusahan sortasi telur. Disisi lain sortasi dari beberapa perusahaan masih menggunakan sistem manual ataupun semi-otomatis untuk memilah telur yang artinya masih ada campur tangan manusia dalam melakukan penyortiran telur tersebut. Jika diperhatikan ulang mengenai aturan Standar Nasional Indonesia SNI No. 3926:2008, dimana kesterillan dari fisik dan bentuk fisik telur juga harus diperhatikan, maka bukan tidak mungkin ketika sortasi dilakukan secara manual ataupun semi automatis, terdapat kesalahan seperti *human error* yang mengakibatkan keretakan tak disadari pada cangkang telur sehingga bakteri luar

mudah untuk masuk dan berkembang didalam telur yang mengakibatkan pembusukan telur secara berkala.

Oleh sebab itu proyek akhir "Prototype dan Implementasi Sortasi Telur Dengan Menggunakan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom* (DOF)" ini dibuat dengan tujuan untuk dapat mereka ulang pengkategorian kualitas dan didesain mampu untuk menjamin mutu telur melalui sortasi ukuran fisik telur yang diaplikasikan dengan manipulator sebagai media pensortir dan *opencv* sebagai *library* untuk pengolahan citra digital untuk mendeteksi ukuran dan fisik tiap telur, sehingga kerugian-kerugian kecil dapat segera diminimalisir agar tidak mengakibatkan kerugian yang lebih besar jika tidak segera dibenahi pada sistem sortasi tersebut.

B. IDENTIFIKASI MASALAH

Berdasarkan paparan dari latar belakang masalah di atas, dapat diidentifikasi masalah sebagai berikut :

1. Penyortiran pada industri penyortir telur masih bersifat manual untuk menentukan mutu, kualitas dan ukuran telur yang bersifat *judgement expert*.
2. Belum adanya media penyortiran telur secara otomatis dengan manipulator robot 6-*Degree of freedom*.
3. Belum adanya alat penyortir telur yang berbasis sistem cerdas untuk industri penyortir telur secara automatis dan *real time* agar lebih efektif dalam melakukan proses penyortiran telur.

C. BATASAN MASALAH

Berdasarkan identifikasi masalah yang muncul, maka diperlukan adanya pembatasan masalah untuk mempersempit ruang lingkup dalam pengerjaan proyek akhir ini, sehingga nantinya akan terfokus pada inti dari permasalahan. Pembatasan ini didasarkan pada keterbatasan penulis untuk meminimalisir pembiasan bidang ilmu pengetahuan yang belum diketahui oleh penulis. Adapun batasan dari masalah tersebut adalah sebagai berikut :

1. Dalam mengkategorikan ulang antara kualitas dan mutu telur, pengolahaan citra difokuskan hanya berdasarkan bentuk ukuran fisik telur.
2. Rancang bangun dari penyortir telur dengan manipulator 6-DOF(*Degree Of Freedom*) dalam bentuk *prototype*.
3. Rancang bangun penyortir telur berbasis sistem cerdas menggunakan logika fuzzy dengan metode mamdani serta mengunakan opencv sebagai *library* pengolahan citra dalam ekstraksi ciri fisik telur.

D. RUMUSAN MASALAH

Berdasarkan latar belakang, identifikasi masalah dan batasan masalah, maka rumusan masalah yang akan dibahas dalam mengerjakan proyek akhir ini dapat diidentifikasikan sebagai berikut :

1. Bagaimana merancang dan bangun serta mengimplementasikan penyortir telur dengan manipulator *6-DOF(Degree Of Freedom*) menggunakan logika fuzzy ?

2. Bagaimana merancang dan bangun logika fuzzy dengan menggunakan opencv sebagai library untuk pengolahan citra ?
3. Bagaimana unjuk kerja dari manipulator 6-DOF*(Degree Of Freedom*) beserta opencv sebagai penyortir telur berbasis logika *fuzzy* ?

E. TUJUAN

Dengan merujuk pada perumusun masalah yang telah dijabarkan, maka di harapkan dapat mencapai tujuan sebagai berikut :

1. Rancang bangun serta mengimplementasikan penyortir telur dengan manipulator robot *6-DOF(Degree Of Freedom*).
2. Rancang bangun logika fuzzy dengan menggunakan opencv sebagai library untuk pengolahan citra sebagai aplikasi dari penyortir telur dengan manipulator robot *6-DOF(Degree Of Freedom*).
3. Mengetahui unjuk kerja dari manipulator 6-DOF(*Degree Of Freedom*) sebagai media penyortir beserta opencv sebagai sistem pengolah citra untuk sortasi telur berbasis logika *fuzzy*.

F. MANFAAT

Proyek akhir dengan judul “Prototype dan Implementasi Sortir Telur dengan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom* (DOF)” diharapkan dapat bermanfaat :

1. Bagi Mahasiswa
    a. Sebagai sumber rujukan untuk pengembangan ilmu tentang logika fuzzy.
    b. Sebagai sarana untuk mengimplementasikan logika fuzzy sebagai penyortir telur.
    c. Sebagai sarana untuk rancang bangun manipulator 6-*Degree Of Freedom* (DOF).

2. Bagi Perguruan Tinggi
    a. Sebagai media pembelajaran tentang otomasi industri.
    b. Sebagai referensi tambahan untuk mengaplikasikan ilmu terapan logika fuzzy.

3. Bagi Masyarakat, Peneliti dan Industri

    a. Mempermudah proses sortasi telur pada industri penyortir telur.
    b. Meningkatkan kredibilitas produksi dari industri penyortir telur.
    c. Sebagai sarana rujukan penelitian di bidang otamasi industri dengan sistem kendali cerdas.

G. KEASLIAN GAGASAN

Proyek akhir dengan judul "Prototype dan Implementasi sortir telur dengan Logika Fuzzy pada Manipulator *6-DOF(Degree Of Freedom*" merupakan proyek

akhir yang didesain dengan maksud dan tujuan sebagai terobosan baru untuk pengaplikasian penyortir telur berbasis pengolahaan citra yang memanfaatkan library dari openCV. Adapun karya yang digunakan sebagai rujukan pembuatan penyortir telur adalah *Desain Kerja Mesin Grading Telur Ayam* yang di kerjakan oleh Feby Nopriandi sebagai Thesis tahun 2015 dari program studi teknik mesin pertanian dan pangan, pascasarjana Institut Pertanian Bogor. Implementasi Desain Kerja Mesin *Grading* Telur Ayam yang di kerjakan oleh Feby Nopriandi berfokus pada penelitian mengenai rancang bangun dari sebuah mesin *Grading* telur. Penelitian tersebut dibuat secara kompleks dan bukan sebagai *prototype* penyortir telur dimana penyortir telur yang dikerjakan Feby Nopriandi tersebut memiliki sensor *load cell* sebagai media proses *grading* untuk penyortiran telur, kamera sebagai sensor untuk menentukan orientasi dari telur, konveyor pengumpan sebagai transportasi telur untuk masuk proses *grading* telur, konveyor pengarah untuk memastikan orientasi dari telur dengan tepat, konveyor *grading* sebagai konveyor utama untuk melakukan sortasi berdasarkan berat menggunakan sensor *load cell* dan konveyor keluar sebagai wadah hasil dari *grading* telur tersebut.

Adapun perbedaan Alat yang dibuat pada proyek akhir ini dengan thesis yang dibuat oleh Feby Nopriandi adalah :

1. Penyortiran dilakukan dengan Manipulator *6-DOF(Degree Of Freedom*.
2. Penyortiran telur berfokus pada ukuran telur dengan sistem pengambilan keputusan klasifikasi ukuran telur menggunakan logika fuzzy.
3. Media Penyortiran telur memanfaatkan pengolahan citra dengan metode Algoritma Codebook yang diintegrasikan dengan library openCV.

Pengolahan citra untuk identifikasi telur yang dikerjakan oleh Syahrul Awalludin Sidiq dan Dessy Irmawati dalam jurnal penelitian Fakultas Teknik Universitas Negeri Yogyakarta "Electronics, Informatics and Vocational Education(ELINVO),Volume 1, Nomor 3, November 2016". Untuk jurnal yang dikerjakan oleh Syahrul Awalludin Sidiq dan Dessy Irmawati, menggunakan software GUI Fuzzy dari matlab sebagai Fuzzyfikasi dan pengolah citra untuk diidentifikasi jenis ukuran telur apakah termasuk sangat kecil,kecil, sedang, besar atau sangat besar. Untuk mengoptimalkan identifikasi jenis ukuran telur sistem cerdas yang digunakan adalah logika fuzzy dengan metode mamdani dengan basis rule menggunakan operator AND dan defuzzyfikasi menggunakan metode Centroid.

Perbedaan proyek akhir ini dengan jurnal penelitian yang dibuat oleh Syahrul Awalludin Sidiq dan Dessy Irmawati adalah :

1. Sistem pengolahan citra menggunakan bahasa pemrogram C pada Microsoft Visual Studio 2013 yang diitengrasikan dengan openCV sebagai library pengolah citra dengan metode Codebook Algorima.
2. Sistem fuzzy yang dibuat tanpa menggunakan GUI Fuzzy dari matlab namun dengan menggunakan bahasa C pada Microsoft Visual Studio 2013.
3. Aplikasi yang digunakan tidak berupa simulasi, melainkan Menggunakan prototype manipulator *6-DOF(Degree Of Freedom*) robot sebagai media penyortir telur.

Berdasarkan paparan dari perbedaan diatas, karya ini benar-benar murni dari rancangan dan gagasan karya yang dibuat oleh penulis sendiri. Adapun jika terdapat persamaan penulis yakin masih terdapat perbedaan mendasar baik berupa rancangan dan teknik pengimplementasian yang diusung.

BAB II

PENDEKATAN PENYELESAIAN MASALAH

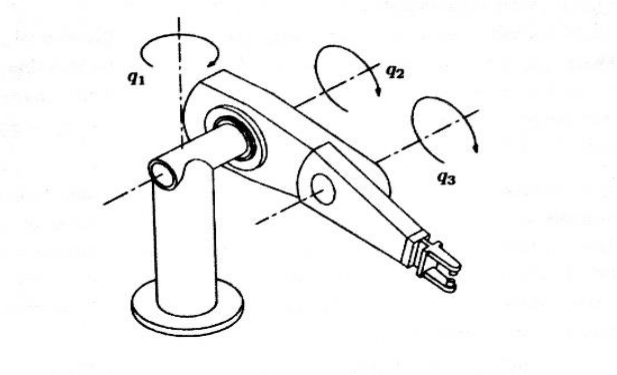
A. Penyortir Telur dengan Manipulator 6-Degree Of Freedom (DOF)

Sorting adalah sebuah metode untuk merangkai dan mengurutkan suatu objek dalam urutan tertentu yang dipisahkan satu persatu berdasarkan ciri fisik masing-masing objek agar memiliki kesamaan antara objek satu dengan objek lain dalam suatu wadah yang dihimpun, oleh karena itu sorting memiliki dua arti umum yang berbeda, yakni:

1) pengurutan: merangkai benda yang sejenis, sekelas, dll, dalam urutan yang teratur,
2) kategorisasi: pengelompokan dan pemberian label kepada benda dengan sifat yang serupa. (id.wikipedia.org/wiki/Sorting).

Manipulator adalah suatu mekanik yang memiliki fungsi sebagai memindah, mengangkat atau memanipulasi benda sehingga benda dapat bergerak(Endra Pitowarno, 2006:16). Umumnya bentuk dari robot manipulator memiliki ciri berupa lengan atau pada umumnya disebut dengan *link* yang saling terhubung membentuk beberapa *joint* yang biasa dikenal dengan *Degree of Freedom*. Robot manipulator idealnya memiliki 6-DOF, dimana 3 bagian DOFnya berfungsi untuk menentukan posisi ujung terakhir dari sebuah lengan dalam ruang cartesian, sedangkan 3 DOF sisa lainnya digunakan sebagai orientasi(Widodo Budiharto, Djoko purwanto, 2015:4-5).

Sebagai contoh perhatikan gambar 1. Variabel q1, q2 dan q3 merupakan akhir dari sebuah lengan yang biasa disebut dengan *joint*. *Joint* tersebut terkoneksi antara satu sama lain sehingga membentuk batas ujung akhir dari lengan robot tersebut yang disebut dengan *Degree of Freedom*



Gambar 1. Robot Manipulator

Sumber: Widodo Budiharto, Djoko Purwanto (2015, 5)

Untuk dapat menggerakan robot manipulator secara leluasa dibutuhkan suatu kinematika dari robot manipulator itu sendiri, kinematika merupakan suatu metode analisis untuk menggerakkan robot manipulator tanpa memperhitungkan *payload* dan *force* penyebab pergerakkan dari sebuah robot. Umumnya beberapa metode terapan kinematika pada robot manipulator dibagi menjadi 2 yaitu:

1. *Forward Kinematics*

Forward Kinematics adalah suatu metode analisis kinematika yang digunakan untuk mendapatkan koordinat posisi cartesian dari sebuah robot manipulator tiap sendi robot, Sebagai contoh jika sudut cartesian dari sebuah robot manipulator telah diketahui sebelumnya, dengan memanfaatkan trigonometri dasar dapat ditentukan

posisi, orientasi dan sudut dari end effector sehingga dapat *forward kinematics* merupakan solusi yang tepat untuk mendapatkan sudut sendi dari robot manipulator. Berikut merupakan contoh implementasi dari *Forward kinematics* dalam ruang grafik cartesian 2 dimensi untuk robot manipulator *3-DOF* :

$$x = l_1 cos\ \theta_1 + l_2 cos\ (\theta_1 + \theta_2) + l_3 cos\ (\theta_1 + \theta_2 + \theta_3) \qquad (1.1)$$

$$y = l_1 cos\ \theta_1 + l_2 cos\ (\theta_1 + \theta_2) + l_3 cos\ (\theta_1 + \theta_2 + \theta_3) \qquad (1.2)$$

$$\vartheta = \theta_1 + \theta_2 + \theta_3 \qquad (1.3)$$

Berdasarkan persamaan (1.1) dan (1.2) tentang trigonometri dasar, dapat ditentukan bahwa *forward kinematic* memanfaatkan tiga persamaan nonlinier dimana persamaan diatas dapat menggambarkan hubungan antara akhir koordinat sudut cartesian secara eksplisit untuk akhir dari *efector/link* dari robot manipulator. Secara matematis Untuk menemukan koordinat *Degree of Freedom* dimana diberikan set akhir koordinat *efector* misal (x , y , φ) dengan mengacu pada persamaan (1.3), untuk menentukan set koordinat *Degree of Freedom* dapat dipecahkan dengan memanfaatkan persamaan nonlinear untuk θ1, θ2 dan θ3 . dengan kinematika dari planar robot manipulator lebih mudah untuk dirumuskan dan diaplikasikan.

2. *Inverse Kinematics*

Analisis atau prosedur yang digunakan untuk menghitung koordinat siku untuk satu set akhir koordinat efektor disebut kinematika terbalik. Pada dasarnya, prosedur ini melibatkan pemecahan set persamaan. Namun persamaan, secara umum, nonlinear dan kompleks. Dan karena itu, Analisis kinematika terbalik dapat menjadi terlibat. Seperti yang disebutkan sebelumnya, bahkan jika mungkin untuk memecahkan persamaan nonlinear. Tidak mungkin ada menjadi set koordinat siku untuk akhir koordinat efektor yang diberikan. Dengan mengacu persamaan (1.4) dan (1.5) persamaan kinematika langsung dapat dijabarkan sebagai berikut:

$$x = d_2 \cos\theta_1 \tag{1.4}$$

$$y = d_2 \cos\theta_1 \tag{1.5}$$

Jika kita membatasi revolute siku untuk memiliki sudut sendi dalam interval [0, 2π), ada dua solusi untuk kinematika invers yang dijabarkan pada persamaan (1.6):

$$d_2 = \sqrt[\sigma]{x^2 + y^2},\ \theta_1 = a\tan 2\left(\frac{y}{d2}, \frac{x}{d1}\right), \sigma = \pm 1 \tag{1.6}$$

Di sini kita telah menggunakan fungsi atan2 untuk menentukan θ1 sudut sendi. Namun, tergantung pada pilihan σ, ada dua solusi untuk d2 dan itu untuk θ1. Analisis kinematika terbalik untuk planar 3-R manipulator dapat diperoleh solusi analitis dengan mempertimbangkan bahwa persamaan kinematika langsung pada persamaan (1.1 – 1.3). Anggap bahwa kita diberi koordinat Cartesian x, y, dan φ

dan kita ingin mencari analitis ekspresi untuk sudut sendi θ1, θ2, dan θ3 dalam hal koordinat Cartesian Mengganti (1.3) ke (1.1) dan (1.2) kita dapat menghilangkan θ3 sehingga kita memiliki dua persamaan di θ1 dan θ2:

$$x - l_3 \cos\phi = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (2)$$

$$x - l_3 \sin\phi = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (3)$$

Dimana yang tidak diketahui telah dikelompokkan di sisi kanan, sisi kiri hanya tergantung pada end effector atau koordinat Cartesian dan karena itu dikenal. Ubah nama sisi kiri, x '= x - cos φ l3, y' = y - sin φ, untuk mempermudah. Lalu hasil dari *invers* tersebut dijabarkan persamaan dalam (2) dan (3), sehingga hasilnya menjadi persamaan (3.1) dan (3.2):

$$(x' - l_1 cos\theta_1)^2 = (l_2 \cos(\theta_1 + \theta_2))^2 \quad (3.1)$$

$$(y' - l_1 cos\theta_1)^2 = (l_2 \cos(\theta_1 + \theta_2))^2 \quad (3.2)$$

Setelah menata ulang istilah kita mendapatkan persamaan nonlinear tunggal dalam θ_1:

$$(-2l_1x')\cos\theta_1 + (-2l_1y')\sin\theta_1 + (x'^2 + y'^2 + l_1^2 + l_2^2) = 0 \quad (4)$$

Untuk memecahkan dua persamaan nonlinear dua variabel yang tidak diketahui seperti pada persamaan (2 dan 3). Dapat disederhanakan lebih lanjut dengan

menyederhanakan persamaan nonlinear tunggal dalam single liner persamaan (4). Hasil dari persamaan (4) adalah persamaan (5) :

$$\mathrm{P}\cos\alpha + Q\sin\alpha + R = 0 \qquad (5)$$

Pada jenis Persamaan (5) ini dapat diselesaikan dengan menggunakan substitusi sederhana. Ada dua solusi untuk θ1 diberikan oleh:

$$\theta_1 = \gamma + \sigma cos^{-1}\left(\frac{-x'^2+y'^2+{l_1}^2-{l_2}^2}{2l_1\sqrt{x'^2+y'^2}}\right) \qquad (6)$$

Dimana,

$$\gamma = \mathrm{a}\tan 2\left(\frac{-y'}{\sqrt{x'^2+y'^2}}, \frac{x'}{\sqrt{x'^2+y'^2}}\right) \qquad (7)$$

Dan,

$$\sigma = \pm 1 \qquad (8)$$

Perhatikan bahwa ada dua solusi untuk θ1, satu sesuai dengan σ = + 1, dan solusi ke 2 sesuai dengan σ = -1. salah satu dari solusi ini mengacu ke Persamaan (4) dan (5) sehingga didapatkan persamaan (8.1) dan (8.2):

$$\cos(\theta_1 + \theta_2) = \frac{x'-l_1\cos\theta_2}{l_2} \qquad (8.1)$$

$$\sin(\theta_1 + \theta_2) = \frac{x' - l_1 \sin\theta_2}{l_2} \qquad (8.2)$$

Hal ini memungkinkan kita untuk memecahkan θ2 menggunakan fungsi atan2 :

$$\theta_2 = \text{atan2}(\frac{y' - l_1 \sin\theta_2}{l_2}, \frac{x' - l_1 \cos\theta_2}{l_2}) - \theta_1 \qquad (9)$$

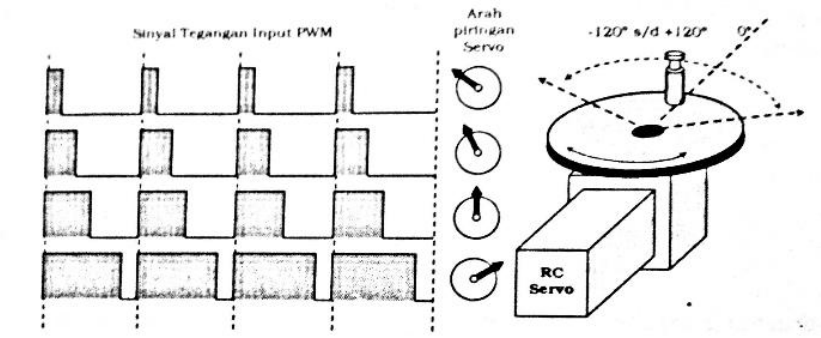
Dengan demikian, untuk setiap solusi untuk θ1, ada satu solusi (unik) untuk θ2. Akhirnya, θ3 dapat dengan mudah ditentukan dari (3c):

$$\theta_3 = \phi - \theta_1 - \theta_2 \qquad (10)$$

Persamaan (6 dan 10) adalah solusi kinematika terbalik untuk 3-R manipulator. Untuk akhir yang diberikan Posisi efektor dan orientasi, ada dua cara yang berbeda untuk mencapai itu, masing-masing sesuai untuk nilai yang berbeda dari σ.

B. SERVO

Servo adalah sebuah perangkat mekanik motor DC permanen yang terintegrasi dengan gabungan beberapa mekanik *gear*. Output keluaran dari servo memiliki torsi, akselarasi dan kecepatan sudut yang relevan untuk aplikasi *controlling* dengan sudut yang diinput dengan kode sinyal berbentuk PWM dengan jangkauan berkisar antara -90 derajat sampai 90 derajat(Endra Pitowarno, 2006: 88). Berikut merupakan ilustrasi hubun gan bentuk sinyal PWM dengan posisi poros servo :



Gambar 2. Prinsip kerja servo

Sumber : Robotika Desain, Kontrol dan Kecerdasan Buatan(2006, 88)

Pada aplikasinya input PWM servo sebagi input dapat diolah dengan memanfaatkan fasilitas timer, timer tersebut digunakan untuk membangkitkan sinyal PWM yang berkesinambungan dengan tegangan sehingga ketika tegangan rendah maka kecepatan motor akan lambat, ketika tegangan tinggi maka putaran motor akan cepat. Teknik kendali servo menggunakan PWM biasa dinyatakan dalam *Duty/Cycle* untuk menetukan lebar pulsa PWM, Duty Cycle terbentuk dari timer mikrokontroller atau perangkat pembangkit timer eksternal, timer pada duty cycle tersebut sangat berperan penting dalam menentukan lebar pulsa dimana jika dimisalkan Duty cyclenya sebesar 10% itu artinya lebar pulsa PWM dari servo tersebut adalah 1/10 bagian dari keseluruhan satu perioda dari PWM servo tersebut, jika duty-cycle 100% maka lebar pulsa akan ekivalen linier dengan tegangan maksimum pada motor dikurangi tegangan saturasi pada kolektor-emitor dari MOSFET internal servo.

C. MOTOR DC

Motor DC merupakan aktuator yang memerlukan suplai tegangan arus searah pada kumparan untuk diubah menjadi gerak mekanik. Kumparan medan pada Motor DC disebut dengan stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar) (any.web.id/pengertian-prinsip-kerja-dan-jenis-jenis-motor-dc).

Motor DC bergerak dengan memanfaatkan gaya lorentz, Pada umumnya motor DC yang sering digunakan adalah jenis motor DC magnet permanen, didalam motor DC tersebut terdapat dua buah magnet permanen yang menyebabkan timbulnya medan magnet di antara kedua magnet tersebut ketika dialiri listrik. Akibat dari medan magnet tersebut rotor berputar berdasarkan jumlah pulsa yang dikeluarkan. Perputaran rotor tersebut dipengaruhi oleh lilitan yang berada ditangah rotor dimana jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan yang terhubung ke area kontak yang disebut dengan komutator. Sikat (brushes) pada motor DC memiliki fungsi sebagai penghubung ke kutub positif dan negatif motor dan memberikan *supply* daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. Kecepatan putar motor DC (N) dirumuskan dengan Persamaan (1) :

$$N = \frac{V_{TM} - I_A R_A}{K_\vartheta} \qquad (1)$$

Keterangan :

V_{TM} : Tegangan terminal
I_A : Arus Jangkar motor
R_A : Hambatan jangkar motor
K : Konstanta motor
Φ : Fluk magnet yang terbentuk pada motor

D. DRIVER MOTOR

Driver motor merupakan bagian yang berfungsi untuk menggerakkan Motor DC dimana perubahan arah motor DC tersebut bergantung dari nilai tegangan yang dimasukkan pada input dari driver itu sendiri. Driver motor berfungsi sebagai piranti yang bertugas untuk menjalankan motor baik mengatur arah putaran motor maupun kecepatan putar motor. Macam-macam driver motor diantaranya sebagai berikut.

a. Driver Kontrol Tegangan

   Dengan driver motor kontrol tegangan menggunakan level tegangan secara langsung untuk mengatur kecepatan dari putaran motor.

b. Driver PWM

   Driver PWM dapat mengatur kecepatan motor dengan memberikan pulsa dengan frekuensi yang tetap ke motor, sedangkan yang digunakan untuk mengatur kecepatan adalah duty cycle dari pulsa yang diberikan.

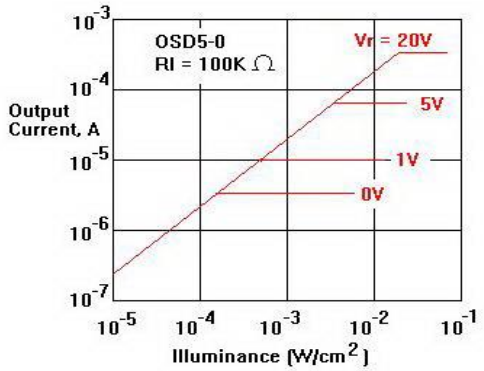
c. Driver H-Bridge

Driver H-Bridge digunakan untuk mengontrol putaran motor yang dapat diatur arah putarannya CW (searah jarum jam) maupun CCW (berlawanan jarum jam). Driver ini pada dasarnya menggunakan 4 buah transistor untuk switching (saklar) dari putaran motor dan secara bergantian untuk membalik polaritas dari motor.

E. SENSOR CAHAYA PHOTODIODA

Sensor cahaya adalah sensor yang memanfaatkan intensitas cahaya sebagai dasar untuk menentukan nilai resistansinya(*http://elektronikadasar.info/sensor-cahaya.htm*). Beberapa komponen yang biasanya digunakan dalam rangkaian sensor cahaya diantaranya Light Dependent Resistor(LDR), Photodiode(fotodioda), dan Photo Transistor (Foto Transistor). Pada dasarnya sensor cahaya merupakan suatu resistor yang mana memiliki nilai resistansi (ohm) yang bergantung pada besaran cahaya yang masuk pada permukaan sensor tersebut. Photodioda akan mengalirkan beberapa arus yang akan membentuk fungsi linear, fungsi linear tersebut merepresentasikan intensitas cahaya yang diterima terhadap arus yang masuk pada kaki photodioda. Arus tersebut bersifat teratur terhadap power density (Dp) yang mana dapat diuraikan bahwa Arus yang dimaksud adalah arus yang keluar ketika photodioda tersebut disinari dengan keadaan dipanjar mundur. Perbandingan antara arus keluaran dengan power density disebut sebagai current responsitivity.

Tanggapan frekuensi sensor photodioda terhadap intensitas cahaya tidak luas, artinya dari rentang tanggapan tersebut sensor photodioda memiliki tanggapan paling baik terhadap cahaya infra merah, tepatnya pada cahaya dengan panjang gelombang sekitar 0,9 µm. Hubungan antara keluaran sensor photodioda dengan intensitas cahaya yang diterimanya ketika dipanjar mundur adalah membentuk suatu fungsi yang linier. Berikut merupakan reprenstasi dari hubungan photodioda dengan intensitas cahaya



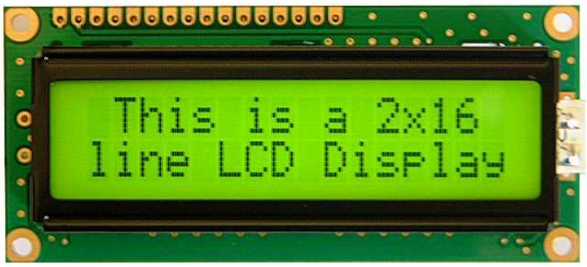
Gambar 3. Kurva Photodioda terhadap cahaya

Sumber : http://elektronika-dasar.web.id/sensor-photodioda

F. LCD (Liquid Crystal Display)

LCD (Liquid Crystal Display) adalah *suatu jenis media penampil yang menggunakan kristal cair sebagai penampil utama*(Aris Munandar, 2012). LCD telah banyak diaplikasikan pada peralatan elektronik seperti misalnya televisi, kalkulator, ataupun layar komputer. Umunya untuk peralatan elektronika pada robotika, LCD yang digunakan berupa LCD dot matrik dengan jumlah karakter sebanyak 16 x 2 (Baris x Kolom) dan salah satu jenis LCD yang sering digunakan adalah seri LM016L. LM016L merupakan modul LCD dengan tampilan 16x2 baris

dengan konsumsi daya rendah, Modul LCD tersebut saat ini telah dilengkapi berbagai fitur tambahan seperti mikrokontroler yang dirancang khusus untuk mengendalikan LCD dalam menentukan jumlah karakter baik kolom maupun barisnya. Salah satu Mikrokontroler yang digunakan pada LCD seri LM016L adalah HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD. Mikrokontroller tersebut diarsitektur untuk memiliki fitur berupa CCRGM(Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory) dan DDRAM (Display Data Random Access Memory). Berikut merupakan bentuk fisik dan sambungan kaki-kaki dari LCD LM016L dengan mikrokontroler:



Gambar 4: Bentuk Fisik LCD 16x2

Sumber : www.leselektronika.com/2012/06/liguid-crystal-display-lcd-16-x-2

Tabel 1. Deskripsi kaki-kaki LCD LM016L.

No.	Nama Pin	Deskripsi
1.	GND	0V
2.	VCC	+5V
3.	VEE	Kontras LCD

4.	RS	Register Select
5.	R/W	1= Read, 0 = Write
6.	EN	Enable LCD,1=enable
7.	D0	Data Bus 0
8.	D1	Data Bus 1
9.	D2	Data Bus 2
10.	D3	Data Bus 3
11.	D4	Data Bus 4
12.	D5	Data Bus 5
13.	D6	Data Bus 6
14.	D7	Data Bus 7
15.	Anoda	Anoda Backlight LED
16.	Katoda	Katoda Backlight LED

Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada tabel diskripsi, interface LCD merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa clock EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroller mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus mengatur EN ke kondisi high “1” dan kemudian mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

G. Mikrokontroller ATMega 2560

Mikrokontroller tersusun atas prosessor, memori dan I/O yang terintegrasi secara satu kesatuan dalam sebuah chip sebagai pengendali utama dari suatu sistem. Mikrokontroller dapat dikatakan sebagai mini komputer dengan kemampuan kerja yang dapat disesuaikan dengan kebutuhan sistem.

Mikrokontroller jenis AVR merupakan mikrokontroller yang banyak digunakan untuk saat ini, salah satu aplikasinya adalah penggunaan sebagai *main controller* dari robot manipulator 4-DOF tersebut, secara garis besar mikrokotroller jenis AVR merupakan mikrokontroller yang lebih fleksibel dalam pemrogramannya karena mikrokontroller jenis AVR banyak didukung untuk pemrograman komputer berbasis bahasa C.

Atmel merupakan sebuah perusahaan mikokontroller yang terkenal dengan produk mikrokontroller seri AT89S51/52-nya, namun seri AT89S51/52 memiliki keterbatasan pada resolusi, memori, dan kecepatan sehingga banyak orang beralih ke mikrokontroler AVR yang memiliki kelebihan berupa berteknologi RISC (Reduce Insruction Set Computing), sedangkan seri MCS51 berteknologi CISC (Complex Instruction Set Computing). AVR dapat dikelompokkan menjadi empat kelas yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFFxx. Perbedaan dari masing - masing keluarga AVR tersebut adalah memori, peripheral, dan fungsinya. Chip mikrokontroler AVR yang digunakan untuk tugas akhir ini adalah ATmega 2560 yang mana mikrokontroller tersebut memiliki kapasitas flash memori 256KB. AVR (Alf and Vegard’s Risc Processor) merupakan seri mikrokontroller CMOS 8-bit buatan Atmel, berbasis arsitektur

RISC (Reduced Instruction Set Computer). Berikut merupakan fitur dan spesifikasi dari ATmega 2560 :

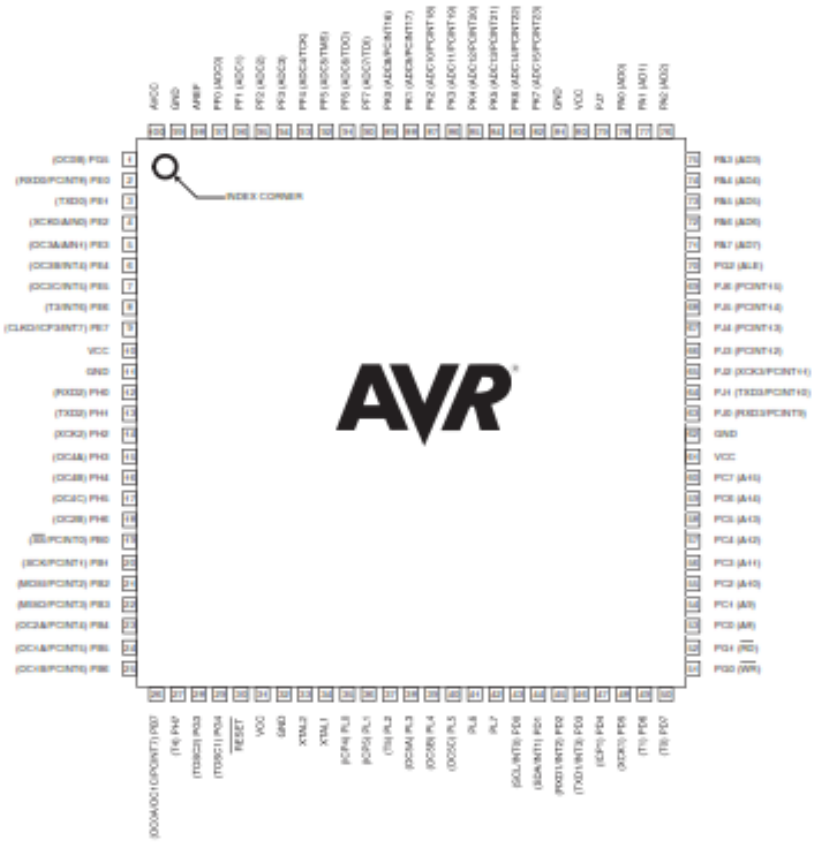
1. Spesifikasi Umum dari Atmega 2560 (Atmel AVR ATmega 2560: 2014, summary)

- High Performance, Low Power Atmel AVR 8-Bit Microcontroller
- Advanced RISC Architecture:
- 135 Powerful Instructions – Most Single Clock Cycle Execution
- 32 × 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16MHz
- On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
- 64K/128K/256KBytes of In-System Self-Programmable Flash
- 4Kbytes EEPROM
- 8Kbytes Internal SRAM
- Write/Erase Cycles:10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85 ^{o}C/ 100 years at 25 ^{o}C
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program
- True Read-While-Write Operation
- Programming Lock for Software Security Atmel

- Endurance: Up to 64Kbytes Optional External Memory Space
- Boundary-scan Capabilities According to the JTAG Standard
- Extensive On-chip Debug Support
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
- Real Time Counter with Separate Oscillator
- Four 8-bit PWM Channels
- Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits(ATmega1281/2561, ATmega640/1280/2560)
- Output Compare Modulator 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
- ATmega2560/ATmega2561: 0 - 16MHz @ 4.5V - 5.5V

2. Konfigurasi pin ATmega 2560

ATmega2560 mempunyai 86 jalur yang bisa digunakan sebagai input/output. ATmega2560 terdiri dari 12 port yang memiliki fungsinya masing-masing. Konfigurasi pin mikrokontroler ATmega2560 dapat dilihat pada gambar 5:



Gambar 5 : Konfigurasi pin ATmega2560

Sumber : Atmel (2014, 2)

Berikut merupakan penjelasan dari setiap kaki yang ada di Mikrokotroller ATmega 2560 :

a. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.

b. GND merupakan pin ground.

c. Reset untuk mereset mikrokontroler

d. Port A (PA0 – PA7) sebagai input ADC

e. Port B (PB0 – PB7) sebagai input/output dua arah

f. Port C (PC0 – PC7) sebagai input/output dua arah

g. Port D (PD0 – PD7) sebagai input/output dua arah

h. Port E (PE0 – PE7) sebagai input/output dua arah

i. Port F (PF0 – PF7) sebagai input ADC

j. Port G (PE0 – PE7) sebagai input/output dua arah

k. Port H (PE0 – PE7) sebagai input/output dua arah

l. Port E (PE0 – PE7) sebagai input/output dua arah

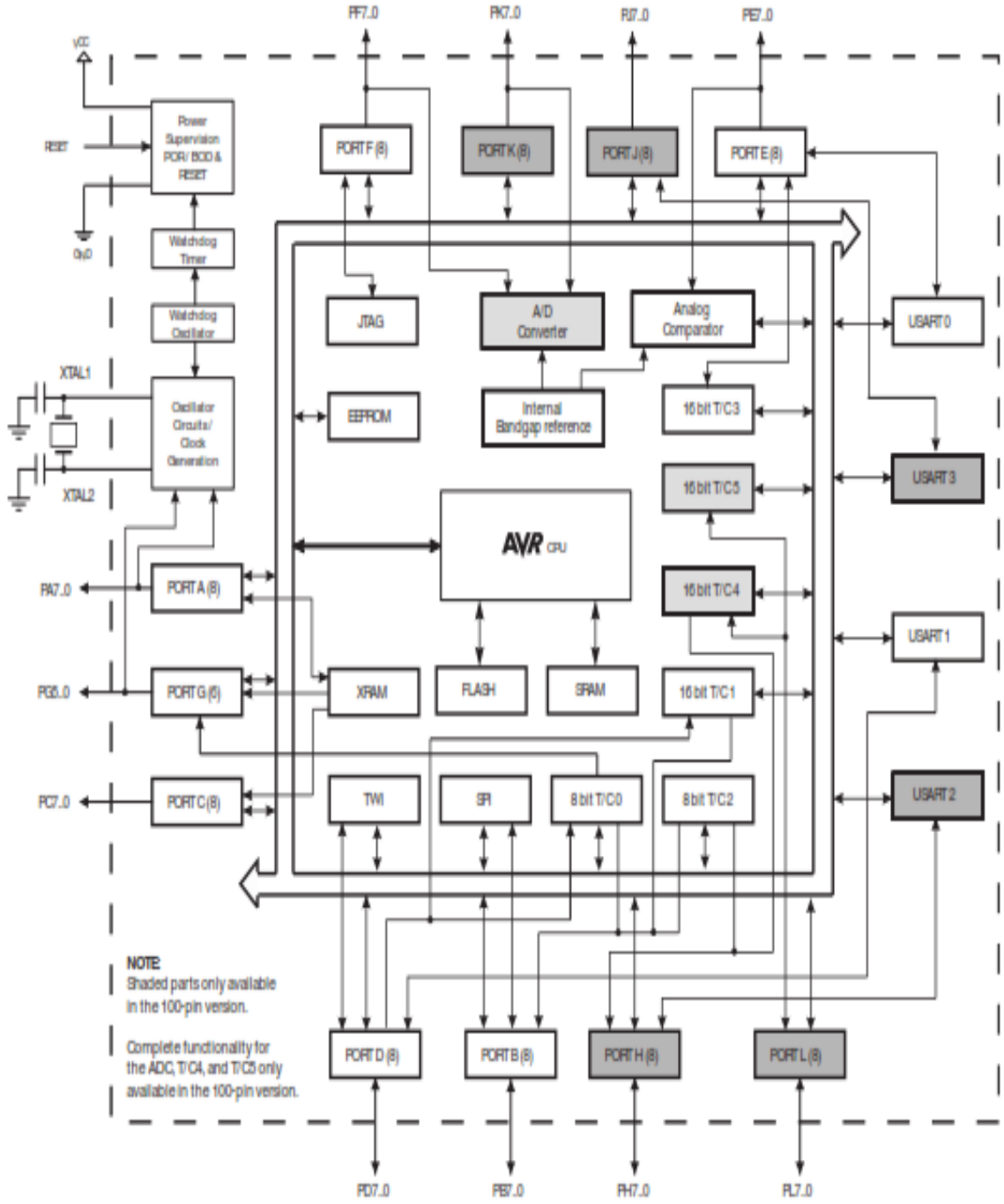
m. Port E (PE0 – PE7) sebagai input/output dua arah

n. Port E (PE0 – PE7) sebagai input/output dua arah

o. Port E (PE0 – PE7) sebagai input/output dua arah

3. Blok Diagram ATmega2560

AVR ATmega 2560 memiliki keunggulan dalam jumlah port yang relatif lebih banyak dan saling terhubung dalam ALU(Arithmetic Logic Unit), sehingga memudahkan pengguna untuk melakukan koneksi 2 arah yaitu input dan output dengan jumlah data yang banyak dengan kapasitas *clock* frekuensi hingga 16 MHz.



Gambar 6: Diagram Block AVR ATmega2560

Sumber : Atmel (2014, 5)

H. CodeVision AVR

CodeVisionAVR merupakan sebuah cross-compiler C, Integrated Development Environtment (IDE), dan Automatic Program Generator yang didesain untuk mikrokontroler buatan Atmel seri AVR(Code Vision AVR 1.25.7 user manual, 2007 : 8). Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem embedded.

File object COFF hasil kompilasi dapat digunakan untuk keperluan debugging pada tingkatan C, dengan pengamatan variabel, menggunakan debugger Atmel AVR Studio. IDE mempunyai fasilitas internal berupa software AVR Chip In-System Programmer yang memungkinkan Anda untuk melakukan transfer program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/asembli secara otomatis. Software In-System Programmer didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 programmers/development boards.

Untuk keperluan debugging sistem embedded, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal. Selain library standar C, CodeVisionAVR juga mempunyai library tertentu untuk:

- Modul LCD alphanumeric

- Bus I2C dari Philips
- Sensor Suhu LM75 dari National Semiconductor
- Real-Time Clock: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor
- Protokol 1-Wire dari Maxim/Dallas Semiconductor
- Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor
- Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor
- EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor
- SPI
- Power Management
- Delay
- Konversi ke Kode Gray

CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengujinkan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- Set-up akses memori eksternal
- Identifikasi sumber reset untuk chip
- Inisialisasi port input/output
- Inisialisasi interupsi eksternal
- Inisialisasi Timer/Counter
- Inisialisasi Watchdog-Timer

- Inisialisasi UART (USART) dan komunikasi serial berbasis buffer yang digerakkan oleh interupsi
- Inisialisasi Pembanding Analog
- Inisialisasi ADC
- Inisialisasi Antarmuka SPI
- Inisialisasi Antarmuka Two-Wire
- Inisialisasi Antarmuka CAN
- Inisialisasi Bus I2C, Sensor Suhu LM75, Thermometer/Thermostat DS1621 dan Real-Time Clock PCF8563, PCF8583, DS1302, dan DS1307
- Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20
- Inisialisasi modul LCD

CodeVisionAVR merupakan hak cipta dari Pavel Haiduc, HP InfoTech s.r.l. CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya codewizard, fasilitas ini memudahkan pengguna dalam inisialisasi mikrokontroler yang akan digunakan.

I. Metode Logika Fuzzy

1. Pengantar Logika fuzzy

Munculnya logika fuzzy dilatarbelakangi oleh adanya kesenjangan antara hukum-hukum matematika dengan permasalahan sesungguhnya dikehidupan nyata. Dengan demikian perlu suatu metode analisa baru untuk mendekati solusi yang optimal terhadap permasalahan real. Metode tersebut dikenal sebagai logika fuzzy (logika kabur atau tidak jelas).

Ada beberapa alasan digunakan logika fuzzy, antara lain:

a. Konsep logika fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
b. Logika fuzzy sangat fleksibel.
c. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
d. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
e. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
f. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
g. Logika fuzzy didasarkan pada bahasa alami.

2. Himpunan Fuzzy

Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan A[x], memiliki 2 kemungkinan, yaitu:

a. satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
b. nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Contoh:

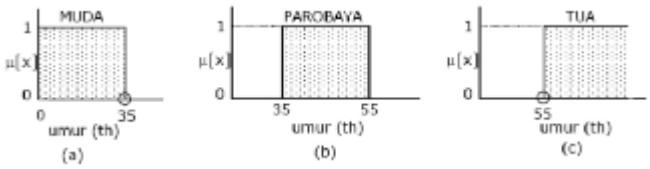
Misalkan variabel umur dibagi menjadi 3 kategori, yaitu:

MUDA umur < 35 tahun

PAROBAYA $35 \geq$ umur ≤ 55 tahun

TUA umur > 55 tahun

Nilai keanggotaan secara grafis, himpunan MUDA, PAROBAYA dan TUA ini dapat dilihat pada Gambar 7.



Gambar 7. Himpunan : MUDA, PAROBAYA dan TUA

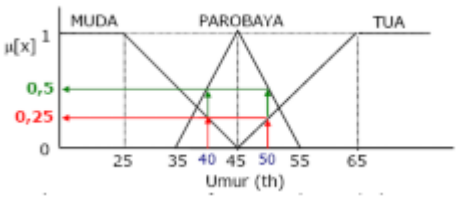
Sumber : Kusumadewi (2010,4)

Menurut gambar 6 dapat dilihat bahwa:

a. apabila seseorang berusia 34 tahun, maka ia dikatakan MUDA (MUDA[34]=1);
b. apabila seseorang berusia 35 tahun, maka ia dikatakan TIDAK MUDA (MUDA[35]=0);
c. apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK MUDA (MUDA[35 th -1hr]=0);
d. apabila seseorang berusia 35 tahun, maka ia dikatakan PAROBAYA (PAROBAYA[35]=1);
e. apabila seseorang berusia 34 tahun, maka ia dikatakan TIDAK PAROBAYA (PAROBAYA[34]=0);
f. apabila seseorang berusia 35 tahun, maka ia dikatakan PAROBAYA (PAROBAYA[35]=1);
g. apabila seseorang berusia 35 tahun kurang 1 hari, maka ia dikatakan TIDAK PAROBAYA (PAROBAYA[35 th- 1 hr]=0);

Dari sini bisa dikatakan bahwa pemakaian himpunan crisp untuk menyatakan umur sangat tidak adil, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan.Himpunan fuzzy digunakan untuk mengantisipasi hal tersebut. Seseorang dapat masuk dalam 2 himpunan yang berbeda, MUDA dan PAROBAYA, PAROBAYA dan TUA, dsb. Seberapa besar eksistensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya.

Gambar 8 menunjukkan himpunan fuzzy untuk variabel umur.



Gambar 8. Himpunan Fuzzy Untuk Variabel Umur

Sumber : Kusumadewi (2010,5)

a. seseorang yang berumur 40 tahun, termasuk dalam himpunan MUDA dengan μMUDA[40]=0,25; Namun dia juga termasuk dalam himpunan PAROBAYA dengan μPABOBAYA[40]=0,5.
b. seseorang yang berumur 50 tahun, termasuk dalam himpunan TUA dengan μTUA[50]=0,25; namun dia juga termasuk dalam himpunan PAROBAYA dengan μPABOBAYA[50]=0,5.

Jika pada himpunan crisp, nilai keanggotaan hanya ada 2 kemungkinan, yaitu 0 atau 1,maka pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan fuzzy A[x]=0 berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan fuzzy A[x]=1 berarti x menjadi anggota penuh pada himpunan A.

Himpunan fuzzy memiliki 2 atribut.

a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA.
b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu:

a. Variabel fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan, dsb.

b. Himpunan fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

d. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy.

Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

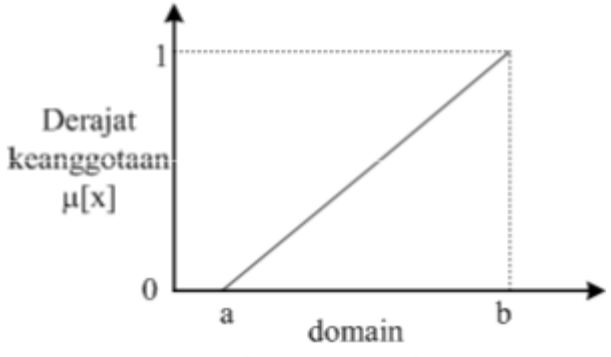
3. Fungsi Keanggotaan

Fungsi Keanggotaan (membership function) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

Ada beberapa fungsi yang bisa digunakan diantaranya :

a. Representasi Linear

Pada representasi linear, pemetaan input ke derajat keanggotannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti terlihat pada gambar 9.



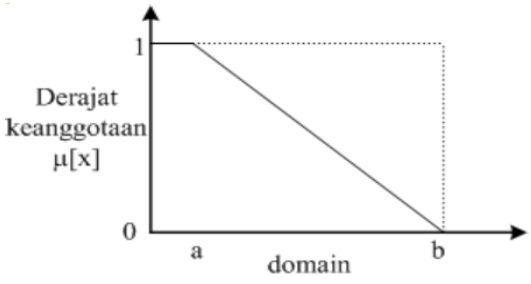
Gambar 9: Representasi Linear Naik

Sumber : Kusumadewi (2010,9)

Fungsi keanggotaan kurva linear naik :

$$\mu[x] = \begin{cases} 0 & ,x < a \\ (x-a)/(b-a) & ,a \le x \le b \\ 1 & ,x > b \end{cases}$$

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah seperti terlihat pada gambar 10.



Gambar 10: Representasi linear turun

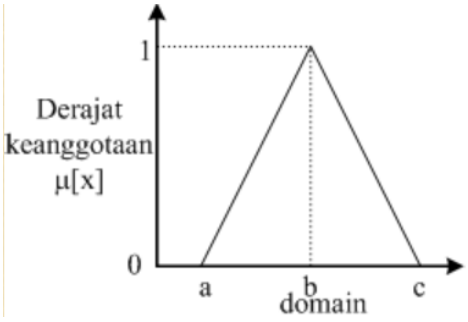
Sumber : Kusumadewi (2010,10)

Fungsi keanggotaan Linear Turun

$$\mu[x,a,b] = \begin{cases} (b\text{-}x) / (b\text{-}a); & a \leq x \leq b \\ 0; & x \geq b \end{cases}$$

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis linear naik dan linear turun seperti terlihat pada gambar 11.



Gambar 11: Kurva Segitiga

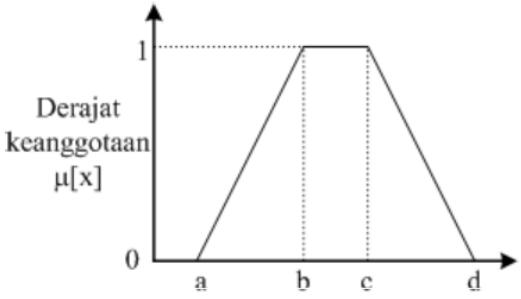
Sumber : Kusumadewi (2010,11)

Fungsi keanggotaan Kurva Segitiga

$$\mu\,[x,a,b,c] = \begin{cases} 0 & x \leq a \text{ atau } x \geq c \\ (x\text{-}a) / (b\text{-}a); & a \leq x \leq b \\ (c\text{-}x) / (c\text{-}b); & b \leq x \leq c \end{cases}$$

c. Representasi Kurva Trapesium

Kurva Trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Gambar 12 menunjukkan kurva trapesium.



Gambar 12: Kurva Trapesium

Sumber : Kusumadewi (2010,11)

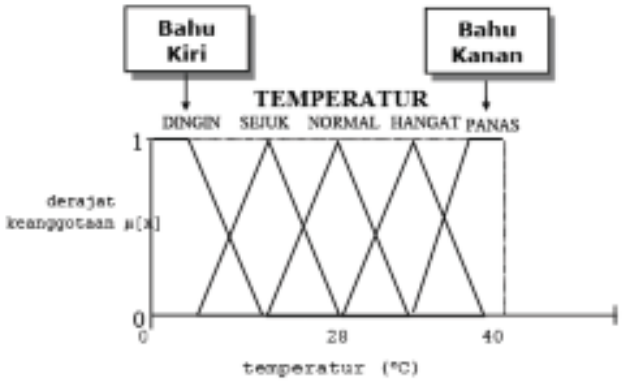
Fungsi keanggotaan Kurva Trapesium

$$\mu[x,a,b,c,d] \begin{cases} 0; & x \leq a \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \\ 0; & x \geq d \end{cases}$$

d. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan: DINGIN bergerak ke SEJUK bergerak ke HANGAT dan bergerak ke PANAS). Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi PANAS, kenaikan temperatur akan tetap berada pada kondisi PANAS. Himpunan fuzzy ‘bahu’, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari

benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Gambar 13 menunjukkan variabel TEMPERATUR dengan daerah bahunya.



Gambar 13: Daerah Bahu pada Variabel TEMPERATUR

Sumber : Kusumadewi (2010, 14)

4. Operator Dasar Zadeh untuk Operasi Himpunan Fuzzy

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama fire strength atau –predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu:

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α–predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A \cap B = \min(\mu A[x], \mu B[y])$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α–predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu A \cup B = \max(\mu A[x], \mu B[y])$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α–predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu A' = 1-\mu A[x]$$

5. Sistem Inferensi Fuzzy

Sistem inferensi fuzzy adalah proses transformasi dari suatu input dalam domain fuzzy ke suatu output dalam domain fuzzy. Proses transformasi pada bagian inferensi membutuhkan aturan-aturan fuzzy yang terdapat di dalam basis-basis aturan. Blok inferensi menggunakan teknik penalaran untuk menyeleksi basis-basis aturan dari basis pengetahuan. Metode inferensi yang sering digunakan dalam logika fuzzy adalah:

a. Metode Tsukamoto

Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (crisp) berdasarkan α-predikat. Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

b. Metode Sugeno

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI, hanya saja output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan metode TSK. Metode TSK terdiri dari 2 jenis, yaitu:

1. Model Fuzzy Sugeno Orde-Nol

   Secara umum bentuk model fuzzy SUGENO Orde-Nol adalah:

   IF (x1 is A1) (x2 is A2) (x3 is A3) … (xN is AN) THEN z=k

   dengan Ai adalah himpunan fuzzy ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

2. Model Fuzzy Sugeno Orde-Satu

   Secara umum bentuk model fuzzy SUGENO Orde-Satu adalah:

IF (x1 is A1)…(xN is AN) THEN z = p1*x1 + … + pN*xN + q

dengan Ai adalah himpunan fuzzy ke-i sebagai anteseden, dan pi adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode SUGENO, maka deffuzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

c. Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan 4 tahapan.

1) Pembentukan himpunan fuzzy

Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2) Aplikasi fungsi implikasi

Pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

3) Komposisi Aturan

Tidak seperti penalaran monoton, apabila sistem terdiri-dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu: max, additive dan probabilistik OR (probor).

a) Metode Max (Maximum)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke output dengan menggunakan operator OR (union). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan konstribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan:

$$\mu sf[xi] = \max(\mu sf[xi],\ \mu kf[xi])$$

Dengan :

$\mu sf[xi] =$ nilai Keanggotaan solusi fuzzy sampai aturan ke $-\ i$;

$\mu kf[xi] =$ nilai Keanggotaan konsekuen fuzzy hingga aturan ke $-\ i$;

b) Metode Additive (Sum)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan bounded-sum terhadap semua output daerah fuzzy. Secara umum dituliskan:

$$\mu sf[xi] = \min(1, \mu sf[xi] + \mu\ kf[xi])$$

Dengan :

$\mu sf[xi] =$ nilai Keanggotaan solusi fuzzy sampai aturan ke $-\ i$;

$\mu kf[xi] =$ nilai Keanggotaan konsekuen fuzzy hingga aturan ke $-\ i$;

c) Metode Probabilistik OR (probor)

Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara melakukan product terhadap semua output daerah fuzzy. Secara umum dituliskan:

$$\mu sf[xi] = (\mu sf[xi] + \mu kf[xi]) - (\mu sf[xi] * \mu kf[xi])$$

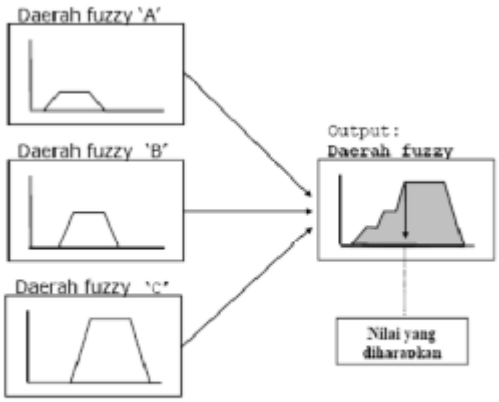
Dengan :

$\mu sf[xi] =$ nilai Keanggotaan solusi fuzzy sampai aturan ke $-\ i$;

$\mu kf[xi] =$ nilai Keanggotaan konsekuen fuzzy hingga aturan ke $-\ i$;

4) Penegasan (defuzzy)

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai crips tertentu sebagai output seperti terlihat pada gambar 14.



Gambar 14: Proses Defuzzyfikasi

Sumber : kusumadewi (2010, 40)

Ada beberapa metode defuzzifikasi pada komposisi aturan MAMDANI, antara lain:

a) Metode Centroid (Composite Moment)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil titik pusat (z*) daerah fuzzy. Secara umum dirumuskan:

$$z^* = \frac{\int_Z z.\mu(z)dz}{\int_Z \mu(z)dz}$$

$$z^* = \frac{\sum_{j=1}^{n} z_j \mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)}$$

b) Metode Bisektor

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai pada domain fuzzy yang memiliki nilai keanggotaan separo dari jumlah total nilai keanggotaan pada daerah fuzzy. Secara umum dituliskan:

$$z_p \quad \text{sedemikian hingga} \quad \int_{R1}^{p} \mu(z)\, dz = \int_{p}^{Rn} \mu(z)\, dz$$

c) Metode Mean of Maximum (MOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d) Metode Largest of Maksimum (LOM)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e) Metode Smallest of Maksimum (SOM)

Solusi crisp diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

6. Fuzzy Reasoning

Fuzzy reasoning merupakan prosedur inferensi yang menghasilkan kesimpulan dari satu set if-then-rules dan fakta yang diketahui. Misalkan:

A = Hari ini cerah

A→B : hari = cerah maka langit = biru

Inferensi : langit biru

Premis 1 (Fakta) : Hari ini cerah (x = A)

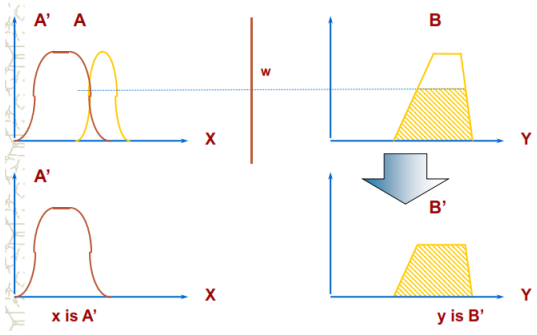
Premis 2 (Rule) : Jika hari ini cerah maka langit berwarna biru

(if x = A then y = B)

Kesimpulan : Hari ini langit berwarna biru (x = B)

Diasumsikan bahwa implikasi fuzzy A → B dinyatakan sebagai R = x*y. Maka himpunan fuzzy B disebabkan oleh x dan rule ”if x = A then y = B” didefinisikan sebagai: μB(y) = maxmin[μA’(x), μR(x,y)]

a. Single Rule Single Antecedent



Gambar 15: Fuzzy Reasoning single rule single antecedent

Sumber : J.S.R Jang, C.T Sun, E.Mizutani (1997)

Gambar 15 merupakan gambaran proses Fuzzy reasoning single rule single antecedent.

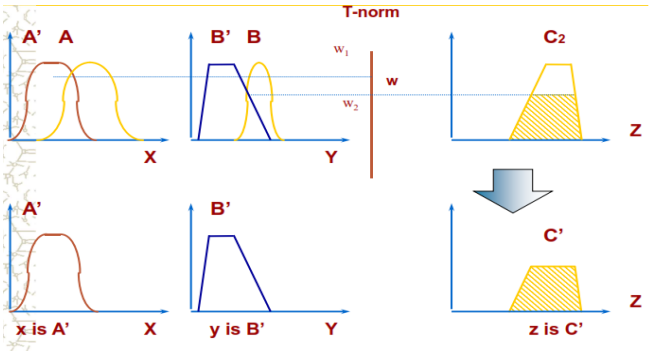
Rule : if x adalah A maka y adalah B

Fakta : x adalah A

Kesimpulan : y is B’ (μB’(y) = [μx(μA’(x) ∩ μA(x)] ∩ μB(y))

Untuk lebih jelasnya dapat dilihat pada gambar diatas.

b. Single Rule Multiple Antecedents



Gambar 16: Fuzzy reasoning Single rule Multiple Antecedents

Sumber : J.S.R Jang, C.T Sun, E.Mizutani (1997)

Gambar 16 merupakan gambaran proses Fuzzy reasoning single rule multiple antecedent.

Premis 1 (Fakta) : x adalah A’ dan y adalah B’

Premis 2 (Rule) : Jika x adalah A dan y adalah B maka z adalah C

Kesimpulan : z adalah C’

Premis 2: A*B → C

$$R_{mamdani}(A,B,C) = (A*B)*C = \int_{X*Y*Z} \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)/(x,y,z)$$

$$C' = \underbrace{(A'*B')}_{\text{premise 1}} \circ \underbrace{(A*B \rightarrow C)}_{\text{premise 2}}$$

$$\mu_{C'}(z) = \bigvee_{x,y}[\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)]$$

$$= \bigvee_{x,y}\{[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)]\} \wedge \mu_C(z)$$

$$= \underbrace{\left\{\bigvee_x[\mu_{A'}(x) \wedge \mu_A(x)]\right\}}_{w1} \wedge \underbrace{\left\{\bigvee_y[\mu_{B'}(y) \wedge \mu_B(y)]\right\}}_{w2} \wedge \mu_C(z)$$

$$= (w_1 \wedge w_2) \wedge \mu_C(z)$$

c. Multiple Rules Multiple Anticedents

Premis 1 (Fakta) : x adalah A' dan y adalah B'

Premis 2 (Rule 1) : Jika x adalah A1 dan y adalah B1 maka z adalah C1

Premis 3 (Rule 2) : Jika x adalah A2 dan y adalah B2 maka z adalah C2

Kesimpulan : z adalah C'

R1 = A1 dan B1→ C1

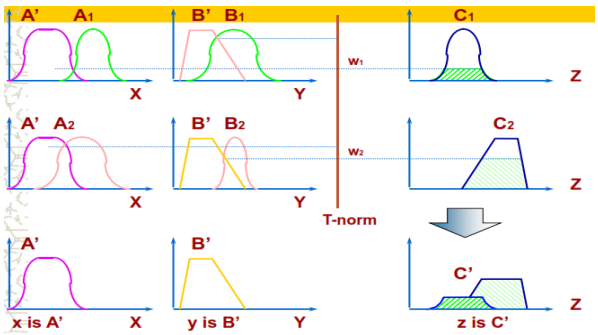
R2 = A2 dan B2 → C2

Karena max – min Operator komposisi o adalah distributive atas

operator union, maka:

C' = (A' * B') o (R1∪ R2) = [(A' * B') o R1] ∪ [(A' * B') o R2]

= C'1 ∪ C'2



Gambar 17 : Fuzzy reasoning multiple rules multiple antecedents

Sumber : J.S.R Jang, C.T Sun, E.Mizutani (1997)

Pada gambar 17 dapat dilihat dimana C'1 dan C'2 adalah himpunan fuzzy tereka untuk aturan 1 dan 2 masing-masing.

J. OpenCV

OpenCV (Open Computer Vision) adalah sebuah API (Application Programming Interface) Library yang sering digunakan sebagai dasar untuk Pengolahan Citra (http://priawadi.com/2012/09/opencv.html). Pada dasarnya penggunaan openCV sebagai library pengolahan citra ditujukan sebagai visualisasi penglihatan komputer atau yang biasa disebut dengan Computer Vision, Computer Vision adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia dengan fleksibel namun terbatas pada program awal. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dll. OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video.

OpenCV sendiri terdiri dari 5 library, yaitu :

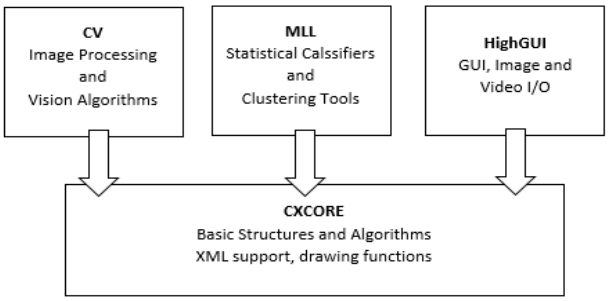
CV : untuk algoritma Image processing dan Vision.

ML : untuk machine learning library

Highgui : untuk GUI, Image dan Video I/O.

CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.

Struktur dan konten didalam OpenCV dapat dideskripsikan pada gambar Gambar 18:



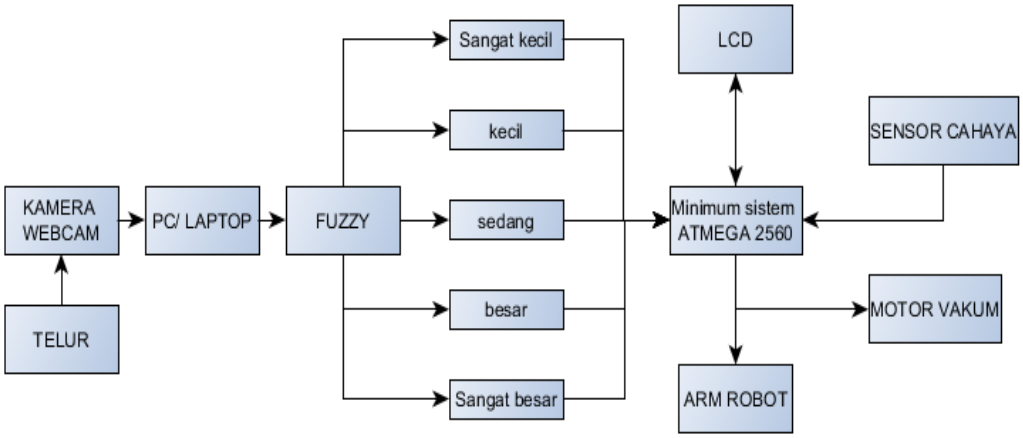
Gambar 18. Struktur dan konten OpenCV

Sumber : http://www.priawadi.com/2012/09/opencv.html

BAB III

KONSEP RANCANGAN

Perancangan Prototipe dan Implementasi Penyortir Telur dengan Menggunakan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom*(DOF) menggunakan metode **ADDIE** (Dick dan Carey, 2010) pada proses rancang bangunnya, dalam pengaplikasiannya model ADDIE(*Analysis, Design, Development, Implementation, and* Evaluation) adalah metode yang berisi jajaran mengenai proses generik dalam melakukan suatu desain instruksional. Metode ini menerapkan pedoman deskriptif untuk mendukung kinerja yang efektif dalam 5 langkah yaitu Analisis yang bertujuan untuk melakukan identifikasi kebutuhan yang diperlukan. Desain yang bertujuan umtuk merancang dan merekonstruksi gambaran perencanaan dari alat baik berupa desain mekanik, desain elektronik dan desain perangkat lunak. Development ditujukan untuk pengembangan lebih lanjut atau perealisasian dari desain mekanik, desain elektronik dan desain perangkat lunak. Implementasi bertujuan untuk mensinergikan dari tiap mekanik, elektronik dan perangkat lunak dan yang terakhir adalah proses Evaluasi yang bertujuan untuk mengevaluasi segala kekurangan dan kelebihan dari proyek akhir ini untuk didapatkan kinerja yang lebih optimal. Secara garis besar blog diagram singkat mengenai konsep rancangan proyek akhir ini dapat dilihat pada gambar 19:



Gambar 19. Blog Diagram Singkat Konsep Perancangan

Pembuatan alat "Prototype dan Implementasi Penyortir Telur dengan Menggunakan Logika Fuzzy pada Manipulator 6-DOF" sebagai alat untuk mensortir telur. Dengan berdasarkan kemampuan sistem cerdas yang telah dibangun untuk mendeteksi ciri fisik telur menggunakan logika fuzzy, manipulator robot tersebut dinilai akan mampu mensortir telur lebih baik lagi, logika fuzzy diterapkan pada program dari pengolahan citra yang telah disusun menggunakan opencv sebagai library pengolahan citra, setelah program logika fuzzy dengan pengolahan citra terbangun, lankah selanjutnya yaitu mengintegrasikan dengan mikrokontroller ATmega 2560 sebagai kontroler utama untuk mengolah robot, kemudian data hasil olahan citra akan digunakan sebagai parameter untuk menentukan letak penataan telur berdasarkan ukuran. Berikut merupakan uraian singkat mengenai kinerja tiap komponen yang digunakan :

1. Webcam digunakan sebagai media pengolahan citra dari telur untuk mendapatkan ukuran telur.

2. Mikrokontroller ATmega 2560 sebagai *main control system* dan pengolah data yang diterima dari Webcam melalu pc/laptop.
3. Lcd display 16x2 sebagai penampil olahan data dari mikrokontroller dan sebagai indikator pergerakan dari alat.
4. Servo Standar sebagai aktuator penggerak manipulator 6-DOF.
5. Motor DC sebagai aktuator untuk vakum dari manipulator 6-DOF
6. Power Supply 12V adaptor dengan UBEC 3A sebagai *suplay* utama manipulator robot.

A. ANALISIS

Analisis pada proyek akhir ini memerlukan beberapa rincian mengenai kebutuhan alat dan bahan guna menunjang proses pembuatannya, pada tahap analisa akan dibagi menjadi 2 topik yakni Identifikasi kebutuhan dan Analisis Kebutuhan, ditiap bagian identifikasi kebutuhan dan analisa kebutuhan akan di bahas lebih mendalam mengenai bagian *hardware* maupun *software*.

1. Identifikasi Kebutuhan

Pada pengerjaan proyek akhir ini, Robot difungsikan sebagai penyortir telur dengan proses yang dijabarkan sebagai berikut :

1. Pengolahan citra mampu untuk mengestraksi ukuran telur.
2. Ukuran telur diklasifikasikan kedalam 5 jenis ukuran telur yakni sangat kecil, kecil, sedang, besar dan sangat besar.

3. Catu daya pada perangkat elektronik dapat terpenuhi sesuai dengan datasheet masing-masing komponen elektronik.
4. Robot manipualtor robot 6-*Degree of Freedom* dapat memindahkan telur kedalam tatakan telur sesuai dengan klasifikasi jenis telur dan berurutan.

Adapun beberapa kebutuhan untuk dapat menunjang pengerjaan proyek akhir ini yang dibagi menjadi perangkat *Hardware* dan S*oftware* :

a) Hardware

Pada bagian hardware akan dibahas mengenai perancangan mekanik dan perancangan elektronik.

- **Perancangan Mekanik**

  Untuk menunjang dari konsep perancangan mekanik ada 3 rincian yang harus diperhatikan :

  1. Perancangan manipulator robot

     Dalam pengerjaan proyek akhir ini, manipulator robot digunakan sebagai media untuk penyortir telur, oleh sebab itu untuk menunjang kinerja dari manipulator robot dibutuhkan yang *fleksibel* untuk dapat memindah telur dari tempat deteksi telur ke tempat tatakan telur. Kondisi umum dari manipulator robot memiliki 2 – 6 *Degree of Freedom(DOF),* namun untuk menentukan jumlah dari *Degree of Freedom(DOF)* tersebut akan disesuaikan dengan kebutuhan pengerjaan proyek akhir ini.

  2. Perancangan tatakan telur

Tatakan telur akan didesain sedemikian rupa dengan tujuan untuk memberi gerak yang mudah dijangkau oleh manipulator robot serta jumlah dari tiap klasifikasi telur dapat tertampung dengan baik dan benar.

3. Perancangan *base* telur dan kamera

   Base telur dan kamera akan didesain sesuai dengan kemampuan jangka manipulator robot untuk mengambil telur yang telah diidentifikasi, begitu juga dengan ketinggan antara kamera dengan tempat peletakkan telur yang harus disesuaikan satu sama lain untuk menghindari *crash* antara telur dengan kamera ketika telur telah terambil oleh manipulator robot.

- **Perancangan Elektronik**

Sedangkan untuk menunjang perancangan elektronik, secara global perancangan elektronik akan disesuaikan dengan kebutuhan input dari tiap komponen elektronik seperti minimum sistem ATmega 2560, Servo standar, UBEC penurun tegangan, sensor cahaya LDR dsb. Tabel 2 merupakan ringkasan secara keseluruhan mengenai Identifkasi kebutuhan dari bagian Elektronik.

Tabel 2. Identifikasi Kebutuhan Perancangan Elektronik

No	Rangkaian	Komponen	Spesifikasi
1	*Power Supply*	Adaptor	12 Volt 5 Amperre
		UBEC	5 Volt 3 Ampere
2	Minimum Sistem dan *Shield* Minimum sistem	Mikrokontroller	ATmega 2560 In. 5 Volt
		Crystall	16 MHz
		konektor	Pin header female
		Konektor	Pin header male

		DC power jack	DC power jack 12 V 3.5mm
3	Input	Sensor cahaya photodiode	LED dan photodiode 3mm warna biru/ in. 5 Volt ADC
4	*Output*	Driver motor relay	5 kaki out 12 Volt
		Motor DC Vakum	In. 12 volt 0.5 pascal
		Servo standar	In. 5Volt, Torsi 10 Kg/cm

b) Software

Pada bagian *Software* juga akan dibahas mengenai identifkasi kebutuhan guna menunjang pengerjaan proyek akhir ini, antara lain :

1. Pemrograman pengolahan citra

   Untuk mendapatkan ukuran telur secara real time maka digunakan pemrograman pengolahan citra untuk mendukung pengerjaan proyek akhir ini. Pengolahan citra dewasa ini memang telah menjadi salah satu basis sensor yang dinilai mampu untuk melakukan tugas sebagai pengganti mata pada teknologi robot, oleh karena kemampuannya tersebut pengolahan citra akan digunakan sebagai identifikator dari ukuran telur.

2. Sistem logika fuzzy

   Untuk dapat mendukung keputusan yang memiliki validitas yang baik, sistem cerdas logika fuzzy akan ditambahkan dalam pengerjaan proyek akhir ini. Sistem logika fuzzy sejauh ini mampu menjadi acuan untuk mendeskripsikan tentang keabu-abuan dari suatu permasalahan yang

terjadi, seperti halnya permasalahan untuk menklasifikasikan jenis ukuran telur yang akan diusung pada pengerjaan proyek akhir ini. Dengan menambahkan sistem cerdas berbasis logika fuzzy, diharapkan keputusan yang telah diambil untuk menentukan klasifikasi ukuran telur dapat terselesaikan dengan baik dan benar serta dapat dipertanggung jawabkan tentang metode keputusan yang diambil.

3. Pemrograman manipulator robot

   Pada pemrograman manipulator robot, ada kaitannya erat dengan perancangan elektronik, dimana untuk pengolahan-pengolahan ditiap aktuator robot maupun sensor pendukung yang digunakan akan diprogram sedemikian rupa, sehingga robot dapat begereak sesuai dengan kebutuhan yang akan disung pada pengerjaan proyek akhir ini, beberapa aktuator robot maupun sensor robot yang digunakan adalah seperti servo, sensor cahaya LDR dsb.

2. Analisis Kebutuhan

Mengacu pada identifikasi kebutuhan yang telah dipaparkan, berikut merupakan beberapa analisis kebutuhan pada bagian *hardware* maupun *software* guna menunjang pengerjaan proyek akhir ini :

a) Hardware

i) Perancangan Mekanik

Pada bagian hardware akan dibahas mengenai analis kebutuhan tentang konsep perancangan mekanik dengan 3 rincian yang telah dipaparkan pada identifkasi kebutuhan, yakni :

1. Perancangan manipulator robot

   Dalam pengerjaan proyek akhir ini, manipulator robot yang digunakan sebagai media untuk penyortir telur, akan menggunakan 6-*Degree of Freedom,* alasan penggunaan manipulator robot sebanyak 6-*Degree of Freedom* adalah karena pada riset sebelumnya dimana penulis menggunakan 4-*Degree of Freedom* terkendala oleh perancangan rangka robot yang menyebabkan gerak dari manipualtor robot menjadi terbatas ketika akan mengambil telur dari *base* telur, oleh sebab itu dipilihlah manipulator robot berbasis 6-*Degree of Freedom* untuk menunjang kinerja dari manipulator robot yang lebih *fleksibel* agar dapat memindah telur dari tempat deteksi telur ke tempat tatakan telur.

2. Perancangan tatakan telur

   Mengacu pada *judgement expert* di industri penyortir telur yang mana tiap klasifikasi telur baik sangat kecil, kecil, sedang, besar dan sangat besar memiliki rata-rata panjang dan lebar telur yang berbeda, maka dari itu tatakan telur didesain sesuai dengan panjang dan lebar dari tiap klasifikasi telur untuk tiap ruangnya sebanyak 25 ruang yang disesuaikan dengan jumlah telur yang akan digunakan sebagai data pengujian pada proyek akhir ini. Dari efisiensi ruang tersebut, manipulator robot akan lebih mudah dalam meletakkan telur pada tatakan telur tersebut.

3. Perancangan *base* telur dan kamera

Pada *base* telur dan kamera ketinggan antara kamera dengan tempat peletakkan telur didesain dengan ketinggian +-20cm, ukuran tersebut didapat dengan mengacu pada ekspansi panjang dan lebar ujung lengan manipulator robot.

ii) Perancangan Elektronik

Untuk perancangan elektronik, pada minimun sistem ATmega2560 akan dibangun sebuah *shield* dengan spesifikasi 6 Port untuk servo, 2 Port untuk Analog to Digital Converter(ADC) dan 1 buah sensor proximity.

iii) *Software*

Pada bagian *Software* yang telah dibahas pada identifkasi kebutuhan, maka selanjutnya akan dibahas mengenai analis kebutuhan dari *software* guna menunjang pengerjaan proyek akhir ini :

1. Pemrograman pengolahan citra

   Pada pengolahan citra, metode yang digunakan untuk mengolah ukuran telur adalah dengan algoritma codebook, algortima codebook dipilih karena untuk mempermudah proses *background substraction*. *background substraction* adalah suatu cara yang digunakan untuk mereduksi tampilan dari kamera webcam, *background substraction* merupakan bagian dari algoritma codebook itu sendiri sehingga pada tahapan ini *background substraction* dimaksudkan agar dapat mendeteksi satu objek yang dianggap asing dan kemudian objek asing tersebut diolah citranya hingga sedemikian rupa untuk dicari panjang

dan lebar objek tersebut. Objek asing yang dimaksud dalam proyek akhir ini adalah satu butir telur.

2. Sistem logika fuzzy

   Pada proyek akhir ini, metode fuzzy yang digunakan adalah mamdani, dengan min-max dan operator AND sebagai penalaran fuzzy, kemudian metode centroid untuk proses defuzzyfikasinya. Alasan digunakannya metode mamdani adalah metode mamdani bersifat *human inference sistem*, yang mana itu artinya metode tersbut dibuat berdasarkan dengan aturan manusia pada umumnya. Oleh sebab itu proyek akhir ini dibuat dengan metode mamdani karena telur merupakan kebutuhan manusia yang mana artinya setiap kaidah ukuran dari telur itu sendiri juga tidak lepas dari pandangan manusia pada umumnya.

3. Pemrograman manipulator robot

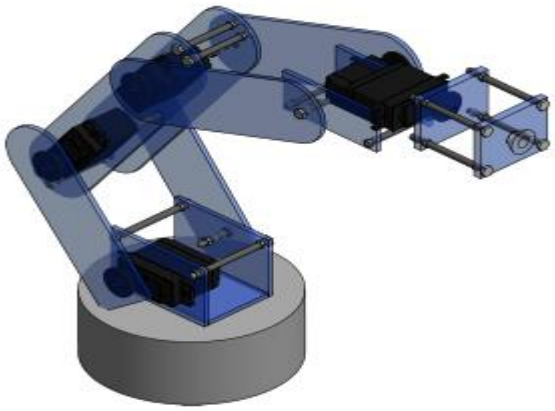
   Untuk pemrograman manipulator robot akan dibangun dengan bantuan software codevision AVR. Codevision AVR dipilih karena pada dasarnya software tersebut memang dibuat untuk mempermudah penggunaan dari mikrokontroller ATmega 2560 itu sendiri, baik dari segi inisialisasi penggunaan *analog digital converter* maupun penggunaan *pulse width modulation* untuk menggerakkan sudut servo.

B. PERANCANGAN ALAT

1. PERANCANGAN MEKANIK ROBOT

1) Desain Mekanik Manipulator Robot 6-*Degree Of Freedom*

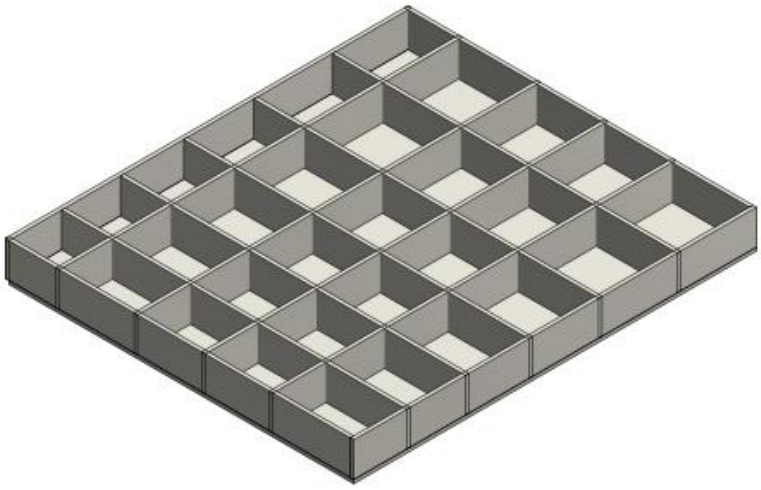
Mekanik robot berupa badan mekanik yang berfungsi sebagai penopang untuk meletakkan servo-servo, rangakaian elektronik, *power supply* dll. Perancangan mekanik merupakan bagian hal yang penting untuk menunjang kinerja robot, ketika salah satu bagian mekanik tidak dapat bekerja dengan semestinya maka bisa dipastikan keseluruhan bagian mekanik yang lain tidak akan bekerja dengan baik pula, oleh sebab itu diperlukan tingkat ketelitian yang sangat tinggi ditiap bagian mekaniknya terutama pada sendi derajat kebebasan manipulator robot berbentuk lengan, setiap derajat kebebasannya memiliki pernanan yang penting dalam menentukan posisi sudut pergerakan manipualtor robot tersebut, semakin banyak derajat kebebasan yang dimiliki oleh manipulator robot maka semakin leluasa pula pergerakannya. Pada proyek akhir ini penulis mendesain manipulator robot berbentuk *Arm Robot* yang memiliki derajat kebebasan sebanyak 6 sendi, tujuan dibuatnya *arm robot* dengan 6 sendi adalah untuk memudahkan dan memberikan efek stimulus gerakan yang lebih fleksibel sehingga ketika melakukan tugasnya robot akan mampu bekerja secara optimal, karena tidak terbatasi oleh gerak dari sendi-sendi robot itu sendiri. Gambar 20 merupakan desain mekanik dari manipulator robot 6 *Degree Of Freedom* :



Gambar 20. Desain Mekanik Manipulator Robot 6-*Degree Of Freedom* (*DOF*)

2) Desain Tatakan Telur

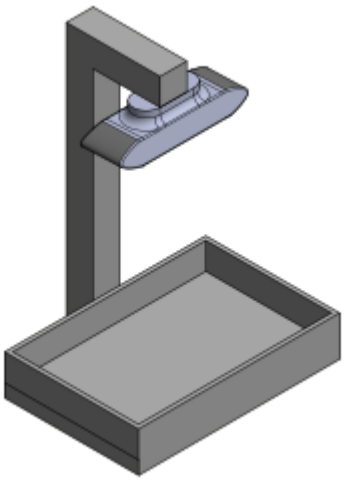
Desain tatakan telur merupakan adalah salah satu dari bagian yang vital dalam pengerjaan proyek akhir ini, tatakan telur tidak bisa sembarang dibuat karena disesuaikan dengan motion dari manipulator robot, tingkat penempatan merupakan hal yang harus diperhatikan, sebab dari keluaran output fuzzy sebanyak 5, maka sebanyak itu pula tatakan telur dibuat baik berupa baris maupun kolomnya. Setiap kolom berjumlah 5 tempat yang artinya robot memiliki 5 motion untuk tiap fungsi keanggotaan output fuzzy dan disesuaikan dengan jumlah maksimal telur yang mampu ditata oleh robot. Gambar 21 merupakan desain dari tatatkan telur :



Gambar 21. Desain Tatakan Telur

3) Desain Base Telur dengan Kamera

Perancangan desain base telur dengan kamera dimaksudkan untuk tempat dari telur ketika akan di identifikasi ciri fisiknya, desain tempat telur disesuaikan dengan jumlah rata-rata maksimal dari telur baik dari panjang maupun lebar dari telur, setelah menentukan panjang dan lebar kemudia ketinggian dari kamera juga perlu disesuaikan mengingat program pengolahan citra yang disusun juga berdasarkan ketinggian kamera, disamping itu agar manipulator mampu mengambil telur dengan baik dan benar, maka ketinggian dari kamera harus benar-benar diperhatikan untuk mengurangi masalah deteksi telur dan masalah dari pengambilan telur yang kurang sempurna, Gambar 22 merupakan desain dari base telur dengan kamera :



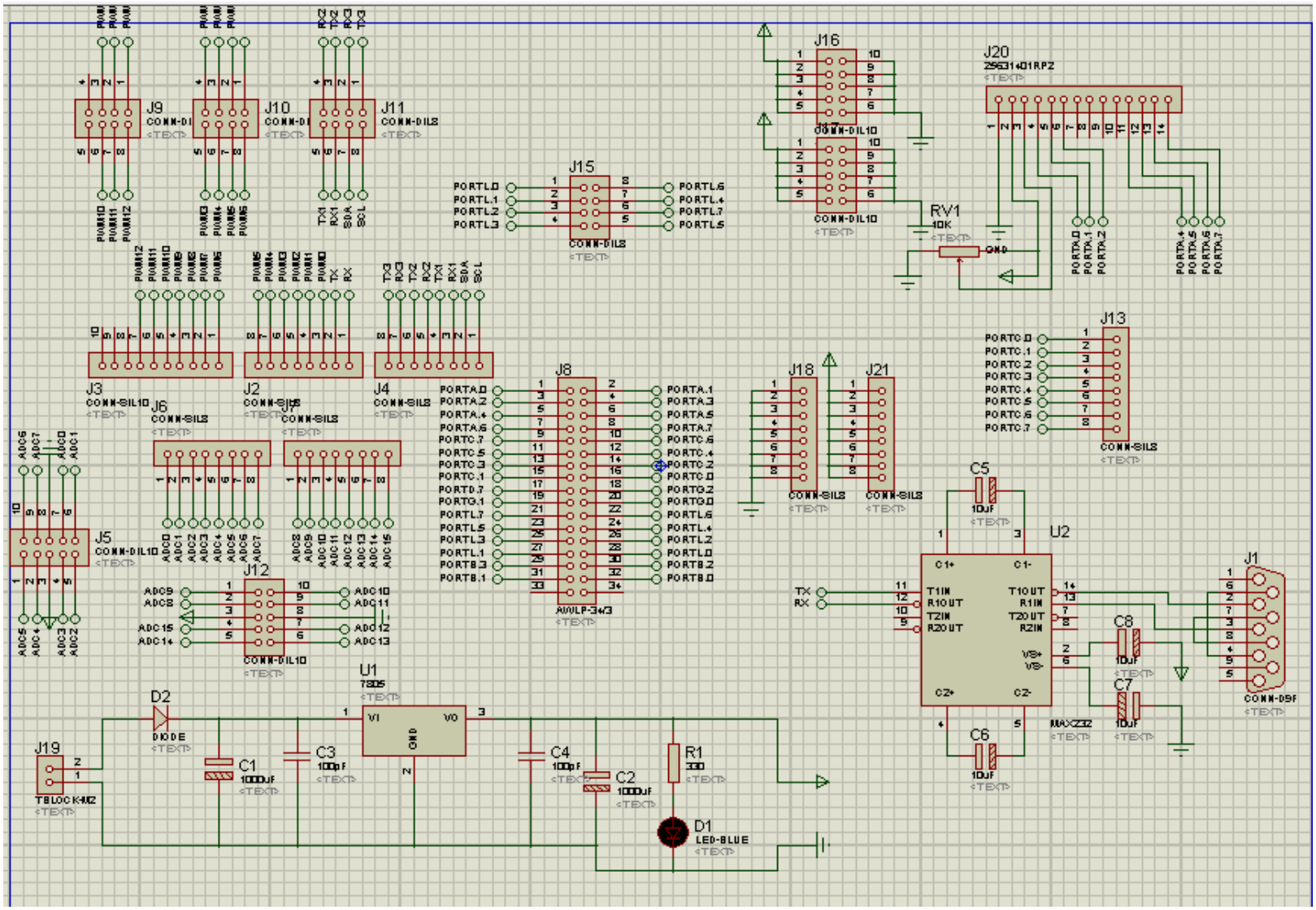
Gambar 22. Desain Base Telur dengan Kamera

2. PERANCANGAN ELEKTRONIK

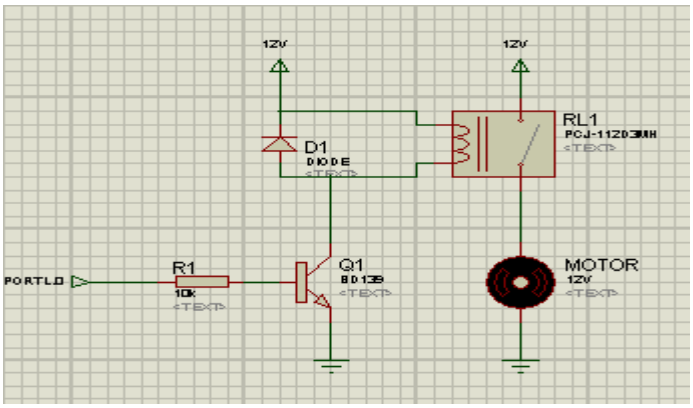
1) Minimum Sistem ATMEGA 2560

Sebuah mikrokontroler membutuhkan rangkaian minimum sistem agar dapat bekerja sesuai dengan kebutuhan dari sebuah piranti cerdas. Rangkaian minimum sistem yang dibuat oleh penulis memanfaatkan IC dari ATmega2560 sebagai controller utama dan beberapa komponen pembangkit sinyal frekuensi seperti oscillator/crystal, kapasitor, resistor dan beberapa pin yang salin terhubung disesuaikan dengan kebutuhan pengguna(*user*). ATmega2560 terdiri dari 12 PORT yang bisa difungsikan sebagai jalur input atau output. pada proyek akhir ini PORT dari ATmega2560(*Arduino mega),* penulis membuat sebuah rangkaian *shield* dari ATmega2560(*Arduino mega*) Gambar 23 merupakan rangkaian dari *shield* ATmega2560(arduino mega) :

Gambar 23. Rangkaian Shield ATmega2560



2) Driver Motor Relay 12V



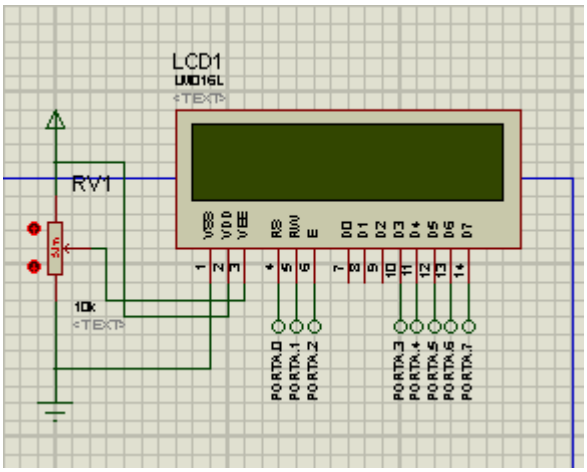
Gambar 24. Driver Motor Relay 12V

Gambar 24 merupakan rangkaian driver motor berbasis relay 12V untuk menghidupkan vakum, driver motor tersebut digunakan sebagai penguat atau *switching* untuk tegangan 12V yang akan dialirkan menuju ke motor dengan triger 5V dari mikrokontroller sehingga kapasitas tegangan yang dibutuhkan motor untuk berputar dapat terpenuhi dengan baik. Prinsip dari kerja driver motor relay tersebut seperti saklar, namun untuk dapat bekerja secara otomatis dibutuhkan tegangan input 5V dari mikrokontroller untuk menghidupkan coil dari relay tersebut sehingga suplai tegangan 12V dapat terpenuhi pada motor tersebut.

3) Piranti Antarmuka LCD

Piranti antarmuka LCD digunakan sebagai indikator kondisi dari program robot sehingga pengguna dapat melihat apakah robot bekerja sesuai dengan instruksi atau tidak, Ketika robot tidak bekerja sesuai intruksi pengguna dapat

melakukan instruksi reset pada mikrokontroller. Gambar 25 merupakan rangkaian dari piranti antarmuka LCD :



Gambar 25. Rangkaian Piranti Antarmuka LCD

4) Power Supply

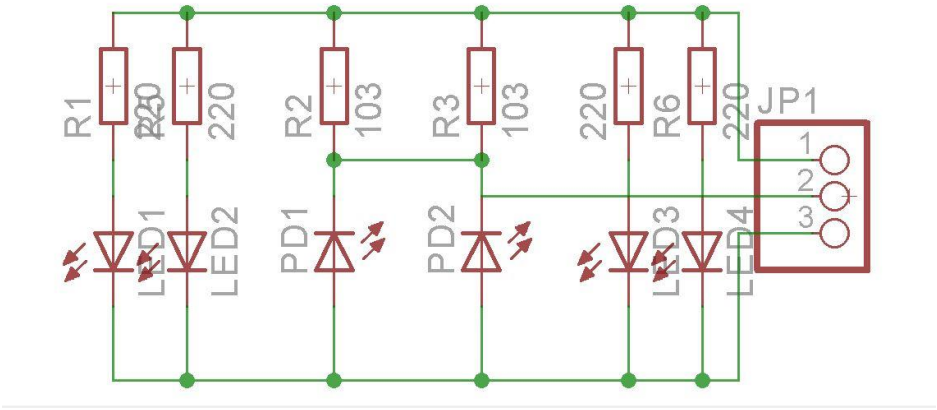
Sumber tegangan untuk robot tersebut menggunakan sebuah adaptor 12V dengan rentang beban 5A. Adaptor tersebut dipilih karena memiliki fitur kemampuan menahan arus yang cukup besar dengan suplai tegangan 12V yang tidak terlalu besar. Adaptor dibutuhkan untuk memberikan sumber daya untuk semua rangkaian, baik sensor servo maupun motor yang digunakan pada proyek akhir ini, namun pada rangkaian controller, servo dan sensor hanya membutuhkan sumber tegangan 5 V. Sehingga diperlukan penurun tegangan DC to DC konverter tegangan 5 V. Untuk dapat menkonversi tegangan tersebut digunakan modul komponen dari UBEC yang mampu menahan beban hingga 3A, cukup digunakan untuk mensuplai kinerja 6 servo dan minimum sistem.

Gambar 26. Adaptor 12V 5A

Gambar 27. UBEC 3A

5) Sensor Cahaya Photodiode

Sensor cahaya photodioda merupakan rangkaian yang terdiri dari sebuah photodioda dan sebuah resistor tetap yang dipasang secara seri sebagai sebuah pembagi tegangan. Hambatan yang terjadi pada photodiode akan memengaruhi jumlah arus yang masuk sehingga arus tersebut akan berubah sesuai dengan intensitas cahaya yang masuk pada photodiode. Rangkaian ini akan menghasilkan keluaran berupa tegangan analog yang berubah-ubah sesuai dengan pembacaan sensor. Pada proyek akhir ini, sensor cahaya photodiode digunakan untuk mendeteksi intensitas warna telur sebagai acuan apakah telur telah disortir dan terpindahkan oleh manipulator robot. Gambar 28 merupakan rangkaian dasar dari sensor cahaya photodiode :



Gambar 28. Rangkaian Sensor Cahaya Photodiode

3. PERANCANGAN PERANGKAT LUNAK

Secara konsep dasar dalam pengerjaan proyek akhir ini, penulis menggunakan tahapan tahapan untuk dapat merancang perangkat lunak, Antara lain seperti pembuatan *Graphical User Interface*(GUI) untuk mempermudah proses penggunaan, pemrograman pada pengolahan citra yang diintegrasikan dengan sistem cerdas berbasis logika fuzzy serta pemgrogaman pada cvavr untuk menunjang manipulator robot. Dalam perancangan perangkat lunak pembuatan *Graphical User Interface*(GUI) merupakan perancangan tambahan guna menunjang penggunaan agar lebih mudah dipahami.

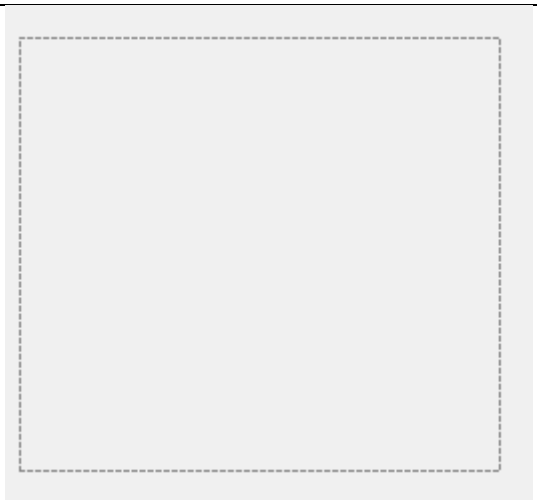
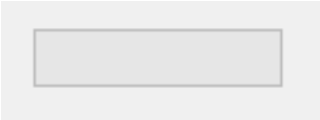
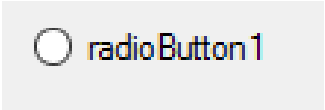
1) Pembuatan *Graphical User Interface*(GUI)

Pembuatan *Graphical User Interface(GUI)* merupakan suatu bagian fitur tambahan untuk mempermudah pengguna dalam mengoprasikan perangkat lunak dari protoype dan implementasi penyortir telur dengan logika fuzzy pada

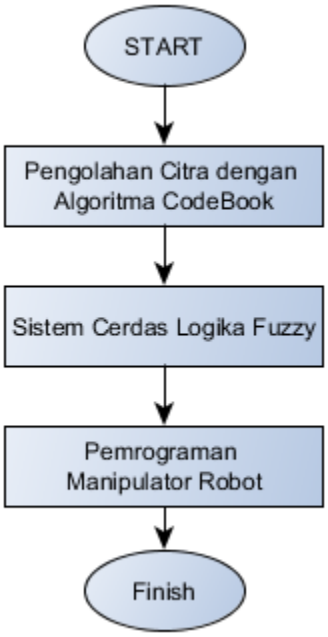
manipulator 6-*Degree of Freedom*. Pembuatan *Graphical User Interface(GUI)* direalisasikan pada pemgrograman visual berbasis bahasa C#, pemrograman visual berbasis bahasa C# lebih mudah untuk direalisasikan karena bahasa pemrograman tersebut hampir mirip dengan bahasa C pada struktur kalimat tiap programnya. Ada 2 mode yang akan dioperasikan pada *Graphical User Interface(GUI)*, yakni mode automatis dan mode simulasi. Mode Automatis lebih difungsikan untuk mempermudah pengguna agar penyortiran manual secara *real time* dapat dioperasikan. Mode manual ditujukan untuk mempermudah pengguna agar dapat melakukan simulasi logika fuzzy dengan hanya menginput panjang dan lebar secara manual, kemudian akan didapat hasil dari logika fuzzy apakah termasuk sangat kecil, kecil,sedang,besar atau sangat besar. Tabel 3 merupakan fungsi beberapa toolbox yang ada pada visual C#:

Tabel 3. Fungsi ToolBox Visual C#

No	TOOLBOX	KEGUNAAN
1.		Button toolbox : Sebagai tombol *trigger* pada umumnya dalam form visual c#.
2.		Combobox : Sebagai penadah *text* dalam jumlah besar.
3.		Label : Sebagai label tulisan pada form visual C#.

4.		Picture box : Sebagai tools untuk menampilkan gambar pada form visual c#.
5.		Progress bar : sebagai indikator prosess dalam form visual C#
6.		Radio button : Sebagai menu pilihan pada form visual C#
7.		Text Box : Sebagai wadah untuk menulis *text.*

Setelah pembuatan graphical user interface(GUI), proses selanjutnya adalah mengolah beberapa fungsi inti dari perancangan perangkat lunak pengerjaan proyek akhir ini. Gambar 29 merupakan flowchart singkat mengenai fungsi inti dari tahapan perancangan perangkat lunak :



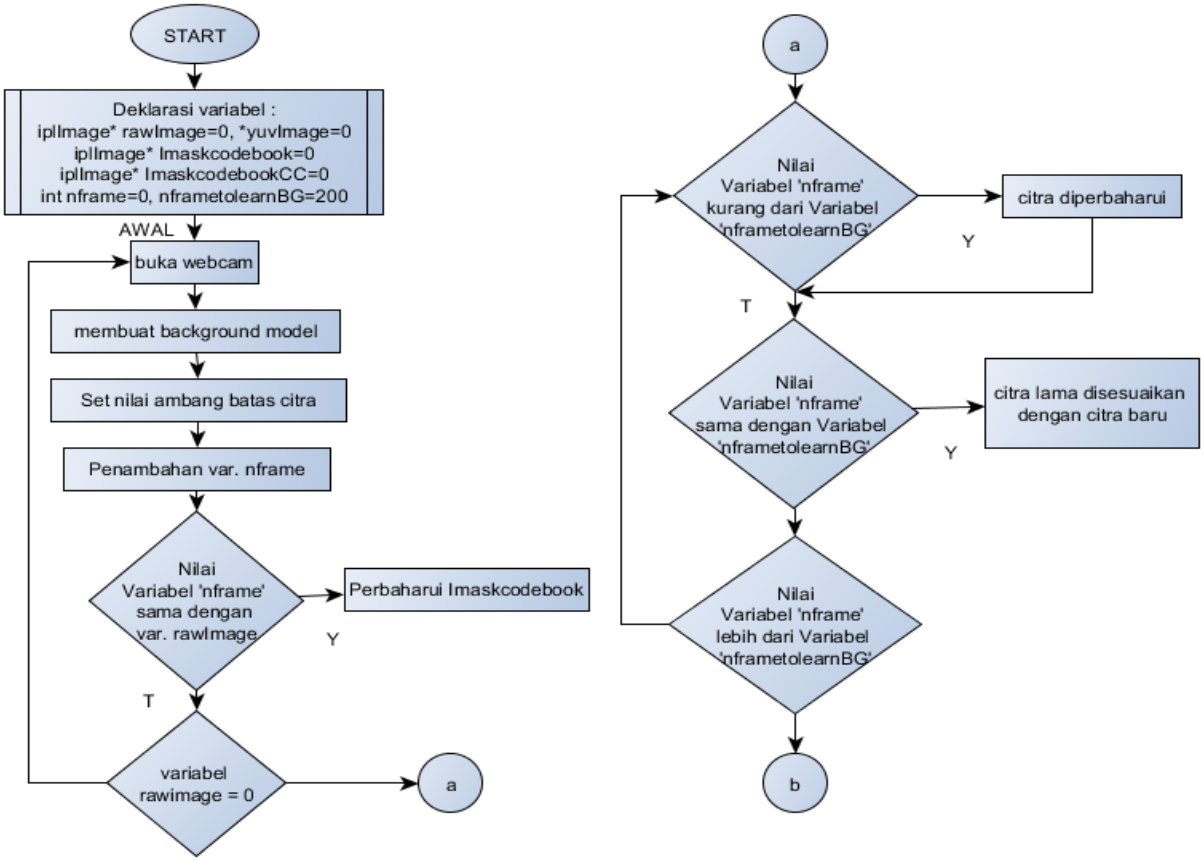
Gambar 29. FlowChart Singkat Perancangan perangkat lunak

2) Pengolahan Citra

Pengolahan citra merupakan suatu bagian yang sangat penting dalam pengerjaan proyek akhir ini, pengolahan citra ini memnfaatkan opencv sebagai *libray* utama yang ditujukan untuk membangun suatu basis pengelihatan robot secara real time. Oleh sebab itu pengolahan citra merupakan bagian yang terpenting karena dari pengolahan citra ini, penulis dapat membangun program logika fuzzy yang terintegrasi dengan pengolahan citra,

Metode yang digunakan untuk pengolahan citra adalah CodeBook algoritma dimana codebook algoritma merupakan suatu metode untuk membuat citra yang diolah menjadi adaptive dengan lingkungannya. Pada prinsipnya codebook algoritma mempelajari lingkungan sekitarnya, tujuan untuk mempelajari

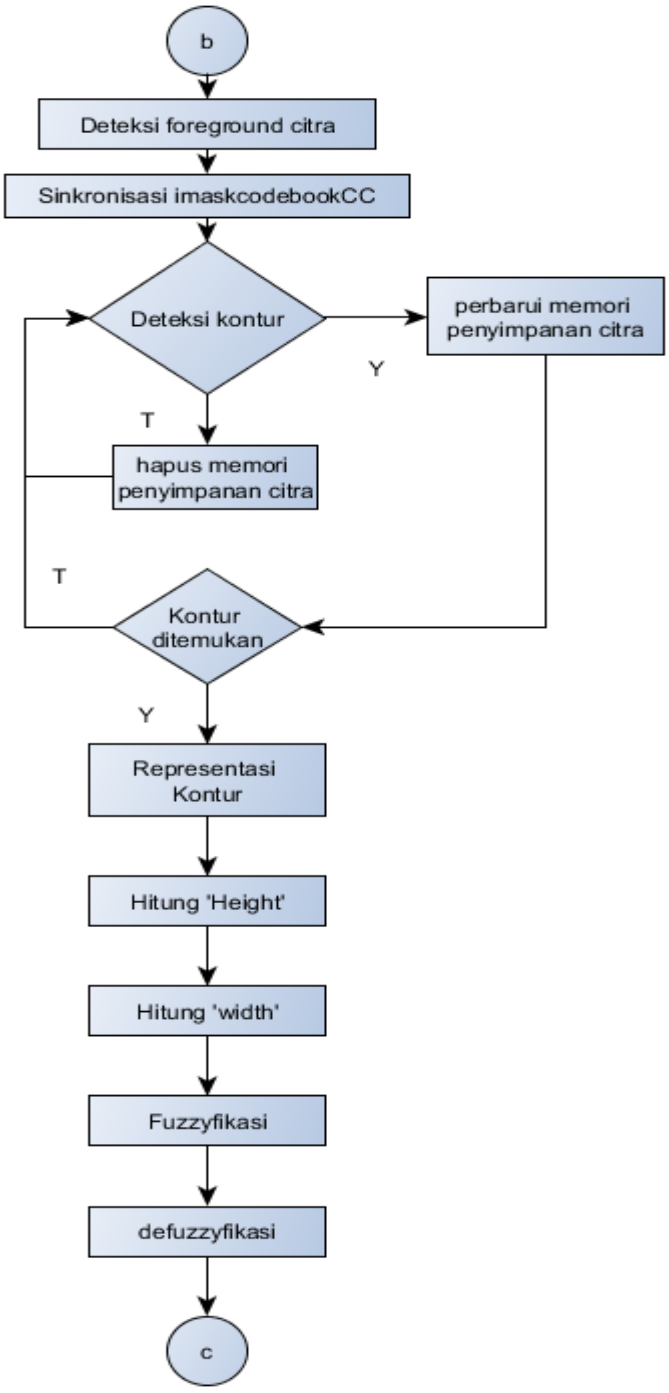
lingkungannya adalah untuk mementukan nilai ambang batas atau threshold dari citra yang telah dipelajari, ketika proses learning telah selesai dan background citra telah dirubah menjadi threshold maka selanjutnya adalah memasukkan suatu object yang dinilai asing untuk diambil nilai biner dari citra asing tersebut, pada proyek akhir kali ini citra asing yang dimakasud adalah telur yang akan dicari panjang dan lebarnya. Dari hasil tersebut dapat diketahui berapa ukuran object asing(telur) yang masuk tersebut, ketika data pengukuran panjang dan lebar telah didaptkan selanjutnya adalah memfuzzyfikasikan data tersebut lalu hasil dari fuzzyfikasi tersebut di defuzzyfikasikan lagi untuk diketahui hasilnya secara pasti sehingga akan lebih mempermudah menentukan paramater output keluaran apakah termasuk sangatkecil, kecil,sedang,besar atau sangat besar. Gambar 30 merupakan flowchart dari pengolahan citra tersebut :



Gambar 30. Bagian Awal Flowchart CodeBook Algoritma

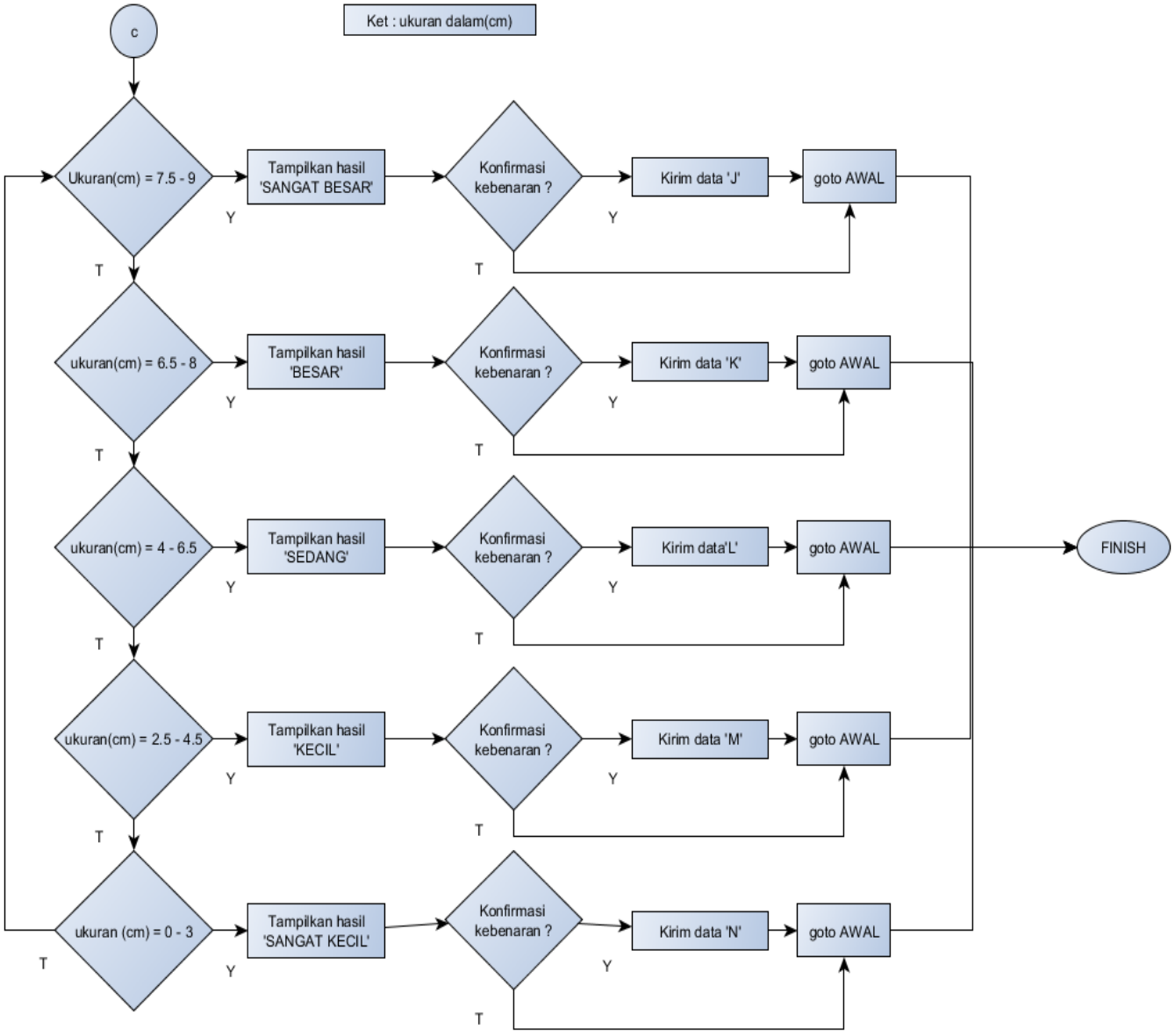
Pada bagian awal algoritma codebook, dibutuhkan beberapa variabel untuk mengakses Codebook dari library opencv yang kemudian dideklarasikan untuk membuka window agar akses webcam terpenuhi dengan window yang telah dibuka tersebut, kemudian ketika webcam telah menampilkan citra langkah selanjutnya adalah mengolah dari citra tersebut untuk dibuat background modelnya lalu mensetting nilai ambang batas secara otomatis dengan kondisi yang telah dibuat pada program. Proses selanjutnya adalah menampung variabel program dalam sebuah variabel, variabel tersebut dinamakan dengan ImaskCodeBookCC yang berfungsi sebagai ilustrator untuk menggambar daerah yang dianggap asing(telur)

menjadi daerah yang terdeteksi, gambar 31 merupakan flowchart dan gambar 35 merupakan lanjutan dari potongan program diatas :



Gambar 31. Bagian Tengah Flowchart CodeBook Algoritma

Langkah selanjutnya adalah mengolah dari hasil output CodeBook Algoritma tersebut yang berupa panjang dan lebar untuk di cari nilai fuzzynya dari proses fuzzyfikasi, rule dan defuzzyfikasi. Gambar 32 merupakan Bagian Akhir Program dan Flowchart CodeBook Algoritma :



Gambar 32. Bagian Akhir Program dan Flowchart CodeBook Algoritma

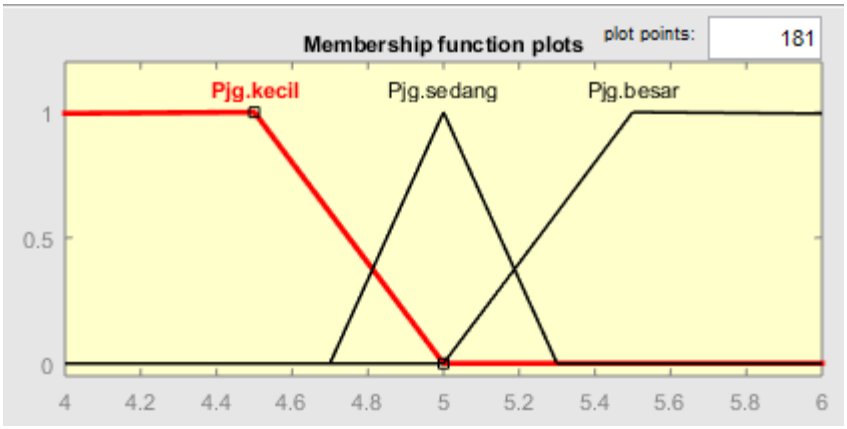
3) Logika Fuzzy

1. Fuzzyfikasi

Fuzzyfikasi adalah suatu metode yang digunakan untuk untuk mengolah suatu masukan dari bentuk tegas (crisp) menjadi fuzzy (variabel linguistik), pada dasarnya disajikan berupa bentuk himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing dari masukan(input). Pada proyek akhir ini bagian dari variabel Input diabgi menjadi 2 variabel yaitu variabel Panjang dan variabel Lebar. Variabel Panjang mempunyai 3 buah linguistik value yaitu Kecil, Sedang dan Besar, begitu juga untuk Variabel Lebar mempunyai 3 buah linguistik value yaitu Kecil, Sedang dan Besar. Pada bagian Output terdapat variabel 5 buah linguistik value yaitu SangatKecil, Kecil, Sedang, Besar dan SangatBesar.

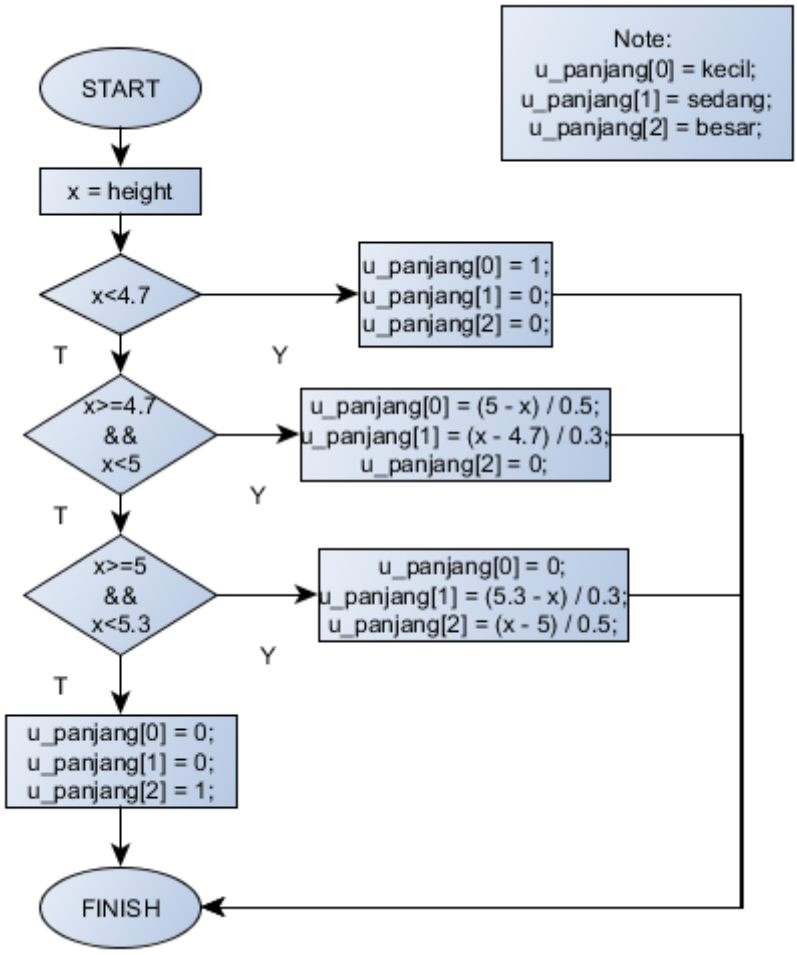
a. Variabel Panjang

Variabel Panajng dibuat dengan range pengukuran mulai dari 4 sampai dengan 6. Nilai ini diperoleh dari mencari rata-rata panjang maksimal dan panjanng minimal dari telur yang dikategorikan sangat besar dan sangat kecil oleh *judgment expert*. Gambar 38 merepresentasikan nilai untuk tiap fungsi keanggotaan Input Panjang. Pada sistem Fuzzy tersebut, dijabarkan menjadi 3 buah membership function yaitu Pjg.kecil, Pjg.sedang dan Pjg.besar,.



Gambar 33. Fungsi Keanggotaan Input Panjang

Gambar 33 adalah implementasi sistem logika fuzzy pada matlab, namun pada proyek akhir ini penulis mengimplementasikan logika fuzzy pada pemrograman bahasa C, yang mana itu artinya setiap persamaan untuk menentukan derajat keanggotaan di tulis secara manual pada worksheet bahasa C, Gambar 34 merupakan algoritma untuk menentukan nilai dari fuzzyfikasi input Panjang :



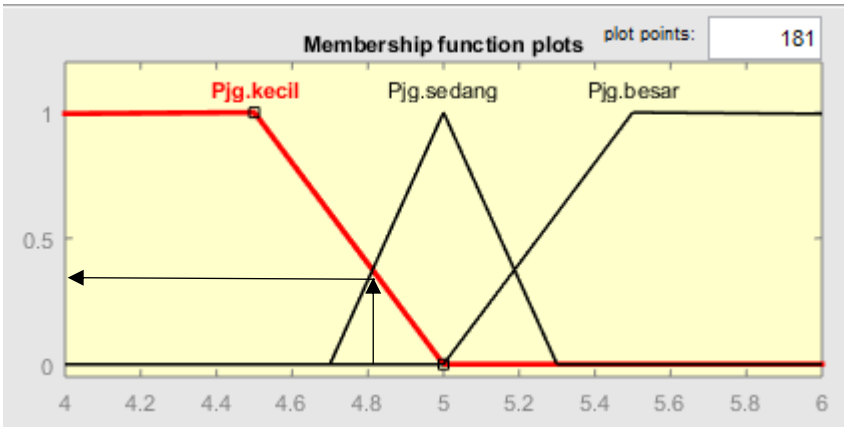
Gambar 34. Flowchart Fuzzyfikasi Variabel Panjang

Variabel nilai x diinputkan dengan variabel height, variabel height diperoleh dari pengolahan citra sebelumnya untuk menentukan panjang dari citra telur. Setelah kedua variabel berhasil disandingkan kemudian variabel x dibandingkan dengan kondisi minimal, sedang, dan maksimal yakni 0, 4 dan 8 untuk menentukan nilai dari derajat keanggotaannya.

Variabel x adalah nilai dari variabel height yang mana kemudian diolah sehingga diperoleh. u_panjang[0] sebagai variabel anggota Pjg.Kecil, u_panjang[1] sebagai variabel anggota Pjg.sedang dan u_panajng[2] sebagai variabel anggota Pjg.besar. Terdapat empat kondisi yang akan menghasilkan nilai derajat keanggotaan untuk masing-masing anggota dari variabel pajang. Berikut ini adalah contoh kasus untuk menentukan fuzzyfikasi pada input Panjang.

Contoh :

Pada saat citra telur diolah kemudian didapatkan hasil dari olahan citra tersebut, kita asumsikan bahwa citra telah mendeteksi panjang dari telur sebesar 4.8 (dalam centi), maka bisa dilihat pada gambar 35 bahwa 4.8 masuk pada kondisi



Gambar 35. Contoh kondisi fuzzyfikasi Panjang

Dapat dilihat bahwa 4.8 tepat berada pada perpotongan antara Pjg. Kecil dan Pjg.sedang, jika kita kalkulasikan secara manual akan didaptkan seperti berikut ini:

X = 4.8

u_kecil = (d - x) / (d - c)

= (5 – 4.8) / (5 – 4.5) = 0.2 / 0.5 = 0.4 dan

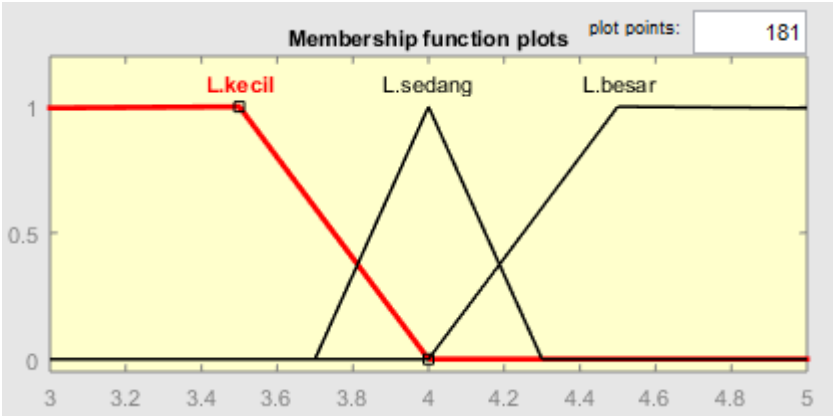
u_sedang = (x - a) / (b - a)

= (4.8 – 4.7) / (5 – 4.7) = 0.1 / 0.3 = 0.34

Berdasarkan data diatas dapat disimpulkan bahwa 4.8 memiliki derajat keanggotaan sebesar 0.4 pada fungsi keanggotaan Pjg.kecil dan 0.34 pada fungsi keanggotaan Pjg.sedang.

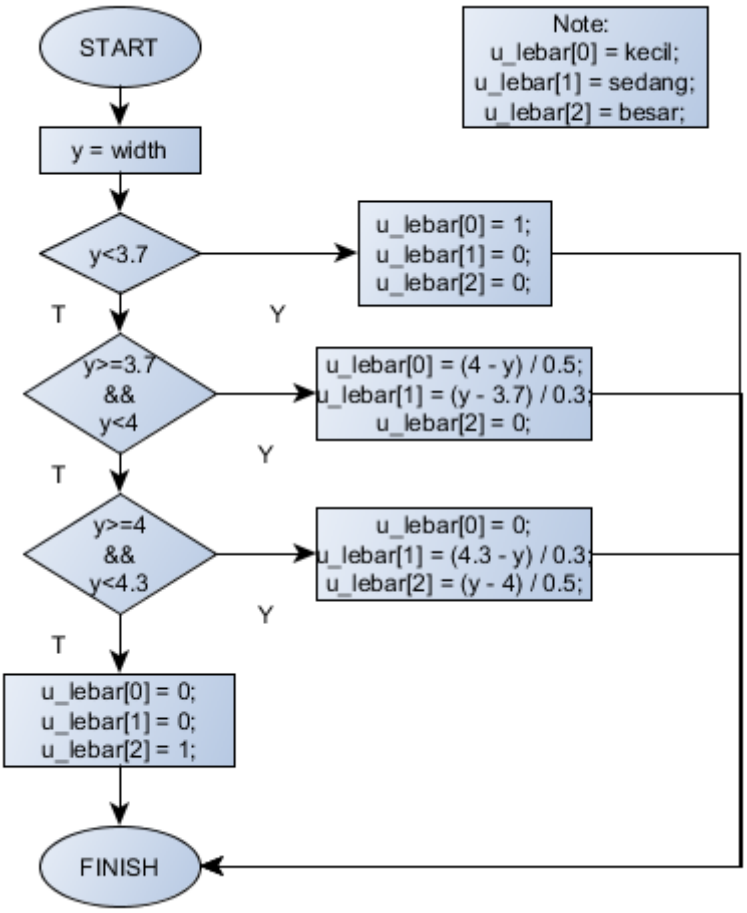
b. Variabel Lebar

Variabel Lebar juga dibuat dengan range yang sama dengan Variabel panjang yakni pengukuran mulai dari 0 sampai dengan 8. Nilai tersebut juga diperoleh dari mencari rata-rata Lebar maksimal dan Lebar minimal dari telur yang dikategorikan sangat besar dan sangat kecil oleh *judgment expert*. Gambar 36 merupakan ilustrasi dari input variabel Lebar yang dijabarkan menjadi 3 buah membership function yaitu L.kecil, L.sedang dan L.besar.



Gambar 36. Fungsi Keanggotaan Input Lebar

Gambar diatas adalah implementasi sistem logika fuzzy pada matlab, pada proyek akhir ini penulis juga mengimplementasikan logika fuzzy pada pemrograman bahasa C. Gambar 37 merupakan algoritma untuk menentukan nilai dari fuzzyfikasi input Lebar :



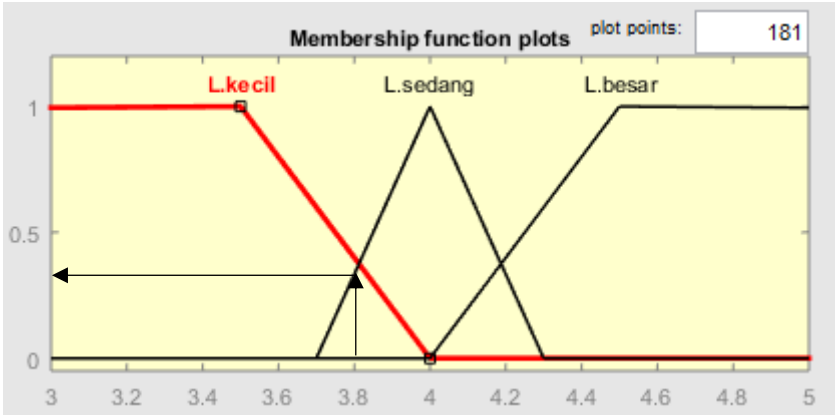
Gambar 37. Flowchart Fuzzyfikasi Lebar

Alur pemrograman hampir sama dengan alur untuk mencari derajat keanggotaan input variabel Panjang, Variabel nilai y diinputkan dengan variabel width, variabel width juga diperoleh dari pengolahan citra sebelumnya untuk menentukan panjang dari citra telur. Setelah kedua variabel berhasil disandingkan

kemudian variabel y dibandingkan dengan kondisi minimal, sedang, dan maksimal yakni 3, 4 dan 5 untuk menentukan nilai dari derajat keanggotaannya. Variabel y adalah nilai dari variabel width yang mana kemudian diolah sedemikian rupa sehingga diperoleh. u_lebar[0] sebagai variabel anggota L.Kecil, u_lebar[1] sebagai variabel anggota L.sedang dan u_lebar[2] sebagai variabel anggota L.besar. untuk menentukan nilai derajat keanggotaan dari variabel Lebar terdapat empat kondisi yang akan menghasilkan nilai derajat keanggotaan untuk masing-masing anggota dari variabel lebar. Berikut ini adalah contoh kasus untuk menentukan fuzzyfikasi pada input lebar.

Contoh :

Asumsikan bahwa citra telah mendeteksi lebar dari telur sebesar 3.8 (dalam centi), maka bisa dilihat bahwa 3.8 masuk pada kondisi



Gambar 38. Contoh kondisi fuzzyfikasi lebar

Dapat dilihat bahwa 3.8 berada pada perpotongan antara L. Sedang dan L.kecil, jika kita kalkulasikan secara manual akan didaptkan seperti berikut ini:

Y = 3.8

u_kecil = (d - x) / (d - c)

= (4 – 3.8) / (4 – 3.5) = 0.2 / 0.5 = 0.4 dan

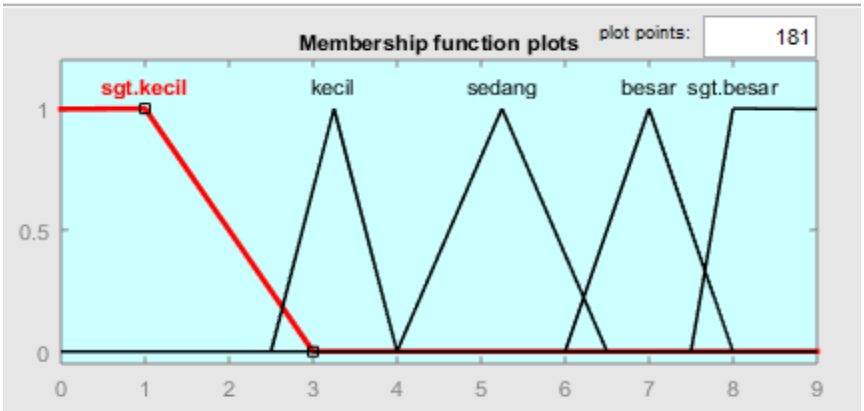
u_sedang = (x - a) / (b - a)

= (3.8 – 3.7) / (4 – 3.7) = 0.1 / 0.3 = 0.34

Berdasarkan data diatas dapat disimpulkan bahwa 3.8 memiliki derajat keanggotaan sebesar 0.4 pada fungsi keanggotaan L.sedang dan 0.34 pada fungsi keanggotaan L.Besar.

c. Variabel Ukuran

Variabel Ukuran merupakan hasil akhir atau output dari perhitungan variabel Panjang dan Variabel Lebar, Variabel ukuran memiliki range antara 0 – 9. Variabel Ukuran memiliki 5 buah *linguistik value* diantaranya SangatKecil, Kecil, Sedang, Besar dan SangatBesar, gambar 39 merupakan ilustrasi dari variabel output :



Gambar 39. Fungsi Keanggotaan Variabel Keluaran

2. *Fuzzy Rule*

Fuzzy Rule merupakan sebuah aturan yang dibuat untuk memenuhi syarat dari sebuah output yang terjadi, Aturan yang digunakan pada proyek akhir ini berdasarkan dari *judgement expert* di Tempat penyortiran telur, aturan tersebut di ilustrasikan pada tabel berikut ini :

Tabel 4. *Fuzzy Rule*

Panjang / Lebar	L.Kecil	L.Sedang	L.Besar
Pjg.Kecil	SangatKecil (R1)	Kecil (R2)	Sedang (R3)
Pjg.Sedang	Kecil (R4)	Sedang (R5)	Besar (R6)
Pjg.Besar	Sedang (R7)	Besar (R8)	SangatBesar (R9)

Berdasarkan Tabel 3. *Fuzzy Rule* diatas, dapat dijabarkan rule yang akan digunakan dalam pengerjaan proyek akhir ini, diantaranya :

R1) IF Panjang is Pjg.Kecil AND Lebar is L.Kecil THEN Ukuran is SangatKecil

R2) IF Panjang is Pjg.Kecil AND Lebar is L.Sedang THEN Ukuran is Kecil

R3) IF Panjang is Pjg.Kecil AND Lebar is L.Besar THEN Ukuran is Sedang

R4) IF Panjang is Pjg.Sedang AND Lebar is L.Kecil THEN Ukuran is Kecil

R5) IF Panjang is Pjg.Sedang AND Lebar is L.Sedang THEN Ukuran is Sedang

R6) IF Panjang is Pjg.Sedang AND Lebar is L.Besar THEN Ukuran is Besar

R7) IF Panjang is Pjg.Besar AND Lebar is L.Kecil THEN Ukuran is Sedang

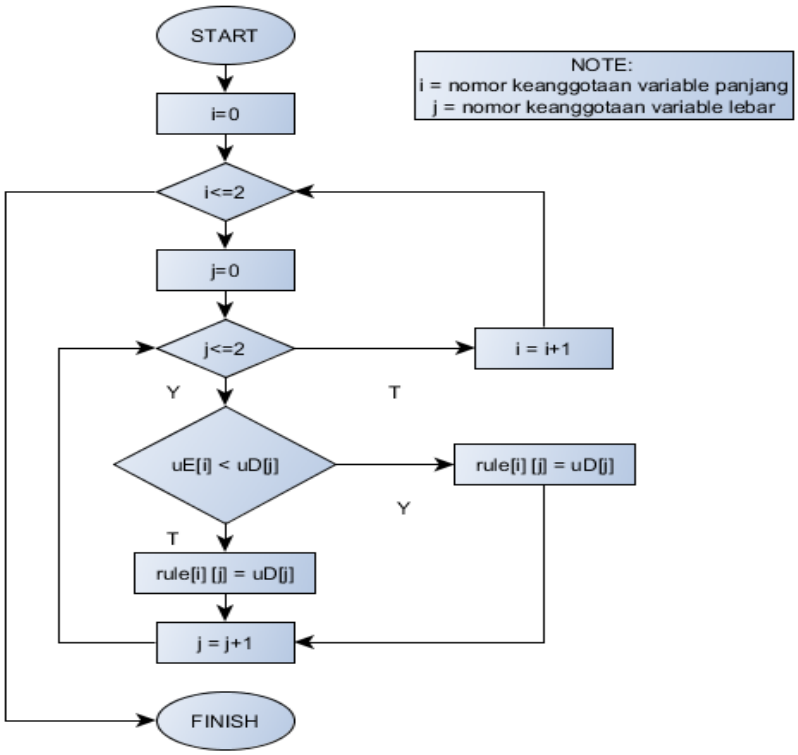
R8) IF Panjang is Pjg.Besar AND Lebar is L.Sedang THEN Ukuran is Besar

R9) IF Panjang is Pjg.Besar AND Lebar is L.Besar THEN Ukuran is SangatBesar

Setelah basis rule telah disusun sedemikian rupa, kemudian menentukan fungsi implikasi dari tiap rule, pada pengerjaan proyek akhir ini penulis menggunakan metode mamdani dalam membangun sistem cerdas logika fuzzy. Aturan pada metode mamdani untuk fungsi implikasi adalah dengan menggunakan fungsi MIN dengan AND sebagai operator untuk menentukan nilai keaggotaan terkecil, berikut merupakan ilustrasi dari persamaan fungsi MIN dengan operator AND:

$$\mu A \cap B = \min(\mu A[x], \mu B[y])$$

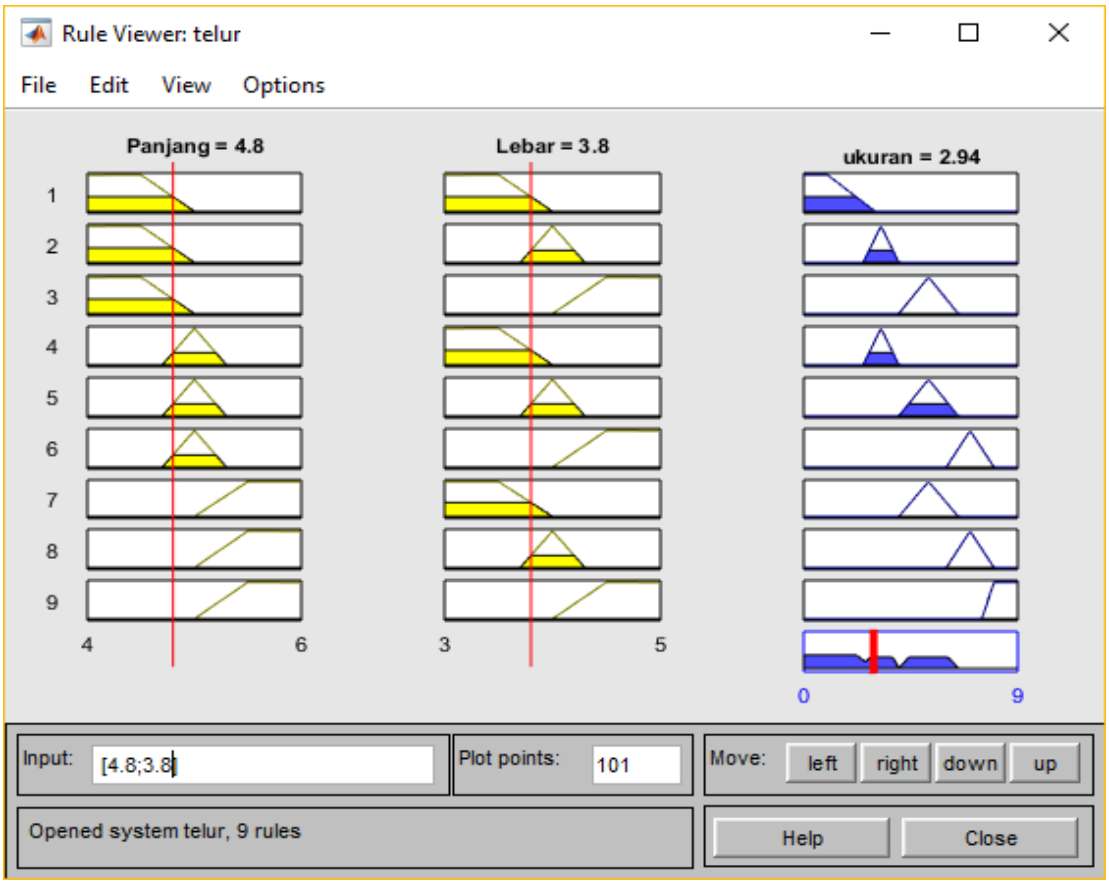
Gambar 44 merupakan flowchart dari aturan fuzzy yang akan diberlakukan pada pengerjaan proyek akhir ini :



Gambar 40. Flowchart *Fuzzy Rule*

Dari hasil proses fuzzyfikasi sebelumnya diperoleh nilai tiap keanggotaan dari $\mu E[\,i\,]$ dan $\mu D[\,j\,]$. Dengan menggunakan fungsi implikasi berupa MIN dan operator AND maka yang diambil berdasarkan metode mamdani adalah $MIN\ (\mu E[\,i\,] <\ \mu D[\,j\,] = rule[\,i\,][\,j\,] =\ \mu E[\,i\,])$. Pada pengaplikasian bahas C aturan fuzzy di *looping* dengan tujuan untuk memeriksa apakah 9 rule tersebut telah memenuhi syarat dari metode mamdani. Tiap *rule* akan membandingkan nilai derajat keanggotaan dari variabel Panjang dan variabel Lebar, untuk subrutin program diatas u_Panjang merupakan anggota variabel Panjang dan u_Lebar merupakan anggota dari variabel Lebar, dengan metode min maka nilai yang paling terkecil diatara tiap rule akan diambil untuk di defuzzyfikasikan agar dapat dlihat keluarannya. Contoh kasus untuk memahami subrutin dari program rule fuzzy:

Asumsi dari fuzzyfikasi sebelumnya, Panjang = 4.8 dan Lebar = 3.8, bila dilustrasikan dalam matlab akan tanpa seperti dibawah ini,



Gambar 41. Ilustrasi Hasil Berdasarkan *Fuzzy Rule*

Pada gambar 41 dapat kita lihat bahwa rule yang memenuhi dari fuzzyfikasi variabel Panjang = 4.8 dan variabel Lebar = 3.8 adalah Rule nomor R1,R2,R4 dan R5. itu artinya bila diimplemetasikan secara perhitungan manual kelima rule yang kosong dapat diabaikan sehingga penulisan dari ke4 rule yang memenuhi syarat akan menjadi seperti :

R1) IF Panjang is Pjg.Kecil AND Lebar is L.Kecil THEN Ukuran is SangatKecil

α – predikat 1 = μPanjang ∩ μLebar

= min(μPjg.Kecil [3], μL.kecil[5])

= min(0.4 ; 0.4) = 0.4

R2) IF Panjang is Pjg.Kecil AND Lebar is L.Sedang THEN Ukuran is Kecil

α – predikat 2 = μPanjang ∩ μLebar

= min(μPjg.Kecil [3], μL.sedang[5])

= min(0.4 ; 0.34) = 0.34

R4) IF Panjang is Pjg.Sedang AND Lebar is L.Kecil THEN Ukuran is Kecil

α – predikat 5 = μPanjang ∩ μLebar

= min(μPjg.Sedang [3], μL.kecil[5])

= min(0.34 ; 0.4) = 0.34

R5) IF Panjang is Pjg.Sedang AND Lebar is L.Sedang THEN Ukuran is Sedang

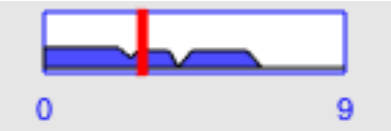
α – predikat 6 = μPanjang ∩ μLebar

= min(μPjg.Sedang [3], μL.sedang[5])

= min(0.34 ; 0.34) = 0.34

Setalah hasil perhitungan dari tiap rule diatas dengan memenuhi kriteria MINIMAL, selanjutnya adalah melakukan menkomposisikan tiap rule diatas

menjadi satu dengan menggunakan metode MAX, hasilnya dapat diilustrasikan pada gambar 42:



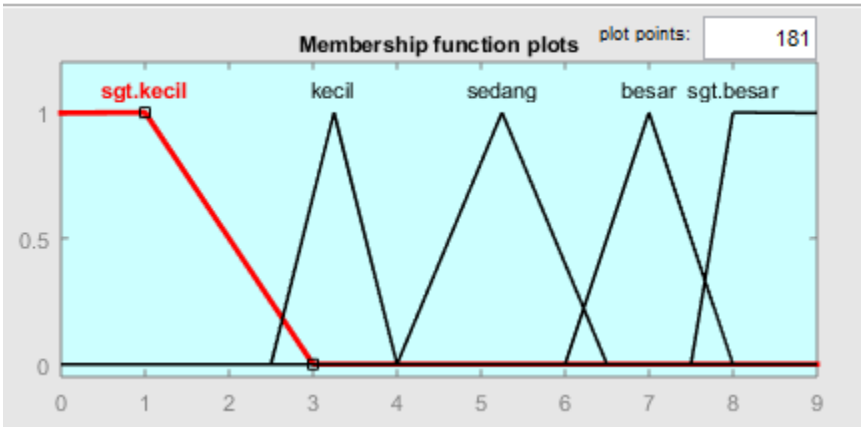
Gambar 42. Hasil Komposisi *Fuzzy Rule*

3. Defuzzyfikasi

Defuzzifikasi merupakan suatu penegasan ulang tiap himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, setelah komposisi tiap aturan ditegaskan ulang kemudian akan diperoleh suatu output yang mana output merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Contoh Jika diberikan suatu himpunan fuzzy dalam range tertentu, maka dari range tersebut dapat di ambil suatu nilai *crisp* sebagai output. Pada proyek akhir ini metode defuzzyfikasi yang digunakan adalah dengan menggunakan metode centroid atau juga dikenal dengan Center of Area (COA). Alasan digunakannya metode centroid ini adalah dengan cara mengambil titik tengah dari tiap daerah fuzzy. Output dari fuzzy yang diharapkan adalah nilai pasti dari ukuran fisik telur.

Dalam pengolahan defuzzyfikasi pada variabel output, Variabel output dibagi menjadi 5 bagian yang tercantum dalam Variabel ukuran dengan fungsi keanggotaan linguistik value yaitu SangatKecil, Kecil, Sedang, Besar dan

SangatBesar dengan range nilai berkisar antara 0 - 9. Gambar 43 merupakan ilustrasi dari Variabel Ukuran pada matlab :



Gambar 43. Fungsi Keanggotaan Variabel Ukuran

Dengan mengacu pada metode centroid untuk penegasan ulang defuzzyfikasi yang artinya titik tengah dari setiap fungsi keanggotaan harus dicari terlebih dahulu, untuk mencari nilai tengah dari fungsi keanggotaan Kecil, Sedang dan Besar, digunakan persamaan luas segitiga karena ketiga dari fungsi keanggotaan tersebut berbentuk sama yakni segitiga. Contoh perhitungannya dapat kita ilustrasikan seperti berikut ini :

Ambil contoh pada fungsi keanggotaan ‘Sedang’, untuk mencari nilai tengahnya kita hitung luas daerah dengan cara mengkalikan alas dengan tinggi lalu dibagi 2 setelah itu ditambah dengan rang nilai minimum dari fungsi keanggotaan Sedang yakni 3,

Panjang alas = 6.5 – 4 = 2.4

Tinggi = 1

Luas daerah $= \frac{a\ x\ t}{2} = \frac{2.4\ x\ 1}{2} = 1.2$

Nilai tengah = 4 + 1.2 = 5.2

Untuk fungsi keanggotaan dari Kecil dan Besar, dapat dilakukan perhitungan dengan cara yang sama seperti mencari luas daerah dan titik tengah dari ‘sedang’ sehingga dapat diilustrasikan sebagai berikut:

- Kecil

  Panjang alas = 4 – 2.5 = 1.5

  Tinggi = 1

  Luas daerah $= \frac{a\ x\ t}{2} = \frac{1.5\ x\ 1}{2} = 0.75$

  Nilai tengah = 2.5 + 0.75 = 3.25

- Besar

  Panjang alas = 8 – 6 = 2

  Tinggi = 1

  Luas daerah $= \frac{a\ x\ t}{2} = \frac{2\ x\ 1}{2} = 1$

  Nilai tengah = 6 + 1 = 7

Untuk mecari COA dari trapesium dapat dilakukan dengan menggunakan rumus ½ x (a + c) x t,

Contoh :

Untuk menentukan COA dari membership function dari sangat kecil dapat dilakukan dengan memasukkan persamaan diatas dengan sisi a = 0 dan sisi c = 2 dan t = 1, menjadi:

= ½ x (0 + 2) x 1

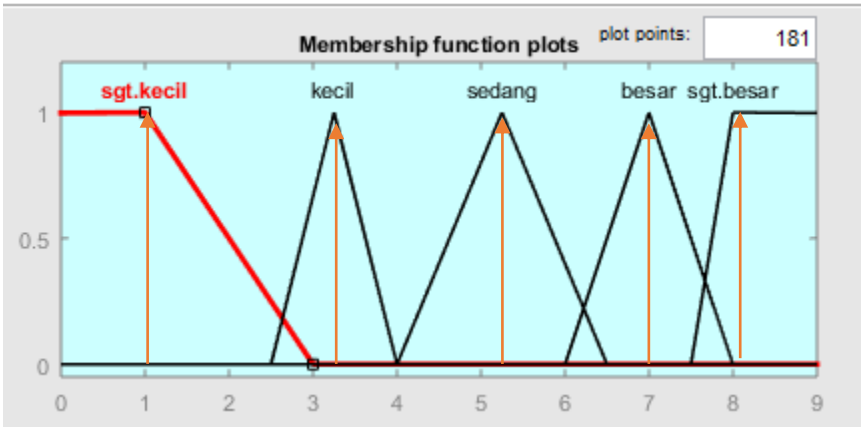
= 1

Dengan demikin nilai tengah dari keanggotaan sangat kecil adalah 1. Untuk menentukan nilai tengah dari keanggotaan sangat besar, caranya sama dengan mecari keanggotaan kecil yakni memasukkan persamaan diatas dengan sisi a = 7, sisi c = 9, t = 1, jika dihitung dengan perssamaan mencari nilai tengah trapesium seperti diatas, maka akan menjadi

= ½ x (7 + 9) x 1

= 8

Secara keluruhan dapat diilustrasikan nilai tengah dari tiap fungsi keanggotaan variabel Ukuran pada gambar 44:



Gambar 44. Nilai Tengah dari Tap Fungsi Keanggotan Variabel Ukuran

Data lengkap dapat dilihat pada tabel 5 :

Tabel 5. Data Luas dan Titik Tengah

Anggota	Luas		Titik Tengah(coa)	
SangatKecil	Z1	1	A1	1
Kecil	Z2	0.75	A2	3.25
Sedang	Z3	1.2	A3	5.2
Besar	Z4	1	A4	7
SangatBesar	Z5	1	A5	8

Selanjutnya setelah data Luas dan Titik Tengah terkumpul, menentukan nilai dari output yang diharapakan dengan menggunakan persamaan :

$$Output = \frac{\sum_{i=1}^{n} \text{rule(i) x coa(i) x luas(i)}}{\sum_{i=1}^{n} \text{rule(i) x luas(i)}}$$

Keterangan :

Rule = Fungsi implikasi fuzzyfikasi

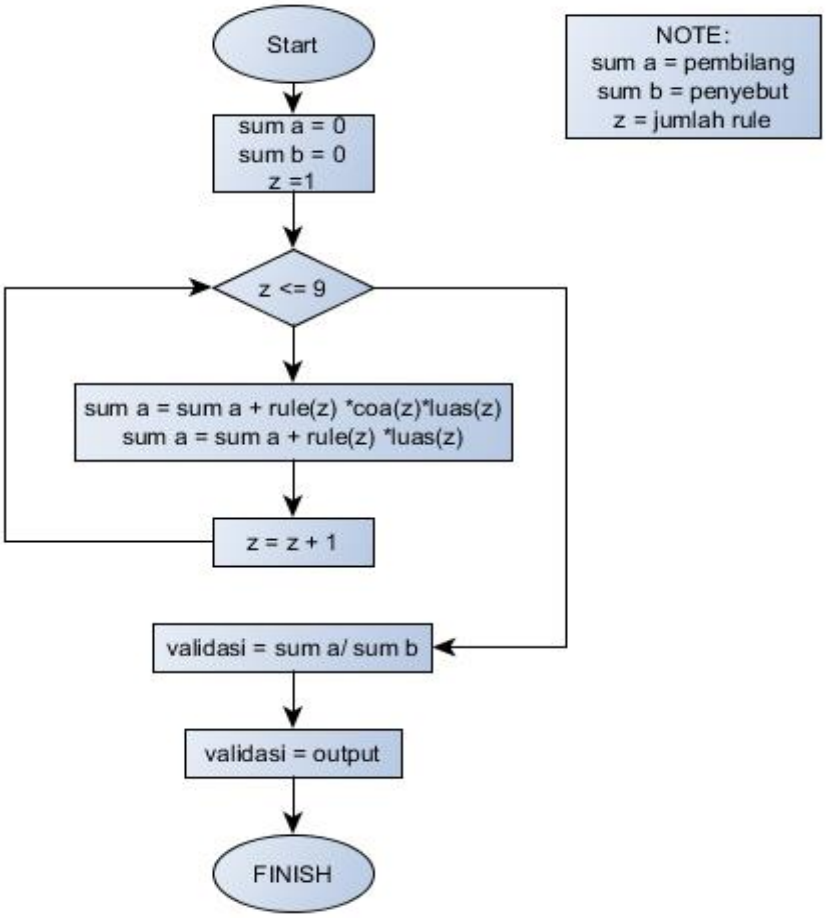
COA = Titik tengah fungsi keanggotaan variabel ukuran

Luas = Luas daerah keanggotaan variabel ukuran

n = Jumlah rule yang digunakan

i = data ke-n

Gambar 45 merupakan flowchart dan subrutin program defuzzyfikasi yang digunakan untuk penyelesaian proyek akhir ini :



Gambar 45. Flowchart Defuzzyfikasi

Hasil dari perhitungan fuzzyfikasi dan defuzzyfikasi diatas merupakan hasil mutlak dari pengukuran ciri fisik telur dengan kriteria telur termasuk Sangat kecil, Kecil, Sedang, Besar atau Sangat besar. Contoh kasus dapat diilustrasikan sebagai berikut :

Dengan mengacu contoh kasus sebelumnya, untuk mencari hasil output atau nilai defuzzyfikasinya diperoleh data fungsi implikasi *input* untuk masing-masing rule, tabel 6 merupakan hasil fungsi implikasi tiap rule :

Tabel 6. Hasil fungsi Implikasi Fuzzy Rule

Rule	Fungsi Implikasi		Output
R1	W1	0.4	Sangat Kecil
R2	W2	0.34	Kecil
R3	W3	0	-
R4	W4	0.34	Kecil
R5	W5	0.34	Sedang
R6	W6	0	-
R7	W7	0	-
R8	W8	0	-
R9	W9	0	-

Setelah data tiap rule telah terkumpul, langkah selanjutnya adalah mencari nilai output, namun pada contoh diatas untuk R1, R4, R7, R8 dan R9 bernilai 0, maka perhitungannya dapat diabaikan. Berikut Ilustrasi dari perhitungan fungsi implikasi Rule :

$$Output = \frac{\sum_{i=1}^{n} rule(i)\ x\ coa(i)\ x\ luas(i)}{\sum_{i=1}^{n} rule(i)\ x\ luas(i)}$$

$$= \frac{\sum_{i=1}^{9} rule(i)\ x\ coa(i)\ x\ luas(i)}{\sum_{i=1}^{9} rule(i)\ x\ luas(i)}$$

$$= \frac{(w1xa1xz1)+(w2xa2xz2)+(w4xa4xz4)+(w5xa5xz5)}{(w1xz1)+(w2xz2)+(w4xz4)+(w5xz5)}$$

$$= \frac{(0.4x1x1) + (0.34x3.25x0.75) + (0.34x7x1) + (0.34x8x1)}{(0.4x1) + (0.34x0.75) + (0.34x1) + (0.34x1)}$$

$$= \frac{0.4 + 0.6375 + 2.38 + 2.72}{0.4 + 2.25 + 0.34 + 0.34}$$

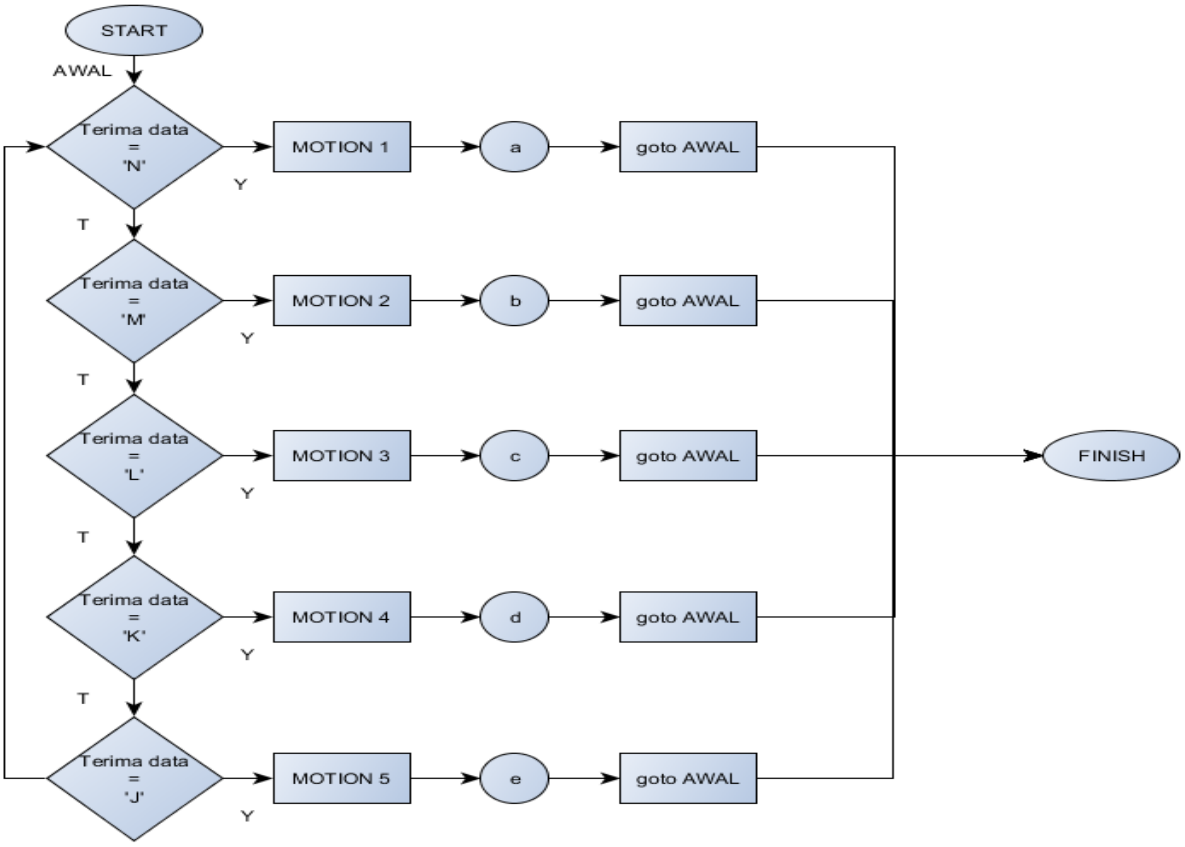
$$= \frac{6.1375}{3.33}$$

$$= 1.84$$

Dengan demikian dapat disimpulkan bahwa ciri fisik telur yang memiliki input variabel panjang = 4.8 dan variabel Lebar = 3.8 akan menghasilkan defuzzyfikasi output nilai sebesar 1.84, yang mana itu artinya kondisi telur tersebut masuk kedalam fungsi anggota ‘Sangat Kecil’.

4) Pemrograman Manipulator Robot

Gambar 46 merupakan flowchart utama dan potongan program manipulator robot, dengan menggunakan CodeVision AVR secara keseluruhan :



Gambar 46. Flowchart utama dari robot manipulator 6-DOF

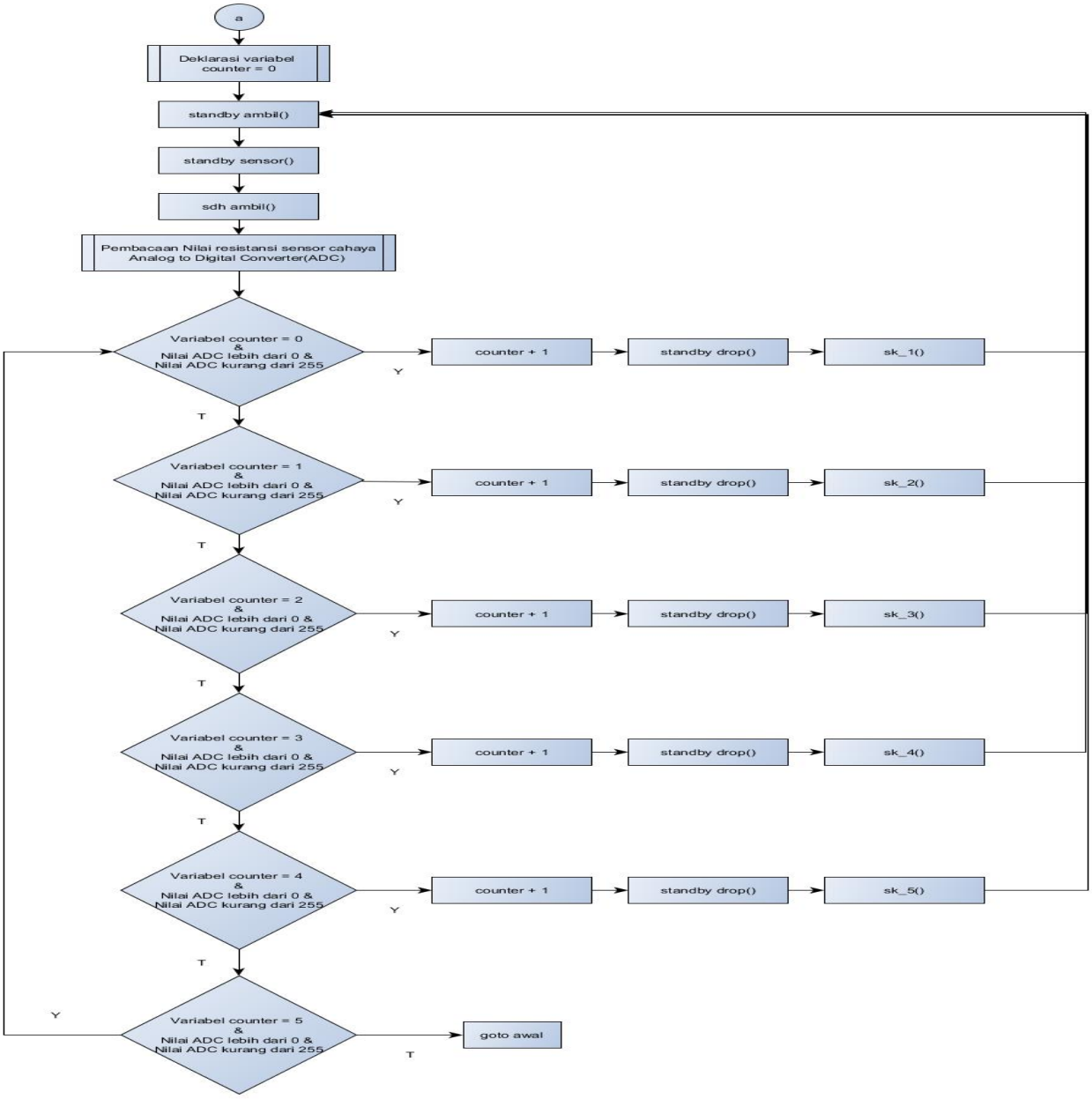
Dari pengolahan citra pada pc/laptop, didapatkan hasil dari ciri fisik telur, kemudian dari ciri fisik telur tersebut akan diindentifikasi ukuran dari telur, hasil ukuran tersebut akan mengirim data berupa *character* yang akan menginstruksikan robot untuk bergerak sesuai dengan *motion* yang diterima dari data *character*, misalkan mikrokontroller menerima data *character* = ‘N’ itu artinya instruksi tersebut akan diteruskan untuk menjalankan motion 1 dari robot yang artinya motion 1 menunjukkan bahwa telur ter-identifikasi SangatKecil, setelah itu robot akan bergerak menuju ke sensor untuk dipastikan bahwa telur telah terambil dengan baik dan kemudian robot akan bergerak menuju tatakan telur sesuai dengan

penempatan urutan untuk jenis ukuran telur yang terklasifikasi SangatKecil. Jika data yang diterima adalah *character* = 'M' itu artinya instruksi tersebut akan diteruskan untuk menjalankan motion 2 dari robot, yang artinya motion 2 menunjukkan bahwa telur ter-identifikasi Kecil. Setelah itu robot akan bergerak menuju ke sensor untuk dipastikan bahwa telur telah terambil dengan baik dan kemudian robot akan bergerak menuju tatakan telur sesuai dengan penempatan urutan untuk jenis ukuran telur yang terklasifikasi Kecil. Data *character* = 'L' artinya instruksi tersebut akan diteruskan untuk menjalankan motion 3 dari robot yang artinya motion 3 menunjukkan bahwa telur ter-identifikasi Sedang, lalu robot akan bergerak menuju ke sensor untuk dipastikan bahwa telur telah terambil dengan baik dan kemudian robot akan bergerak menuju tatakan telur sesuai dengan penempatan urutan untuk jenis ukuran telur yang terklasifikasi Sedang. Data *character* = 'K' itu artinya instruksi tersebut akan diteruskan untuk menjalankan motion 4 dari robot yang artinya motion 4 menunjukkan bahwa telur ter-identifikasi Besar, setelah itu robot akan bergerak menuju ke sensor untuk dipastikan bahwa telur telah terambil dengan baik dan kemudian robot akan bergerak menuju tatakan telur sesuai dengan penempatan urutan untuk jenis ukuran telur yang terklasifikasi Besar dan data *character* = 'J' itu artinya instruksi tersebut akan diteruskan untuk menjalankan motion 5 dari robot yang artinya motion 5 menunjukkan bahwa telur ter-identifikasi SangatBesar dan setelah itu pula robot akan bergerak menuju ke sensor untuk dipastikan bahwa telur telah terambil dengan baik dan kemudian robot akan bergerak menuju tatakan telur sesuai dengan penempatan urutan untuk jenis

ukuran telur yang terklasifikasi SangatBesar. Untuk penjabaran flowchart program diilustrasikan sebagai berikut :

a) Motion 1 (Sangat Kecil)

Gambar 47 merupakan flowchart program dari motion 1 :

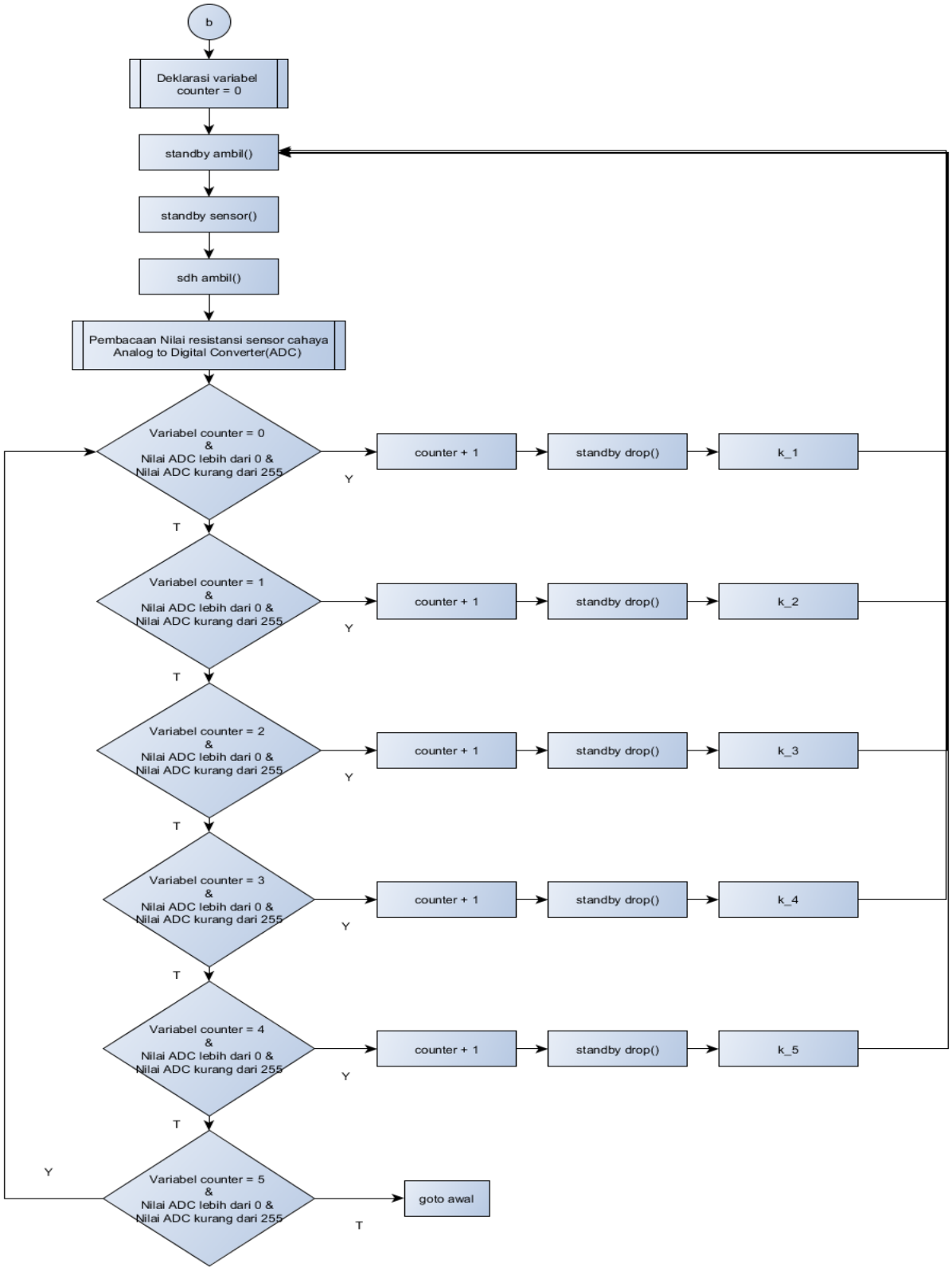


Gambar 47. Flowchart Subroutine dari motion 1

Setelah mikrokontroller menerima data character = ‘M’ yang dikirim dari pc/laptop, proses selanjutnya adalah masuk ke motion-motion robot, untuk motion 1 robot(Sangat Kecil) mikrokontroller akan mendeklarasikan variabel counter, tujuannya adalah sebagai penghitung jumlah motion 1 yang terakses, untuk mencegah penumpukan telur dalam 1 tatakan telur. Setelah deklarasi variabel counter, selanjut telur akan menjalankan beberapa bagian motion seperti standbyambil( ), motion tersebut ditujukan untuk pergerakan robot agar menyiapkan dirinya untuk mengambil telur, setelah itu robot akan mengeksekusi motion standbysensor( ), tujuannya untuk memastikan apakah telur telah ter-ambil dengan benar, jika sensor tidak mendeteksi adanya resistansi yang terjadi maka dipastikan robot belum mengambil telurnya sehingga robot secara otomatis akan mengulang ke motion standbyambil(), jika telur ter-ambil dengan benar maka sensor cahaya pula yang akan merubah counter yang sebelumnya bernilai 0 akan berubah menjadi bernilai 1, yang artinya untuk proses selanjutnya setelah mengeksekusi motion standbysensor() robot akan mengeksekusi motion untuk peletakkan telur yang berbeda, misalkan untuk counter = 0 maka robot akan mengeksekusi tempat penatakan telur dengan motion sk_1(). Jika counter = 1 maka robot akan mengeksekusi tempat penatakan telur dengan motion sk_2() dan begitu seterusnya hingga nilai counter = 5 yang artinya, ke5 tempat penatakan telur telah penuh sehingga robot akan menolak dan mengharuskan untuk melakukan proses *reject* pada tatakan telur.

b) Motion 2(Kecil)

Gambar 48 merupakan flowchart program dari motion 2 :

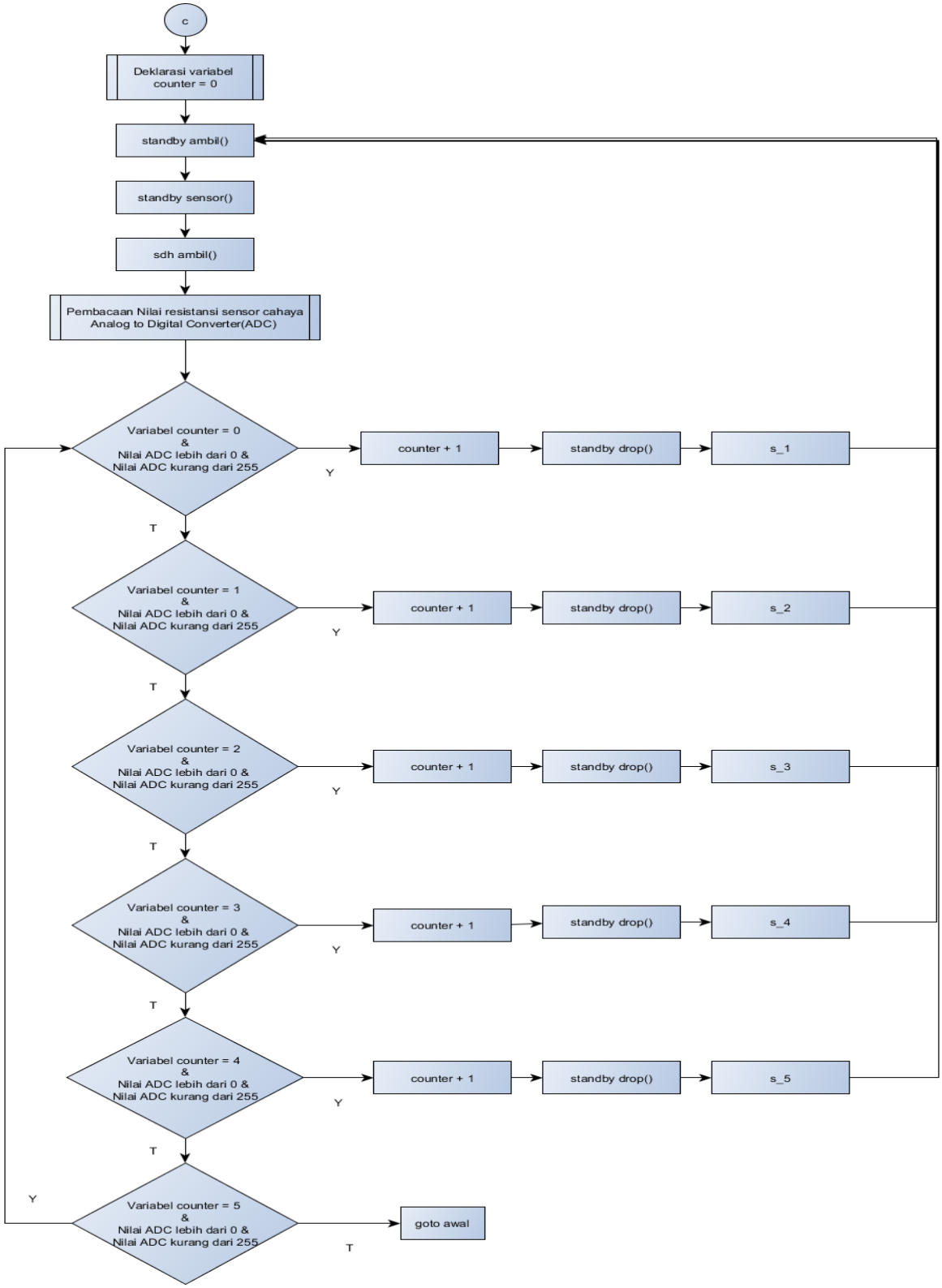


Gambar 48. Flowchart Subroutine dari Motion 2

Sama halnya dengan alur motion sebelumnya, setelah mikrokontroller menerima data character = ‘N’ untuk motion 2 (Kecil), mikrokontroller akan mendeklarasikan variabel counter, tujuannya juga sebagai penghitung jumlah motion 2 yang terakses, untuk mencegah penumpukan telur dalam 1 tatakan telur. Setelah deklarasi variabel counter lalu menjalankan beberapa bagian motion seperti standbyambil( ), yang ditujukan untuk pergerakan robot agar menyiapkan dirinya untuk mengambil telur. Setelah itu robot akan mengeksekusi motion standbysensor( ), untuk memastikan apakah telur telah ter-ambil dengan benar, jika sensor tidak mendeteksi adanya resistansi yang terjadi maka dipastikan robot belum mengambil telurnya sehingga robot secara otomatis akan mengulang ke motion standbyambil(), jika telur ter-ambil dengan benar maka sensor cahaya pula yang akan merubah counter yang sebelumnya bernilai 0 akan berubah menjadi bernilai 1, yang artinya untuk proses selanjutnya setelah mengeksekusi motion standbysensor() robot akan mengeksekusi motion untuk peletakkan telur yang berbeda, misalkan untuk counter = 0 maka robot akan mengeksekusi tempat penatakan telur dengan motion k_1(). Jika counter = 1 maka robot akan mengeksekusi tempat penatakan telur dengan motion k_2() dan begitu seterusnya hingga nilai counter = 5 yang artinya, ke5 tempat penatakan telur telah penuh sehingga robot akan menolak dan mengharuskan untuk melakukan proses *reject* pada tatakan telur.

c) Motion 3(Sedang)

Gambar 49 merupakan flowchart program dari motion 3 :

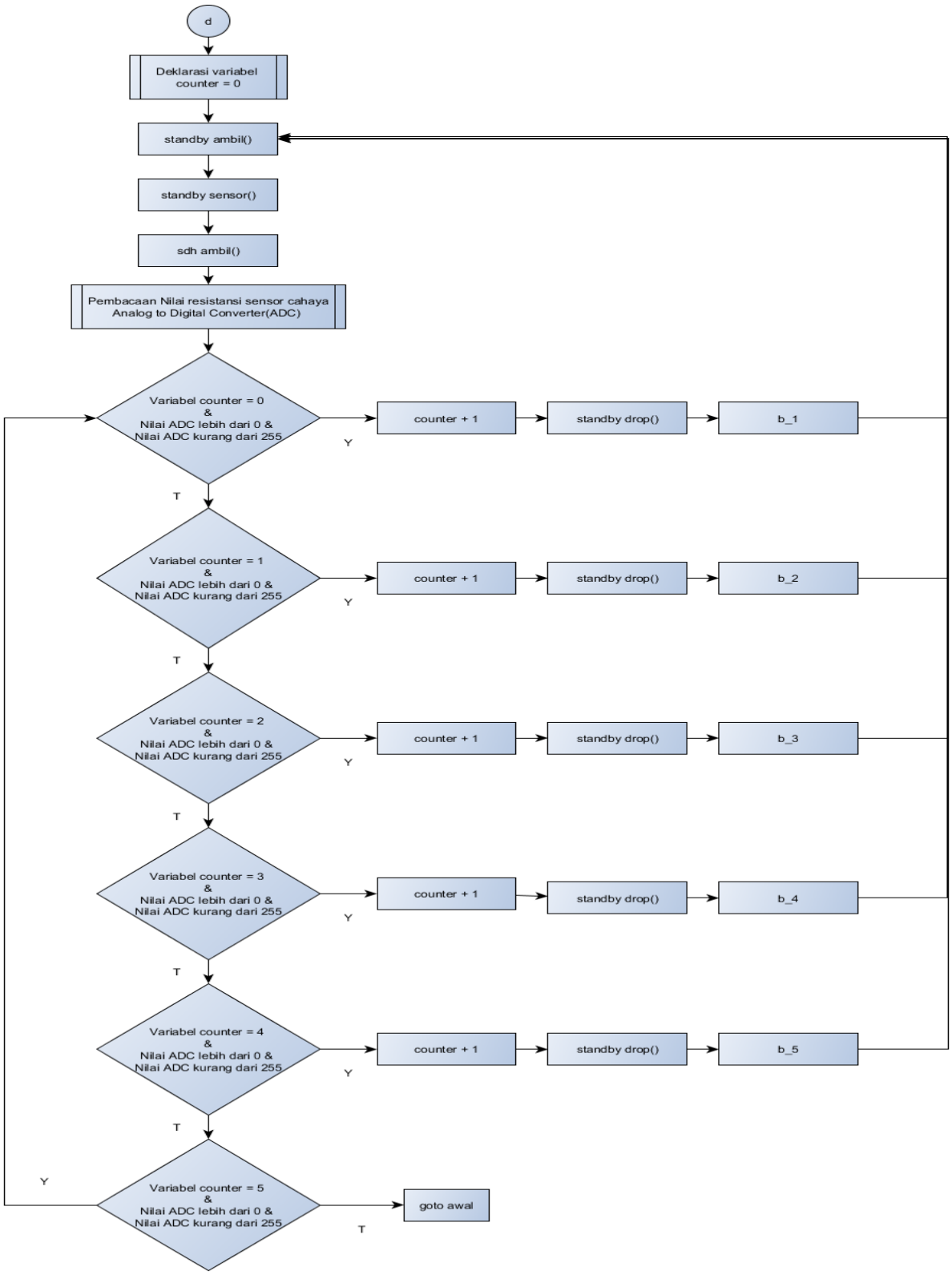


Gambar 49. Flowchart Subroutine dari Motion 3

Pada motion 3 alur program hampir sama dengan alur motion sebelumnya, setelah mikrokontroller menerima data character = ‘L’ untuk motion 3 (Sedang), mikrokontroller akan mendeklarasikan variabel counter, sebagai penghitung jumlah motion 3 yang terakses, untuk mencegah penumpukan telur dalam 1 tatakan telur. Setelah deklarasi variabel counter lalu menjalankan beberapa bagian motion seperti standbyambil( ), yang ditujukan untuk pergerakan robot agar menyiapkan dirinya untuk mengambil telur. Setelah itu robot akan mengeksekusi motion standbysensor( ), untuk memastikan apakah telur telah ter-ambil dengan benar, jika sensor tidak mendeteksi adanya resistansi yang terjadi maka dipastikan robot belum mengambil telurnya sehingga robot secara otomatis akan mengulang ke motion standbyambil(), jika telur ter-ambil dengan benar maka sensor cahaya pula yang akan merubah counter yang sebelumnya bernilai 0 akan berubah menjadi bernilai 1, yang artinya untuk proses selanjutnya setelah mengeksekusi motion standbysensor() robot akan mengeksekusi motion untuk peletakkan telur yang berbeda, misalkan untuk counter = 0 maka robot akan mengeksekusi tempat penatakan telur dengan motion s_1(). Jika counter = 1 maka robot akan mengeksekusi tempat penatakan telur dengan motion s_2() dan begitu seterusnya hingga nilai counter = 5 yang artinya, ke5 tempat penatakan telur telah penuh sehingga robot akan menolak dan mengharuskan untuk melakukan proses *reject* pada tatakan telur.

d) Motion 4(Besar)

Gambar 50 merupakan flowchart program dari motion 4 :

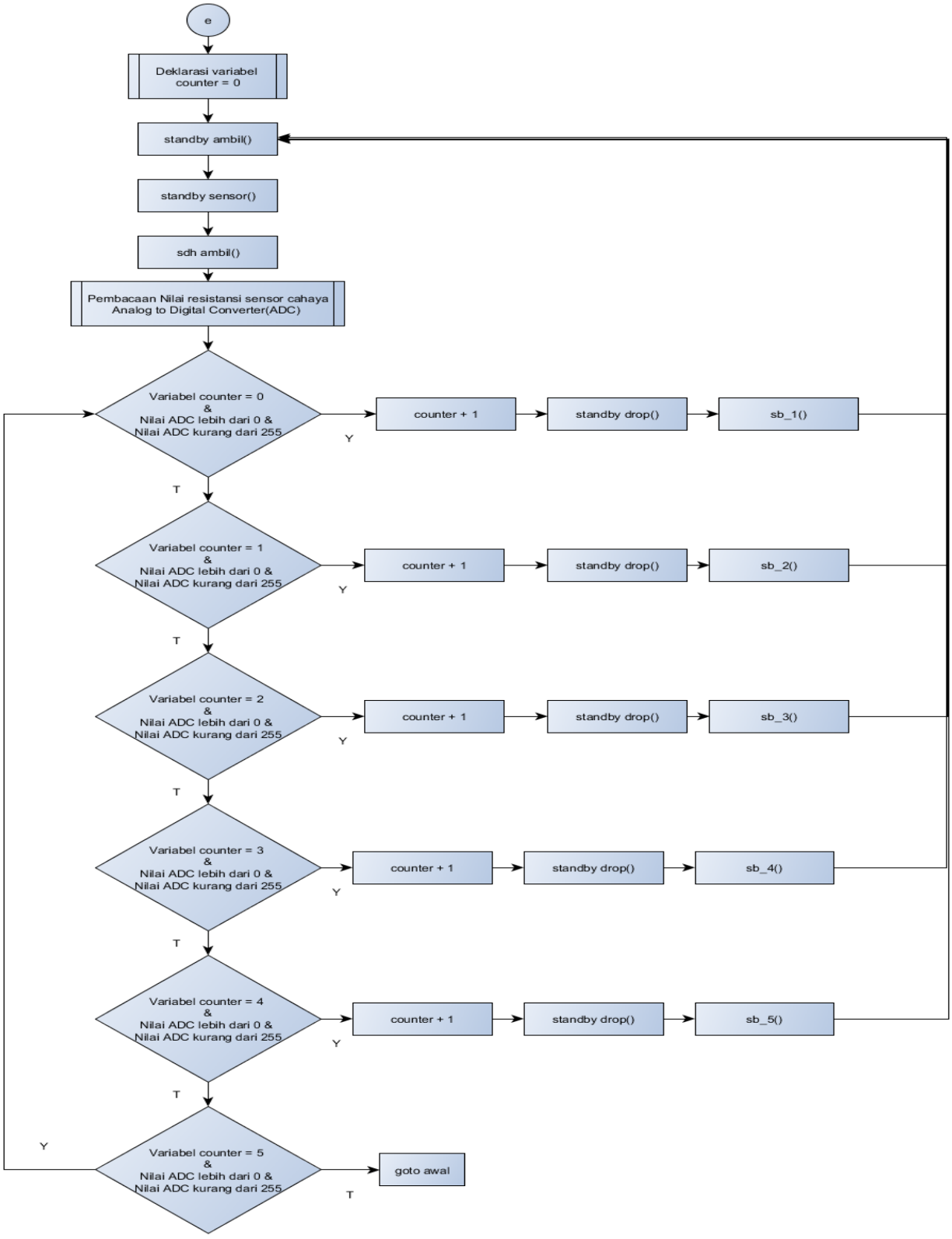


Gambar 50. Flowchart Subroutine dari Motion 4

Pada motion 4 alur program hampir sama dengan alur motion sebelumnya, setelah mikrokontroller menerima data character = ‘K’ untuk motion 4 (Besar), mikrokontroller akan mendeklarasikan variabel counter, sebagai penghitung jumlah motion 4 yang terakses, untuk mencegah penumpukan telur dalam 1 tatakan telur. Setelah deklarasi variabel counter lalu menjalankan beberapa bagian motion seperti standbyambil( ), yang ditujukan untuk pergerakan robot agar menyiapkan dirinya untuk mengambil telur. Setelah itu robot akan mengeksekusi motion standbysensor( ), untuk memastikan apakah telur telah ter-ambil dengan benar, jika sensor tidak mendeteksi adanya resistansi yang terjadi maka dipastikan robot belum mengambil telurnya sehingga robot secara otomatis akan mengulang ke motion standbyambil(), jika telur ter-ambil dengan benar maka sensor cahaya pula yang akan merubah counter yang sebelumnya bernilai 0 akan berubah menjadi bernilai 1, yang artinya untuk proses selanjutnya setelah mengeksekusi motion standbysensor() robot akan mengeksekusi motion untuk peletakkan telur yang berbeda, misalkan untuk counter = 0 maka robot akan mengeksekusi tempat penatakan telur dengan motion s_1(). Jika counter = 1 maka robot akan mengeksekusi tempat penatakan telur dengan motion s_2() dan begitu seterusnya hingga nilai counter = 5 yang artinya, ke5 tempat penatakan telur telah penuh sehingga robot akan menolak dan mengharuskan untuk melakukan proses *reject* pada tatakan telur

e) Motion 5(Sangat Besar)

Gambar 51 merupakan flowchart program dari motion 5 :



Gambar 51. Flowchart Subroutine dari Motion 5

Pada motion 5 alur program hampir sama dengan alur motion sebelumnya, setelah mikrokontroller menerima data character = ‘J’ untuk motion 5 (Sedang), mikrokontroller akan mendeklarasikan variabel counter, sebagai penghitung jumlah motion 3 yang terakses, untuk mencegah penumpukan telur dalam 1 tatakan telur. Setelah deklarasi variabel counter lalu menjalankan beberapa bagian motion seperti standbyambil( ), yang ditujukan untuk pergerakan robot agar menyiapkan dirinya untuk mengambil telur. Setelah itu robot akan mengeksekusi motion standbysensor( ), untuk memastikan apakah telur telah ter-ambil dengan benar, jika sensor tidak mendeteksi adanya resistansi yang terjadi maka dipastikan robot belum mengambil telurnya sehingga robot secara otomatis akan mengulang ke motion standbyambil(), jika telur ter-ambil dengan benar maka sensor cahaya pula yang akan merubah counter yang sebelumnya bernilai 0 akan berubah menjadi bernilai 1, yang artinya untuk proses selanjutnya setelah mengeksekusi motion standbysensor() robot akan mengeksekusi motion untuk peletakkan telur yang berbeda, misalkan untuk counter = 0 maka robot akan mengeksekusi tempat penatakan telur dengan motion sb_1(). Jika counter = 1 maka robot akan mengeksekusi tempat penatakan telur dengan motion sb_2() dan begitu seterusnya hingga nilai counter = 5 yang artinya, ke5 tempat penatakan telur telah penuh sehingga robot akan menolak dan mengharuskan untuk melakukan proses *reject* pada tatakan telur.

C. PROSES PEMBUATAN

Dalam pengerjaan proyek akhir Prototype dan Implementasi Penyortir Telur dengan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom*(DOF) diperlukan sebuah proses atau tahapan dalam pembuatannya guna menunjang pengerjaan yang lebih efektif dan effisien, Berikut merupakan tahapan dari proses pembuatan:

1. Menyiapkan Alat dan Bahan yang dibutuhkan dengan menganalisis kebutuhan baik untuk input, proses maupun output.
2. Merancang desain mekanik baik manipulator robot,tatakan telur dan base telur dengan kamera.
3. Merealisasikan rancang bangun mekanik manipulator robot, tatakan telur dan base telur dengan kamera.
4. Merancang dan mensimulasikan rangkaian elektronik baik berupa rangkaian minimum sistem, rangkaian drievr motor dengan relay, dan rangkaian sensor cahaya photodioda.
5. Mendesain layout PCB dan mencetak layout PCB dari rangkaian minimum sistem, rangkaian drievr motor dengan relay, dan rangkaian sensor cahaya photodioda.
6. Merancang dan mengimpelmentasikan program dari perangkat lunak baik dari pengolahan citra, program logika fuzzy dan program keseluruhan manipulator robot dengan CodeVisionAVR.

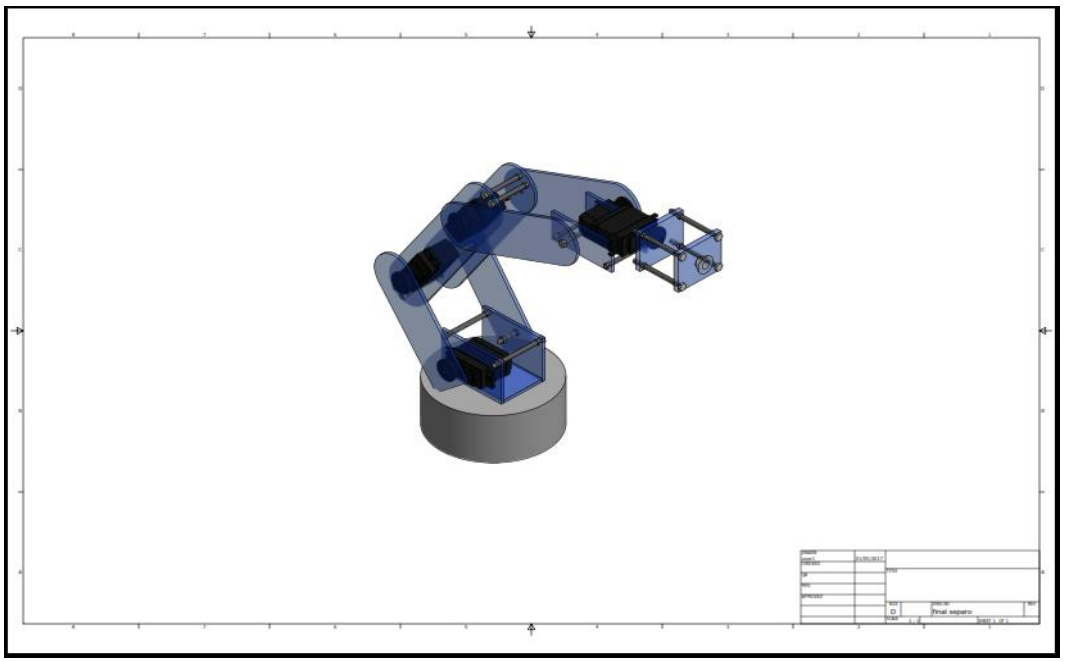
7. Menguji unjuk kerja dari tiap rancang bangun mekanik, rangkaian elektronik dan rancangan perangkat lunak.

8. Menguji unjuk kerja keseluruhan alat baik berupa rancang bangun mekanik, rancangan elektronik dan rancangan perangkat lunak.

D. DESAIN PENERAPAN

Penerapan merupakan tahapan merealisasikan tahap pembuatan yang sebenarnya, Berikut merupakan hasil dari tahap penerapan dari perancangan yang terdiri dari desain mekanik, layout PCB dan rancangan perangkat lunak.

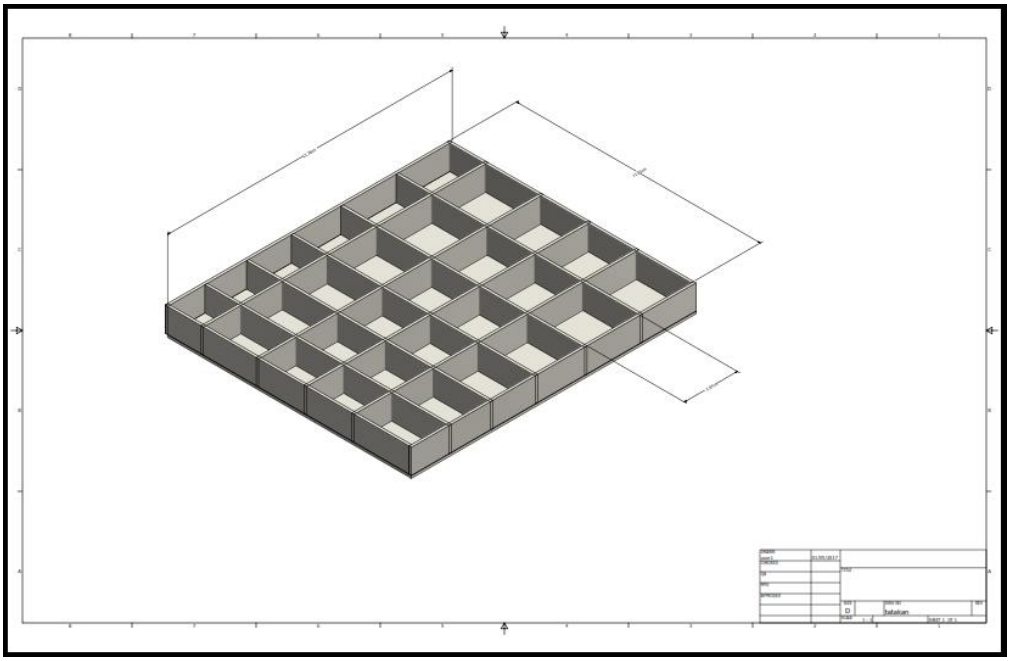
1. Desain Manipulator Robot 6-*Degree Of Freedom*



Gambar 52. Manipulator Robot 6-*Degree Of Freedom*

Gambar 56 merupkan desain dari Manipulator Robot 6-*Degree Of Freedom* yang telah dirubah ke bentuk PDF, tujuan dirubah ke PDF adalah agar lebih mudah dipahami rancangan bangun dari Manipulator Robot 6-*Degree Of Freedom*. Pada proyek akhir desain manipulator robot berbentuk *Arm Robot* dengan jumlah derajat kebebasan sebanyak 6 sendi, tujuan dibuatnya manipulator berbemtul *arm robot* dengan 6 sendi adalah untuk memudahkan dan memberikan gerakan yang lebih fleksibel sehingga ketika melakukan tugasnya robot akan mampu bekerja secara optimal, karena tidak terbatasi oleh gerak dari sendi-sendi robot itu sendiri

2. Desain Tatakan Telur

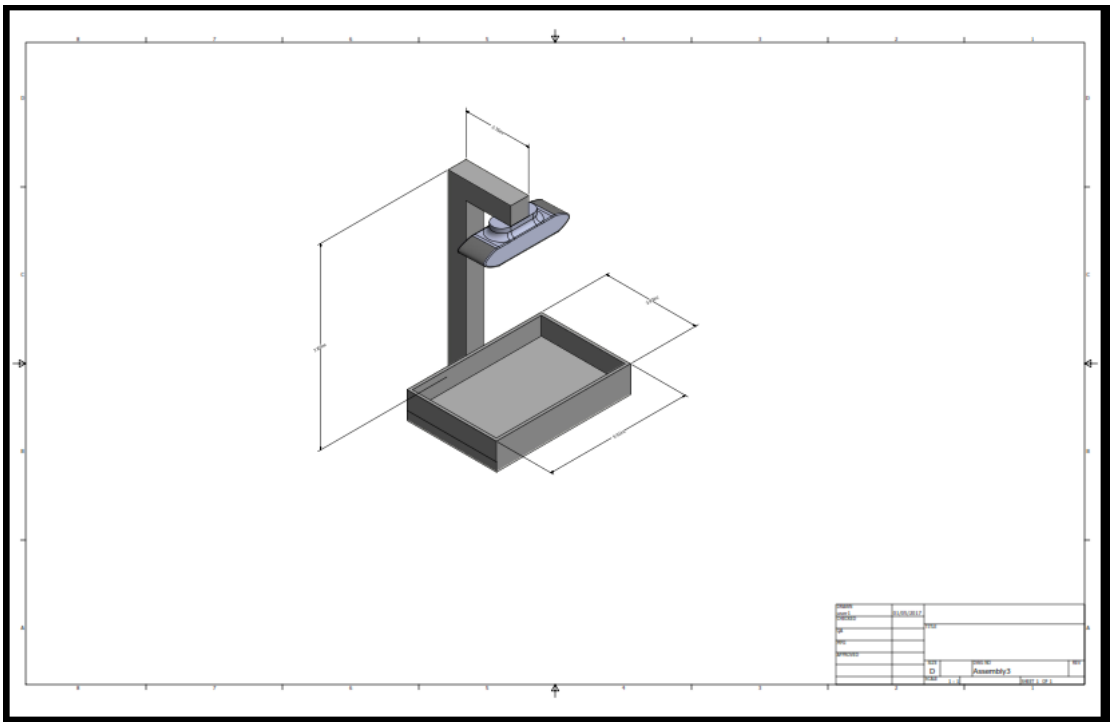


Gambar 53. Tatakan Telur

Gambar 53 merupakan desain dari tatakan telur yang telah di rubah ke PDF disertai dengan ukuran dimensi, tujuannya untuk mempermudah perealisasian bentuk desain dari tatakan telur. Desain tatakan telur merupakan salah satu dari

bagian yang penting dalam pengerjaan proyek akhir ini, tatakan telur tidak bisa sembarang dibuat karena disesuaikan dengan stimulus gerak *motion* dari manipulator robot, tingkat penempatan merupakan hal yang harus diperhatikan, sebab dari keluaran output fuzzy sebanyak 5, maka sebanyak itu pula tatakan telur dibuat baik berupa baris maupun kolomnya. Setiap kolom berjumlah 5 tempat yang artinya robot memiliki 5 motion untuk tiap fungsi keanggotaan output fuzzy dan disesuaikan dengan jumlah maksimal telur yang mampu ditata oleh robot

3. Desain Base Telur dengn Kamera

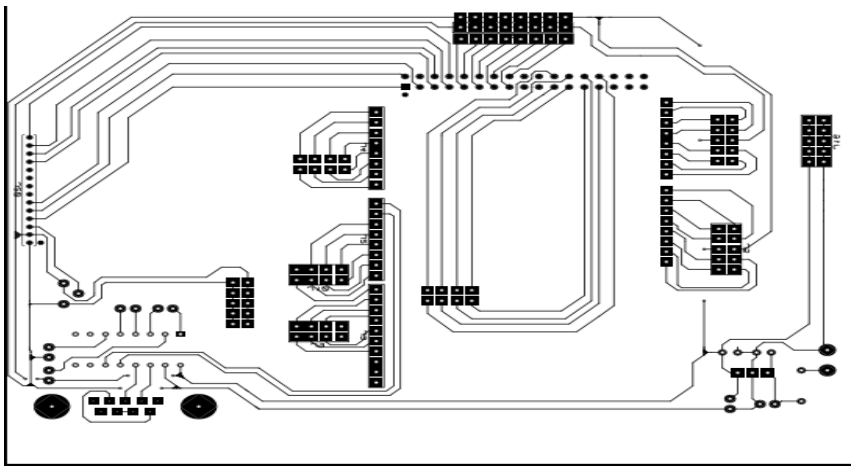


Gambar 54. Base Telur dengan Kamera

Gambar 54 merupakan desain dari Bese telur dengan Kamera yang telah di rubah ke PDF disertai dengan ukuran dimensi, tujuannya sama dengan sebelumnya yakni mempermudah perealisasian bentuk desain dari base telur dengan kamera.

Perancangan desain base telur dengan kamera dimaksudkan untuk tempat dari telur ketika akan di identifikasi ciri fisiknya, desain tempat telur disesuaikan dengan jumlah rata-rata maksimal dari telur baik dari panjang maupun lebar dari telur, setelah menentukan panjang dan lebar kemudia ketinggian dari kamera juga perlu disesuaikan mengingat program pengolahan citra yang disusun juga berdasarkan ketinggian kamera, disamping itu agar manipulator mampu mengambil telur dengan baik dan benar, maka ketinggian dari kamera harus benar-benar diperhatikan untuk mengurangi masalah deteksi telur dan masalah dari pengambilan telur yang kurang sempurna

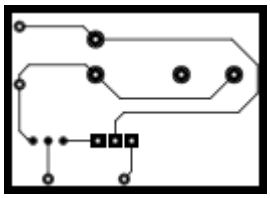
4. Desain Layout Minimum Sistem



Gambar 59. Layout PCB Rangkaian Shield

Gambar 55 merupakan layout PCB rangkaian shield dari ATmega 2560 atau yang biasa disebut dengan arduino mega.

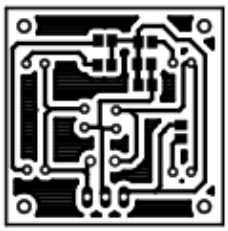
5. Desain Layout Driver Motor dengan Relay



Gambar 56. Layout Driver Motor dengan Relay

Gambar 56 merupakan layout PCB dari driver motor dengan relay 12 volt, rangkaian tersebut didesain sesederhana mungkin untuk menghemat space dari keseluruhan proyek akhir ini.

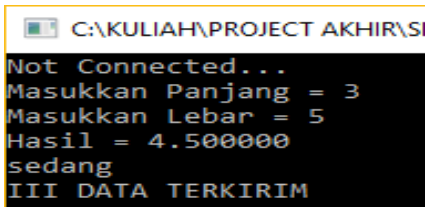
6. Desain Layout Sensor Cahaya dengan photodiode



Gambar 57. Layout PCB Rangkaian Sensor Cahaya Photodiode

Gambar 57 merupakan barisan layout PCB dari 3 sensor cahaya photodiode yang di desain dengan 2 sejejar dan 1 berada diatas 2 sensor yang sejajar.

7. Simulasi Logika Fuzzy



Gambar 58. Hasil Simulasi Logika Fuzzy

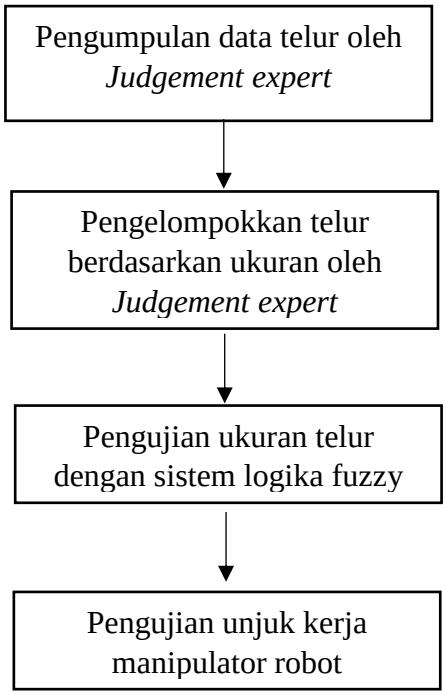
Gambar 58 merupakan hasil dari simulasi logika fuzzy pada bahasa pemrograman C yang dibangun dengan software microsoft Visual Studio 2013.

E. PENGUJIAN ALAT

Pengujian alat dilakukan untuk mendapatkan data penelitian, data penelitian dilakukan dengan tahapan membandingkan antara judgment expert dalam penyortiran telur dengan menggunakan logika fuzzy sebagai sistem cerdas penyortiran telur. Dalam pengujian alat untuk tahap awal setiap anggota variabel dari input seperti sangat kecil, kecil, sedang, besar dan sangat besar dari telur akan dikelompokkan oleh *judgment expert* penyortir telur, setelah *judgement expert* mengelompokkan tiap telur berdasarkan ukuran tersebut, proses selanjutnya telur akan di uji coba dengan mengidentifikasi ukuran fisik telur dengan pengolahan citra yang berbasis logika fuzzy. Setelah diidentifikasi baik berdasarkan *judgment expert* maupun dengan sistem cerdas selanjutnya akan dibandingkan data telur tersebut.

Guna mengetahui unjuk kerja manipulator robot, setelah data diidentifikasi kemudian manipulator akan memindahkan ketempat tatakan telur, dalam proses ini perekaman data diperoleh dengan mempertimbangkan ketepatan manipulator

robot dalam meletakkan telur keatas tatakan telur. Gambar 59 merupakan diagram proses pengujian alat :



Gambar 59. Diagram Proses Pengujian Alat

1. Pengumpulan data telur oleh *Judgement expert*

Pada tahap ini pengumpulan data telur akan dikumpulkan berdasarkan klasifikasi ukuran oleh *Judgement expert*. Tujuan dari proses ini adalah untuk mengetahui rentang nilai minimal dan maksimal dari satu jenis klasifikasi telur baik yang Sangat Kecil, Kecil, Sedang, Besar dan Sangat Besar. Setelah nilai dari data telur tiap klasifikasi telah diidentifikasi oleh *Judgement expert* kemudian telur akan dikelompokkan sesuai dengan masing-masing klasifikasi jenis telur. Setelah *Judgement expert* mengelompokkan telur sesuai dengan klasifikasi jenis telur,

kemudian telur akan diukur satu persatu bobotnya dan disesuaikan dengan Standar Nasional Indonesia yang mengatur tentang berat dari telur baik dari ukuran sangat kecil, sedang, besar dan sangat besar.

2. Pengelompokkan telur berdasarkan ukuran oleh *Judgement expert*

Proses selanjutnya dari pengumpulan data telur oleh *Judgement expert* adalah pengelompokkan telur berdasarkan ukuran oleh *Judgement expert*, tujuan dari proses ini adalah untuk menentukan dari beberapa telur yang ada untuk diklasifikasikan berdasarkan ukuran telur apakah termasuk Sangat Kecil, Kecil, Sedang, Besar atau Sangat Besar.

3. Pengujian ukuran telur dengan sistem logika fuzzy

Proses berikutnya dari pengelompokkan ukuran oleh *Judgement expert* adalah menguji program dari logika fuzzy, setelah data minimal dan maksimal dari ukuran tiap klasifikasi telur diperoleh lalu kemudian *Judgement expert* mengelompokkan ukuran telur tersebut apakah termasuk Sangat Kecil, Kecil, Sedang, Besar dan Sangat Besar. Untuk pengukuran panjang dan lebar telur secara automatis dari pengolahan citra akan diberikan toleransi pembacaan sebesar 0.1 cm dari pengukuran yang sebenarnya, hal ini dilakukan karena faktor pembacaan codebook algoritma yang menghitung panjang dan lebar berdasarkan garis terluar dari kontur *object*. Selebihnya dari toleransi 0.1, data output tidak dapat dipertanggung jawabkan keakuratannya. Selanjutnya program logika fuzzy akan diuji dengan membandingkan hasil dari *Judgement expert* tersebut ditiap klasifikasi

jenis telur. Setelah diketahui hasilnya maka hasil tersebut akan dibandingkan berdasarkan proses manual oleh *Judgement expert* dengan sistem logika fuzzy.

4. Pengujian unjuk kerja manipulator robot

Pada proses pengujian unjuk kerja manipulator robot akan diidentifikasi kemampuan robot untuk menempatkan telur pada tatakan telur. Percobaan akan dilakukan dengan mempertimbangkan ketetpatan dan ketidak tepatan manipulator dalam meletakkan telur kedalam ruang tatakan telur yang telah tersedia.

BAB IV

PENGUJIAN DAN PEMBAHASAN

A. HASIL PENGUJIAN

1. Sistem Logika Fuzzy

Pengujian sistem logika Fuzzy ditujukan untuk mengetahui unjuk kerja dari program logika fuzzy dari hasil perancangan prototype dan implementasi penyortir telur dengan logika fuzzy pada manipulator 6-*Degree of Freedom*(DOF). Output dari sistem logika fuzzy tersebut akan dibandingkan dengan hasil perhitungan secara Automatis berbasis pengolahan citra dan secara manual yang disesuaikan dengan ukuran panjang dan lebar telur untuk diukur secara manual menggunakan jangka sorong.

Hasil dari pengujian tersebut akan dimuat pada tabel klasifikasi jenis telur secara terpisah, setelah data hasil pengujian dimuat selanjutnya akan dicari *error* atau kesalahannya. Perhitungan *error* diperoleh dengan mencari kondisi fisik ukuran telur secara manual dan dibandingkan dengan kondisi fisik ukuran telur secara automatis, Berikut merupaan rumus yang akan digunakan untuk mencari persentase dari nilai *error* tersebut :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

Dengan mengacu pada persamaan diatas, selanjutnya tiap klasifikasi telur akan dicari nilai *persentase* dari ketepatan tersebut, tabel 7 dan penjabaran dari tiap klasifikasi ukuran telur :

a) **Sangat Kecil**

Tabel 7. Hasil Pengujian Sangat Kecil.

No	MANUAL (*JUDGEMENT EXPERT*)			SISTEM FUZZY		
	Klasifikasi Telur Manual(MATLAB)	UKURAN TELUR MANUAL (CM)		Klasifikasi Telur Automatis	UKURAN TELUR AUTOMATIS (CM)	
1	Sangat Kecil(1.2)	P = 4.7	L= 3.7	Sangat Kecil(1.06)	P= 4.71	L=3.61
2	Sangat Kecil(1.2)	P= 4.7	L= 3.7	Sangat Kecil(1.06)	P= 4.71	L= 3.64
3	Sangat Kecil(1.81)	P= 4.8	L= 3.7	Sangat Kecil(1.68)	P= 4.87	L= 3.51
4	Sangat Kecil(4.43)	P= 5.0	L= 3.8	Kecil(3.85)	P= 5.16	L= 3.66
5	Sangat Kecil(1.2)	P= 4.5	L= 3.7	Sangat Kecil(1.06)	P= 4.52	L= 3.59

Dengan mengacu persamaan untuk mencari *% ketepatan* dari pengukuran telur secara automatis, maka tabel diatas dapat diuraikan sebagai berikut :

Diketahui : Data error = 1

Jumlah data = 5

Ditanya : % Ketepatan = ...?

Jawab :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{5-1}{5}\ x\ 100\%\ = 80\%$$

Dengan demikian dari analisis data yang dilakukan pada *linguistik value* sangat kecil, dapat disimpulkan bahwa unjuk kerja program logika fuzzy memiliki persentase akurasi ketepatan sebesar 80%

b) **Kecil**

Tabel 8. Hasil Pengujian Kecil.

No	MANUAL (*JUDGEMENT EXPERT*)			SISTEM FUZZY		
	Klasifikasi Telur Manual	UKURAN TELUR MANUAL (CM)		Klasifikasi Telur Automatis	UKURAN TELUR AUTOMATIS (CM)	
1	Kecil(3.46)	P = 4.9	L= 3.8	Kecil(3.48)	P= 4.96	L= 3.86
2	Kecil(4.43)	P= 5.0	L= 3.8	Kecil(4.43)	P= 5.07	L= 3.76
3	Kecil(3.87)	P= 4.9	L= 3.9	Kecil(4.29)	P= 4.99	L= 3.83
4	Kecil(4.43)	P= 5.0	L= 3.8	Kecil(4.43)	P= 5.0	L= 3.80
5	Kecil(2.94)	P= 4.8	L= 3.8	Sangat Kecil(2.70)	P= 4.83	L= 3.75

Dengan mengacu persamaan untuk mencari *% ketepatan* dari pengukuran telur secara automatis, maka tabel diatas dapat diuraikan sebagai berikut :

Diketahui : Data error = 1

Jumlah data = 5

Ditanya : % Ketepatan = ...?

Jawab :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{5-1}{5}\ x\ 100\% = 80\%$$

Dengan demikian dari analisis data yang dilakukan pada *linguistik value* kecil, dapat disimpulkan bahwa unjuk kerja program logika fuzzy memiliki persentase akurasi ketepatan sebesar 80%

c) **Sedang**

Tabel 9. Hasil Pengujian Sedang.

No	MANUAL (*JUDGEMENT EXPERT*)			SISTEM FUZZY		
	Klasifikasi Telur Manual	UKURAN TELUR MANUAL (CM)		Klasifikasi Telur Automatis	UKURAN TELUR AUTOMATIS (CM)	
1	Sedang(6.08)	P = 5.2	L=4.0	Sedang(6.32)	P=5.20	L=4.09
2	Sedang(6.44)	P= 5.2	L=4.0	Sedang(6.32)	P=5.28	L=4.03
3	Sedang(6.08)	P= 5.2	L=4.0	Sedang(5.08)	P=5.10	L=3.80
4	Sedang(5.95)	P= 5.3	L= 3.8	Sedang(5.02)	P= 5.42	L=3.83
5	Sedang(7.67)	P=5.3	L= 4.2	Besar(7.99)	P= 5.26	L=4.3

Dengan mengacu persamaan untuk mencari *% ketepatan* dari pengukuran telur secara automatis, maka tabel diatas dapat diuraikan sebagai berikut :

Diketahui : Data error = 1

Jumlah data = 5

Ditanya : % Ketepatan = ...?

Jawab :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{5-1}{5}\ x\ 100\%\ = 80\%$$

Dengan demikian dari analisis data yang dilakukan pada *linguistik value* seddang, dapat disimpulkan bahwa unjuk kerja program logika fuzzy memiliki persentase akurasi ketepatan sebesar 80%

d) **Besar**

Tabel 10. Hasil Pengujian Besar.

No	MANUAL (*JUDGEMENT EXPERT*)			SISTEM FUZZY		
	Klasifikasi Telur Manual	UKURAN TELUR MANUAL (CM)		Klasifikasi Telur Automatis	UKURAN TELUR AUTOMATIS (CM)	
1	Besar(7.3)	P= 5.5	L= 4.1	Besar(6.59)	P= 5.62	L= 3.95
2	besar6.64)	P= 5.2	L=4.1	Besar(6.95)	P=5.28	L=4.12
3	Sedang(6.08)	P=5.2	L=4.0	Sedang(5.03)	P= 5.18	L= 4.04
4	Besar(7.67)	P= 5.1	L= 4.1	Besar(7.44)	P= 5.18	L= 4.16
5	besar(6.54)	P= 5.2	L=4.1	Besar(6.95)	P=5.28	L=4.12

Dengan mengacu persamaan untuk mencari *% ketepatan* dari pengukuran telur secara automatis, maka tabel diatas dapat diuraikan sebagai berikut :

Diketahui : Data error = 1

Jumlah data = 5

Ditanya : % Ketepatan = ...?

Jawab :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{5-1}{5}\ x\ 100\%\ = 80\%$$

Dengan demikian dari analisis data yang dilakukan pada *linguistik value* Besar, dapat disimpulkan bahwa unjuk kerja program logika fuzzy memiliki persentase akurasi ketepatan sebesar 80%.

e) **Sangat Besar**

Tabel 11. Hasil Pengujian Sangat Besar.

No	MANUAL (*JUDGEMENT EXPERT*)			SISTEM FUZZY		
	Klasifikasi Telur Manual	UKURAN TELUR MANUAL (CM)		Klasifikasi Telur Automatis	UKURAN TELUR AUTOMATIS (CM)	
1	Sangat Besar	P= 5.8	L= 4.2	Sangat Besar(8.20)	P= 5.93	L= 4.20
2	Sangat Besar	P= 5.8	L= 4.4	Sangat Besar(8.39)	P= 5.88	L= 4.60
3	Sangat Besar	P= 5.7	L= 4.1	Sangat Besar(7.58)	P= 5.56	L= 4.07
4	Sangat Besar	P= 5.4	L= 4.2	Sangat Besar(8.20)	P= 5.4	L= 4.20
5	Sangat Besar	P= 5.5	L= 4.1	Sangat Besar(8.20)	P= 5.32	L= 4.20

Dengan mengacu persamaan untuk mencari *% ketepatan* dari pengukuran telur secara automatis, maka tabel diatas dapat diuraikan sebagai berikut :

Diketahui : Data error = 0

Jumlah data = 5

Ditanya : % Ketepatan = ...?

Jawab :

$$\%\ Ketepatan = \frac{Jumlah\ data - error}{jumlah\ data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{5-0}{5}\ x\ 100\%\ = 100\%$$

Dengan demikian dari analisis data yang dilakukan pada *linguistik value* sangat Besar, dapat disimpulkan bahwa unjuk kerja program logika fuzzy memiliki persentase akurasi ketepatan sebesar 100%

2. Unjuk Kerja Keseluruhan

Pengujian unjuk kerja keseluruhan ditujukan untuk mengetahui unjuk kerja secara menyeluruh baik dari hasil sistem logika fuzzy maupun kemampuan manipulator untuk melakukan penyortiran telur. Unjuk kerja keseluruhan tersebut akan dimodelkan sebagaimana ketepatan manipulator melakukan proses penyortiran telur dari tiap kliasifikasi jenis telur yang ada(Sangat Kecil,Kecil, Sedang, Besar dan Sangat Besar). Rentang percobaan sebanyak 5 kali penyortiran telur dari total keseluruhan jenis klasifikasi telur.

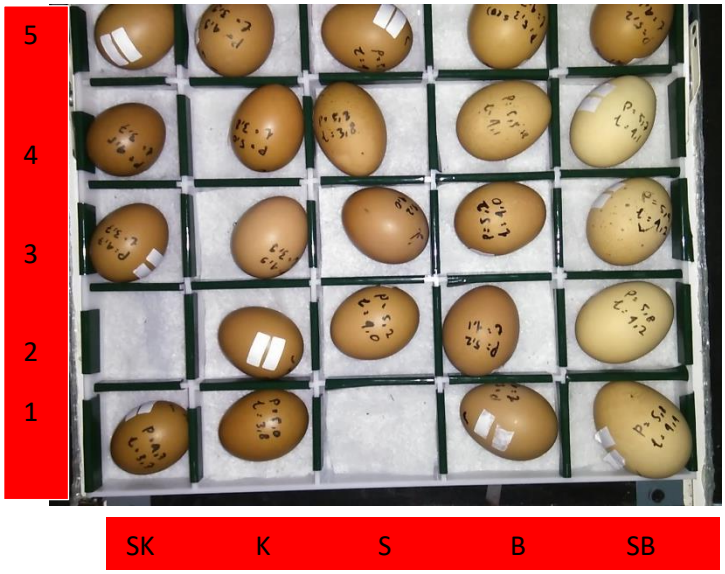
Hasil dari pengujian tersebut akan dimuat pada tabel klasifikasi jenis telur secara terpisah, dan setelah data hasil pengujian dimuat selanjutnya akan dicari *error* atau kesalahannya dari proses ketepatan manipulator dalam melakukan penyortiran telur dan logika fuzzy dalam menklasifikasikan ukuran telur. Perhitungan *error* diperoleh dengan mencari selisih dari jumlah tepat dan kurang tepat dari kemapuan manipulator melakukan penyortiran telur dari tiap klasifikasi telur. Hasil tersebut kemudian di rangkum dalam besaran persen(prosentase) dari *error*. Persamaan 1 merupakan rumus yang akan digunakan untuk mencari persentase dari nilai *error* manipulator robot terhadap tatakan telur dan persamaan 2 merupakan rumus yang akan digunakan untuk mencari persentase kemampuan logika fuzzy dalam menklasifikasikan ukuran telur:

$$\%\ Ketepatan = \frac{Jumlah\ Data - Jumlah\ Tdk\ Tepat}{jumlah\ Data}\ x\ 100\% \qquad (1)$$

$$\%\ \text{Ketepatan klasifikasi ukuran telur} = \frac{Jumlah\ Data - Jumlah\ \text{ukuran kurang tepat}}{jumlah\ Data}\ x\ 100\% \quad (2)$$

Gambar 60 dan Gambar 61 merupakan hasil unjuk kerja keseluruhan proses penyortiran telur dari kondisi telur berdasarkan klasfikasinya berurutan mulai kiri: Sangat Kecil, Kecil, Sedang, Besar, Sangat Besar .

Gambar 60. Jumlah dan Kondisi telur berdasarkan klasifikasinya (dari kiri: Sangat Kecil, Kecil, Sedang, Besar, Sangat Besar)



Gambar 61. Kondisi Telur setelah disortir

Tabel 12. Hasil Pengujian Unjuk Kerja Alat Keseluruhan

Percobaan No.	*Judgement expert*	Klasifikasi ukuran telur dengan pengolahan citra dan Logika fuzzy	Berat Telur (gram)	Penempatan Telur Dengan Robot Mengacu pada sistem Logika Fuzzy		Penempatan Telur Dengan Robot Mengacu pada *Judgement expert*	
				Tepat	Kurang Tepat	Tepat	Kurang Tepat
1.	Sangat Kecil	Sangat Kecil	40.3 gr	✓	-	✓	-
2.	Sangat Kecil	Sangat Kecil	42.5 gr	-	✓	-	✓
3.	Sangat Kecil	Sangat Kecil	42.4 gr	✓	-	✓	
4.	**Sangat Kecil**	**Kecil**	45.6 gr	✓	-	-	✓
5.	Sangat Kecil	Sangat Kecil	43.7 gr	✓	-	✓	-
6.	Kecil	Kecil	45.9 gr	✓	-	✓	-
7.	Kecil	Kecil	47.5 gr	✓	-	✓	-
8.	Kecil	Kecil	46.8 gr	✓	-	✓	-
9.	Kecil	Kecil	47.2 gr	✓	-	✓	-
10.	**Kecil**	**Sangat Kecil**	41.7 gr	✓	-	-	✓
11.	Sedang	Sedang	55.8 gr	-	✓	-	✓
12.	Sedang	Sedang	57.2 gr	✓	-	✓	-
13.	Sedang	Sedang	54.4 gr	✓	-	✓	-
14.	Sedang	Sedang	59.3 gr	✓	-	✓	-
15.	**Sedang**	**Besar**	62.2 gr	✓	-	-	✓
16.	Besar	Besar	65.7 gr	✓	-	✓	-
17.	Besar	Besar	64.3 gr	✓	-	✓	-
18.	**Besar**	**Sedang**	58.6 gr	✓	-	-	✓
19.	Besar	Besar	62.2 gr	✓	-	✓	-
20.	Besar	Besar	64.5 gr	✓	-	✓	-
21.	Sangat Besar	Sangat Besar	68.9 gr	✓	-	✓	-
22.	Sangat Besar	Sangat Besar	68.5 gr	✓	-	✓	-
23.	Sangat Besar	Sangat Besar	68.8 gr	✓	-	✓	-
24.	Sangat Besar	Sangat Besar	69.4 gr	✓	-	✓	-
25.	Sangat Besar	Sangat Besar	67.4 gr	✓	-	✓	-

1) Dengan mengacu pada Tabel 12. Didapatkan hasil validitas data sebanyak 3 data, yakni Klasifikasi ukuran telur dengan pengolahan citra dan Logika fuzzy, Penempatan Telur dengan Robot Mengacu pada sistem Logika fuzzy, serta Penempatan Telur dengan Robot mengacu pada judgement expert untuk lebih lengkapnya dapat dilihat pada analisis data 1, 2 dan 3:

1) Klasifikasi ukuran telur dengan pengolahan citra dan Logika fuzzy

Data klasifikasi ukuran kurang tepat = 4

Jumlah Data = 25

$$\%\ \text{Ketepatan klasifikasi ukuran telur} = \frac{Jumlah\ Data - Jumlah\ \text{ukuran kurang tepat}}{jumlah\ Data}\ x\ 100\%$$

$$\%\ \text{Ketepatan klasifikasi ukuran telur} = \frac{25-4}{25}\ x\ 100\% = 84\%$$

2) Penempatan Telur dengan Robot Mengacu pada sistem Logika fuzzy

Data Tidak Tepat = 2

Jumlah Data = 25

$$\%\ Ketepatan = \frac{Jumlah\ Data - Data\ Kurang\ Tepat}{jumlah\ Data}\ x\ 100\%$$

$$\%\ Ketepatan = \frac{25-2}{25}\ x\ 100\% = 92\%$$

3) Penempatan Telur dengan Robot mengacu pada judgement expert

Data *error* = 6

Jumlah Data = 25

$$\%\ \text{Ketepatan} = \frac{\text{Jumlah Data} - \text{Data } Kurang\ Tepat}{jumlah\ Data}\ x\ 100\%$$

$$\%\ \text{Ketepatan} = \frac{25-6}{25}\ x\ 100\% = 76\%$$

Berdasarkan analisis data yang telah dilakukan, diperoleh persentase keberhasilan dari unjuk kerja penyortir telur berbasis manipulator 6-*Degree Of Freedom(*DOF) sebasar 92% ketepatan dalam meletakkan telur pada tatakan telur

yang telah disiapkan dan kemampuan logika fuzzy dalam menklasifikasikan ukuran telur memiliki persentase keberhasilan sebesar 84% dan untuk unjuk kerja keseluruhan memiliki persentase keberhasilan sebesar 76%.

B. PEMBAHASAN

Sebelum melakukan pengujian baik sistem logika fuzzy maupun unjuk kerja dari manipulator robot, hal utama yang harus diperhatikan untuk menguji sistem logika fuzzy adalah konfigurasi ukuran telur yang dilakukan oleh *judgement expert*, dari hasil penyortiran yang dilakukan oleh *judgement expert* ada beberapa kondisi ukuran telur yang tidak sesuai, berikut merupakan paparan dari perbedaan kondisi ukuran telur tersebut :

1. Untuk ukuran Sangat Kecil, terlihat pada tabel 7 nomor 4 yang telah disusun, telur tersebut memiliki P(panjang) = 5.0 dan L(lebar) = 3.8, *judgement expert* menganggap ukuran tersebut adalah ideal sangat kecil, namun ketika dilakukan proses perhitungan oleh sistem logika fuzzy, telur tersebut memiliki ukuran **kecil** dengan representasi nilai sebesar 3,85. Hasil tersebut dipertimbangkan dengan linguistik value dari keanggotaan output fuzzy masuk dalam kategori kecil didukung dengan perhitungan secara simulasi oleh MATLAB memiliki representasi nilai sebesar 4.45, yang mana nilai tersebut juga termasuk dalam linguistik value **kecil.**

2. Pada konfigurasi ukuran Kecil, dapat dilihat pada tabel 8 nomor 5 yang telah disusun, telur tersebut memiliki P(panjang) = 4.8 dan L(lebar) = 3.8, *judgement expert* menganggap ukuran tersebut adalah ideal **kecil**, namun ketika dilakukan proses perhitungan dengan sistem logika fuzzy, telur tersebut memiliki ukuran **Sangat Kecil** dengan representasi nilai sebesar 2,7. Hasil tersebut dipertimbangkan terhadapat linguistik value dari keanggotaan output fuzzy termasuk dalam kategori sangat kecil, pelugasan tersebut didukung dengan perhitungan secara simulasi oleh MATLAB memiliki representasi nilai sebesar 2.94, yang mana nilai tersebut juga termasuk dalam linguistik value **Sangat kecil.**
3. Untuk konfigurasi ukuran Sedang, terlihat pada tabel 9 nomor 5 yang telah disusun, telur tersebut memiliki P(panjang) = 5.3 dan L(lebar) = 4.2, *judgement expert* menilai ukuran tersebut adalah ideal Sedang, namun ketika dilakukan proses perhitungan oleh sistem logika fuzzy, telur tersebut memiliki ukuran **Besar** dengan representasi nilai sebesar 7,99. Hasil tersebut jika dipertimbangkan dengan linguistik value dari keanggotaan output fuzzy masuk dalam kategori besar, hasil tersebut juga didukung dengan perhitungan secara simulasi oleh MATLAB memiliki representasi nilai sebesar 7.67, yang mana nilai tersebut juga termasuk dalam linguistik value **Besar.**
4. Untuk ukuran Besar, terlihat pada tabel 10 nomor 3 yang telah disusun, telur tersebut memiliki P(panjang) = 5.2 dan L(lebar) = 4.0, *judgement expert* menganggap ukuran tersebut adalah ideal Besar, namun ketika

dilakukan proses perhitungan oleh sistem logika fuzzy, telur tersebut memiliki ukuran **Sedang** dengan representasi nilai sebesar 5,05. Hasil tersebut dipertimbangkan dengan linguistik value dari keanggotaan output fuzzy masuk dalam kategori sedang didukung dengan perhitungan secara simulasi oleh MATLAB memiliki representasi nilai sebesar 6.08, yang mana nilai tersebut juga termasuk dalam linguistik value **Sedang.**

5. Untuk mengetahui %keberhasilan dari keseluruhan sistem logika fuzzy dengan pengolahan citra, dapat dianalisis dengan melakukan perhitungan sabagai berikut :

$$\%\, Tot.Keberhasilan = \frac{Jumlah\ \%Ketepatan\ tiap\ kondisi}{jumlah\ Data\ tiap\ kondisi}$$

$$\%\, Tot.Keberhasilan = \frac{80\% + 80\% + 80\% + 80\% + 100\%}{5}$$

$$\%\, Tot.Keberhasilan = 84\%$$

Dengan demikian dapat disimpulkan bahwa persantese pembacaan ukuran telur pada logika fuzzy tersebut memiliki nilai sebesar 84%.

6. Untuk berat dari tiap butir telur yang didapatkan dengan mengukur berat telur dengan timbangan manual digital(ketelitian 0.1), didapatkan hasil untuk kategori sangat kecil dan kecil memiliki berat kurang dari 50gram, untuk kategori sedang memiliki berat antara 50 gram – 60 gram, untuk ukuran besar hingga sangat besar memiliki berat lebih dari 60gram. Hal tersebut menunjukkan bahwa telur yang digunakan pada

proyek akhir ini, sesuai dengan berat dari Standar Nasional Indonesia SNI No. 3926:2008 tentang berat telur.

7. Penempatan Telur dengan Robot Mengacu pada sistem Logika fuzzy memiliki keberhasilan sebesar 92%. Penempatan telur mengacu pada sistem logika fuzzy merupakan jenis validitas data yang diperoleh dari kemampuan robot memindahlan telur ke tatakan telur berdasarkan logika fuzzy, sehingga data *error* diperoleh bilamana justifikasi dari sistem logika fuzzy adalah kecil, maka penempatan telur pada tatakan telur seharusnya pada baris dan kolom dari kecil, jika robot tidak dapat menempatkan telur pada baris dan kolom tatakan telur kecil maka data tersebut dianggap *error*.
8. Penempatan Telur dengan Robot mengacu pada judgement expert memiliki keberhasilan sebesar 76%. Penempatan telur mengacu pada judgement expert merupakan jenis validitas data yang diperoleh dari kemampuan robot memindahlan telur ke tatakan telur berdasarkan *judgement expert*, sehingga data *error* diperoleh bilamana justifikasi telur sangat kecil maka seharusnya telur ditempatkan pada kolom dan baris sangat kecil dari tatakan telur.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan pengujian dan pembahasan pada Proyek Akhir ini, dapat diambil beberapa kesimpulan mengenai rancang bangun dari prototype dan implementasi penyortir telur dengan logika fuzzy pada manipulator 6-*Degree of freedom,* yakni :

1. Rancang Bangun Manipulator Robot 6-*Degree of Freedom* keseluruhan

a. Perancangan Mekanik

1. Manipulator robot 6-*Degree of Freedom*

Manipulator robot didesain dengan memanfaatkan derajat kebebasan sebanyak 6 ditujukan untuk ruang gerak dari manipulator robot agar lebih leluasa dalam melakukan tugas sebagai penyortir telur, manipukator robot disinergikan dengan rangka manipulator berbahan akrilik dengan tebal 3 mm yang ringan dan kuat sehingga tiap aktuator pada sendi manipulator robot(*Degree of Freedom*) tidak terlalu menopang beban yang terlalu berat serta arus beban yang dihasilkan dari aktuator tidak terlalu besar pula.

2. Tatakan telur

Tatakan telur didesain dan dibuat dengan berbahan dasar akrlik tebal 3 mm ditujukan untuk kemudahan dalam proses pembuatan tatakan telur itu sendiri. Tiap ukuran tatakan telur disesuaikan dengan kondisi

minimal dan maksimal dari jenis ukuran telur, sehingga didapatkan ruang yang lebih efisien pada tatakan telur.

3. Desain Base Telur dengan Kamera

   Desan base telur dengan ketinggian kamera didesain berdasarkan rentang kemampuan kamera dalam membaca objek, hal tersebut ditujukan selain untuk pemrosesan citra yang lebih optimal juga disesuaikan dengan kemampuan posisi sendi robot ketika melakukan proses pengambilan telur, dan diharapkan tidak terjadi *crash* antara manipulator robot dengan kamera itu sendiri.

b. Perancangan Elektronik

   Perancangan elektronik merupakan bagian dari perancangan minimum sistem sebagai kontroler utama manipulator robot, sensor, aktuator, driver motor dan I/O dari rancang bangun proyek akhir ini. Diharapkan ketika membangun sebuah rancangan elektronik dari minimum system atau shield minimum system, port I/O telah disusun sesuai dengan kebutuhan seperti kabel servo yang memiliki 3 kabel (gnd, vcc, data), maka desain dari minimum system telah menepatkan 3 pin header yang tersambung pada vcc, gnd dan data dari mikrokontroller.

2. Perancangan Perangkat Lunak

a. Pengolahan citra dengan Algoritma Codebook

Perancangan perangkat lunak Pengolahan citra dengan Algoritma Codebook menggunakan visual studio 2013 yang telah diintegrasikan dengan opencv ditujukan untuk menunjang proses pengolahan citra telur. Pada proyek akhir ini pengolahan citra telur difungsikan sebagai input untuk mencari panjang dan lebar dari suatu telur yang akan disortir. Dengan memperhatikan fungsi kinerja dari pengolah citra tersebut, dapat disimpulkan bahwa penggunaan 1 kamera masih kurang untuk akurasi ukuran panjang dan lebar dari suatu telur. Untuk dapat menentukan panjang dan lebar dengan akurasi yang baik dan benar, dibutuhkan suatu model kamera yang dipasang secara *stereo* atau 2 buah kamera yang saling terintegrasi agar ukuran panjang dan lebar yang terdeteksi lebih akurat.

b. Sistem logika fuzzy

Pembuatan sistem cerdas berbasis logika fuzzy yang telah dibuat pada proyek akhir ini dapat disimpukan bahwa validitas dari nilai hasil fuzzy memiliki persentase akurasi sebesar 84% dalam memilah ukuran telur, hal tersebut dibuktikan dengan data total hasil dari perbandingan antara *judgement exepert* dengan logika fuzzy yang dibuat dalam tiap klasifikasi ukuran telur yang mana data dari SangatKecil memiliki persentase akurasi sebesar 80%, klasifikasi Kecil sebesar 80%, klasifikasi Sedang sebesar

80%, klasifikasi Besar sebesar 80% dan klasifikasi Sangat Besar sebesar 100% dengan total persentase sebesar 84%.

3. Unjuk Kerja dari Prototype dan Implementasi Penyortir Telur Dengan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom*(DOF) berdasarkan hasil pengujian dapat berfungsi dengan optimal baik dari segi sistem cerdas yang berbasis logika fuzzy maupun manipulator robot saat melakukan proses penyortiran. Hasil pengujian pada tiap klasifikasinya dapat disimpulkan bahwa sistem pengolahan citra dengan logika fuzzy memiliki persentase rata-rata keberhasilan 84%. Sedangkan untuk penempatan telur dengan robot mengacu pada sistem Logika Fuzzy sebesar 92%, penempatan telur dengan robot mengacu pada *judgement expert* memiliki keberhasilan sebesar 76%.

B. Keterbatasan Alat

"Prototipe dan Implementasi Penyortir Telur dengan Logika Fuzzy pada Manipulator 6-*Degree Of Freedom*(DOF)" memiliki keterbatasan yang akan dipaparkan sebagai berikut :

1. Konstruksi Mekanik manipulator robot kurang memiliki ketahan sambungan antar *Degree of Freedom* sehingga rangka robot memiliki momen elastisitas terlalu besar. Diharapkan kerangka dari manipulator robot memiliki bahan dan metode sambungan *Degree of Freedom* yang lebih *rigid* seperti menggunakan bahan *stainless steel* yang disusun secara rapi tiap *Degree of Freedom*.

2. Perancangan Elektronik port I/O yang kurang efisien sehingga terdapat banyak kabel *jumper* disekitaran *shield* mikrokontroller, diharapkan agar lebih mengefisienkan dan mengoptimalkan struktur port I/O yang telah disediakan dari ATmega2560.
3. Pembacaan ukuran telur berdasarkan pengolahan citra hanya menggunakan satu kamera webcam yang menyebabkan kurangnya kepresisian dalam pembacaan, mengingat sudut baca dari kamera ke object sangat mempengaruhi. Sistem pembacaan tersebut akan lebih bagus dan presisi bilamana kamera yang digunakan lebih dari 1(Stereo vision) untuk mendapatkan sudut pembacaan lebih maksimal.

C. Penelitian Lanjutan

Berdasarkan hasil dari proyek akhir tersebut, penulis mengakui masih terdapat banyak kekurangan dari proyek akhir ini karena keterbatasan materi, kemampuan dan waktu, sehingga penulis menyarankan untuk melakukan penelitian lanjutan sebagai berikut :

1. Untuk mendapatkan kontruksi mekanik dari manipulator robot, akan lebih bagus jika menggunakan bahan dasar yang kuat seperti aluminium ataupun *stainless steel* dengan memaksimalkan sambungan antara *Degree of Freedom* terhadap aktuator servo secara langsung sehingga momen elastisitasnya dapat dikurangi dan lebih *rigid*.
2. Struktur perancangan elektronik akan lebih bagus jika penempatan port I/O disusun secara effisien dengan mempertimbangkan segala aspek kebutuhan seperti aktuator servo, driver motor maupun sensor.

3. Pembacaan ukuran telur akan lebih optimal bilamana kamera yang digunakan berjumlah lebih dari satu, tujuannya untuk mengkonfigurasikan kamera menjadi *stereo vision* sehingga sudut dari object dapat terbaca dengan optimal dan menghasilakn pembacaan ukuran yang lebih valid.
4. Sistem yang dibuat pada proyek akhir ini masih bersifat mendasar karena hanya ukuran yang diambil sebagai parameternya, untuk kedepan akan lebih baik lagi jika pembacaan tidak hanya sekedar mencari ukuran, namun kondisi kesegaran telur, warna telur juga harus dapat dijadikan sebagai parameter kualitas telur.

DAFTAR PUSTAKA

Dick dan Carey. (2010). Metode ADDIE. Diambil pada tanggal 18 Desember 2016, dari Http://en.wikipedia.org/wiki/ADDIE_Model

Elektronika Dasar.(2017). Sensor Cahaya. Dari http://elektronikadasar.info/sensor-cahaya.htm. Diambil pada tanggal 28 April 2017

Endra Pitowarno(2006). Robotika Desain, Kontrol dan Kecerdasan Buatan.Yogyakarta: Andi

Kulinologi Indonesia. Menentukan Mutu Telur. Dari http://kulinologi.co.id/acrobat/index1.php?view&id=900. Diambil pada tanggal 7 Desember 2016,

Levardy.(2015). Pengertian, Prinsip Kerja, dan Jenis-Jenis Motor DC. Dari http://any.web.id/pengertian-prinsip-kerja-dan-jenis-jenis-motor-dc.info. Diambil pada tanggal 18 Desember 2016

Sri Kusumadewi. (2003). Artificial Intelligence. Yogyakarta: Graha Ilmu

Sri Kusumadewi dan Hari Purnomo. (2013). Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu

Syahrul Awaluddin dan Dessy Irmawati.(2016). Pengolahan Citra Untuk Identifikasi Ukuran Telur. Electronics, Informatics and Vocational Education(ELINVO), Volume 1, Nomor 3, November 2016

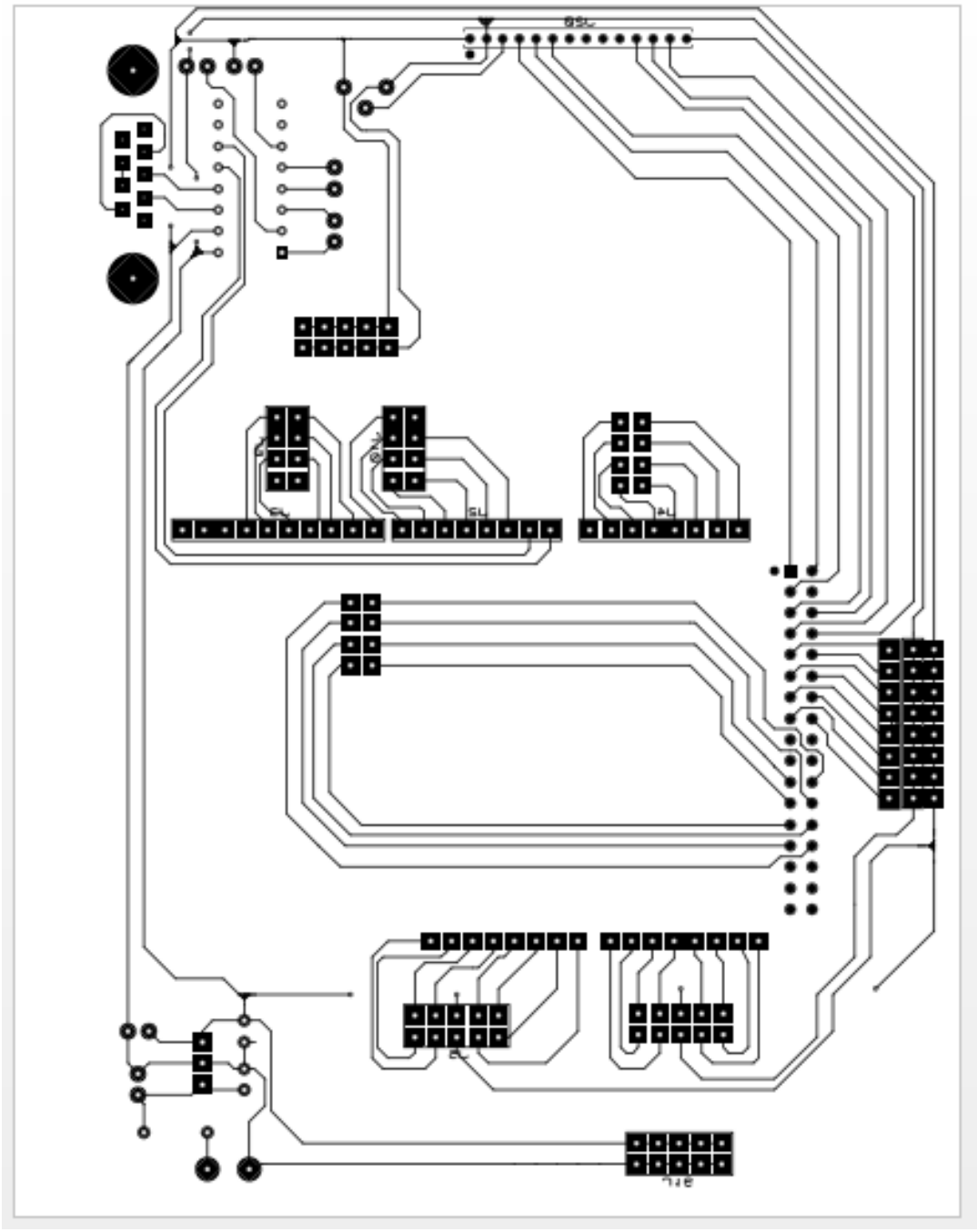
Widodo Budiharto dan Djoko purwanto.(2015). Robot Vision Teknik Membangun Robot Cerdas Masa Depan(edisi Revisi). Yogyakarta: Andi

.

LAMPIRAN

LAMPIRAN 1

Layout Minimum System Shield ATmega 2560

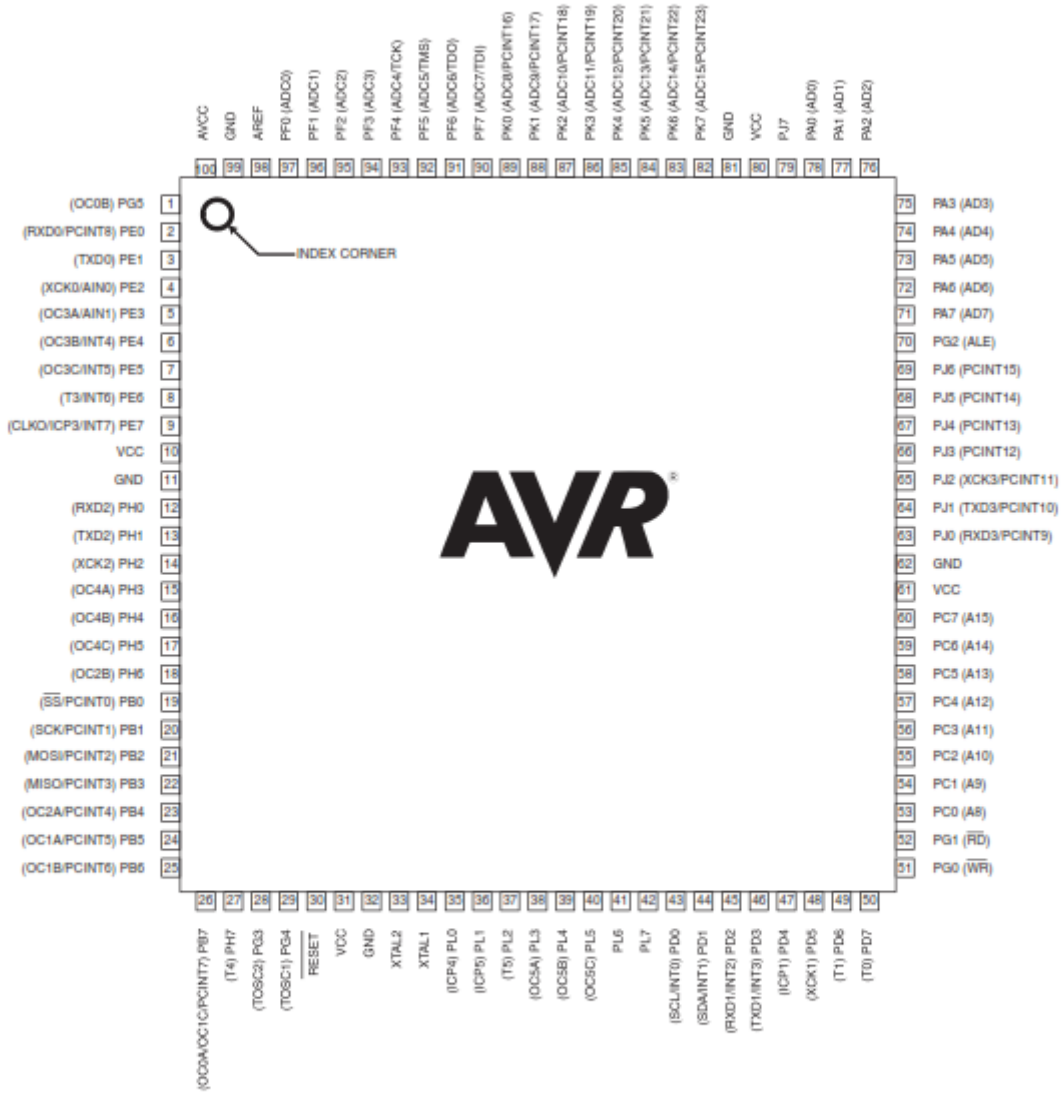


LAMPIRAN 2

Datasheet ATmega 2560

1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560



Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V

8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash

SUMMARY

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Fully Green
- Temperature Range:
  - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
  - Active Mode: 1MHz, 1.8V: 500µA
  - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
  - ATmega640V/ATmega1280V/ATmega1281V:
    - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega2560V/ATmega2561V:
    - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega640/ATmega1280/ATmega1281:
    - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
  - ATmega2560/ATmega2561:
    - 0 - 16MHz @ 4.5V - 5.5V

2549QS-AVR-02/2014

Figure 1-2. CBGA-pinout ATmega640/1280/2560

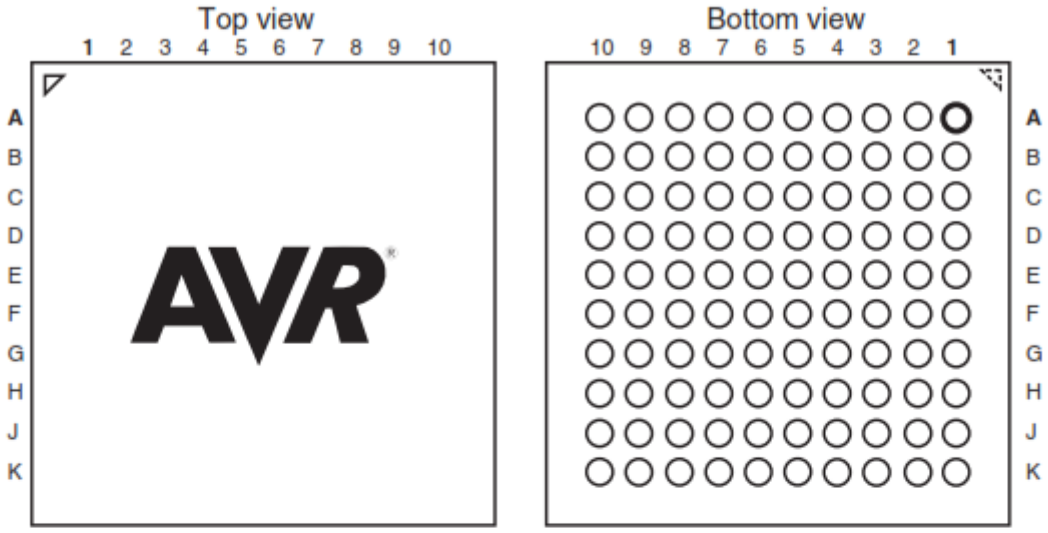


Table 1-1. CBGA-pinout ATmega640/1280/2560

	1	2	3	4	5	6	7	8	9	10
A	GND	AREF	PF0	PF2	PF5	PK0	PK3	PK6	GND	VCC
B	AVCC	PG5	PF1	PF3	PF6	PK1	PK4	PK7	PA0	PA2
C	PE2	PE0	PE1	PF4	PF7	PK2	PK5	PJ7	PA1	PA3
D	PE3	PE4	PE5	PE6	PH2	PA4	PA5	PA6	PA7	PG2
E	PE7	PH0	PH1	PH3	PH5	PJ6	PJ5	PJ4	PJ3	PJ2
F	VCC	PH4	PH6	PB0	PL4	PD1	PJ1	PJ0	PC7	GND
G	GND	PB1	PB2	PB5	PL2	PD0	PD5	PC5	PC6	VCC
H	PB3	PB4	RESET	PL1	PL3	PL7	PD4	PC4	PC3	PC2
J	PH7	PG3	PB6	PL0	XTAL2	PL6	PD3	PC1	PC0	PG1
K	PB7	PG4	VCC	GND	XTAL1	PL5	PD2	PD6	PD7	PG0

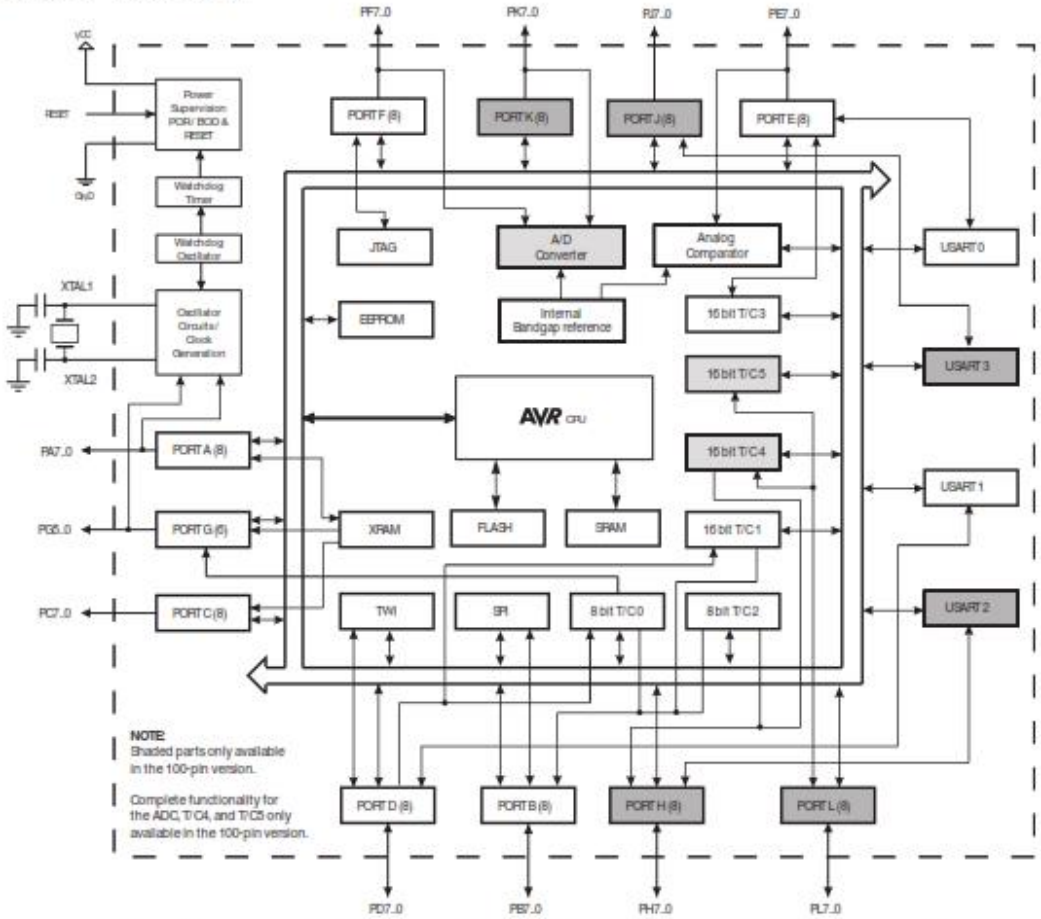
Note: The functions for each pin is the same as for the 100 pin packages shown in Figure 1-1 on page 2.

2. Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

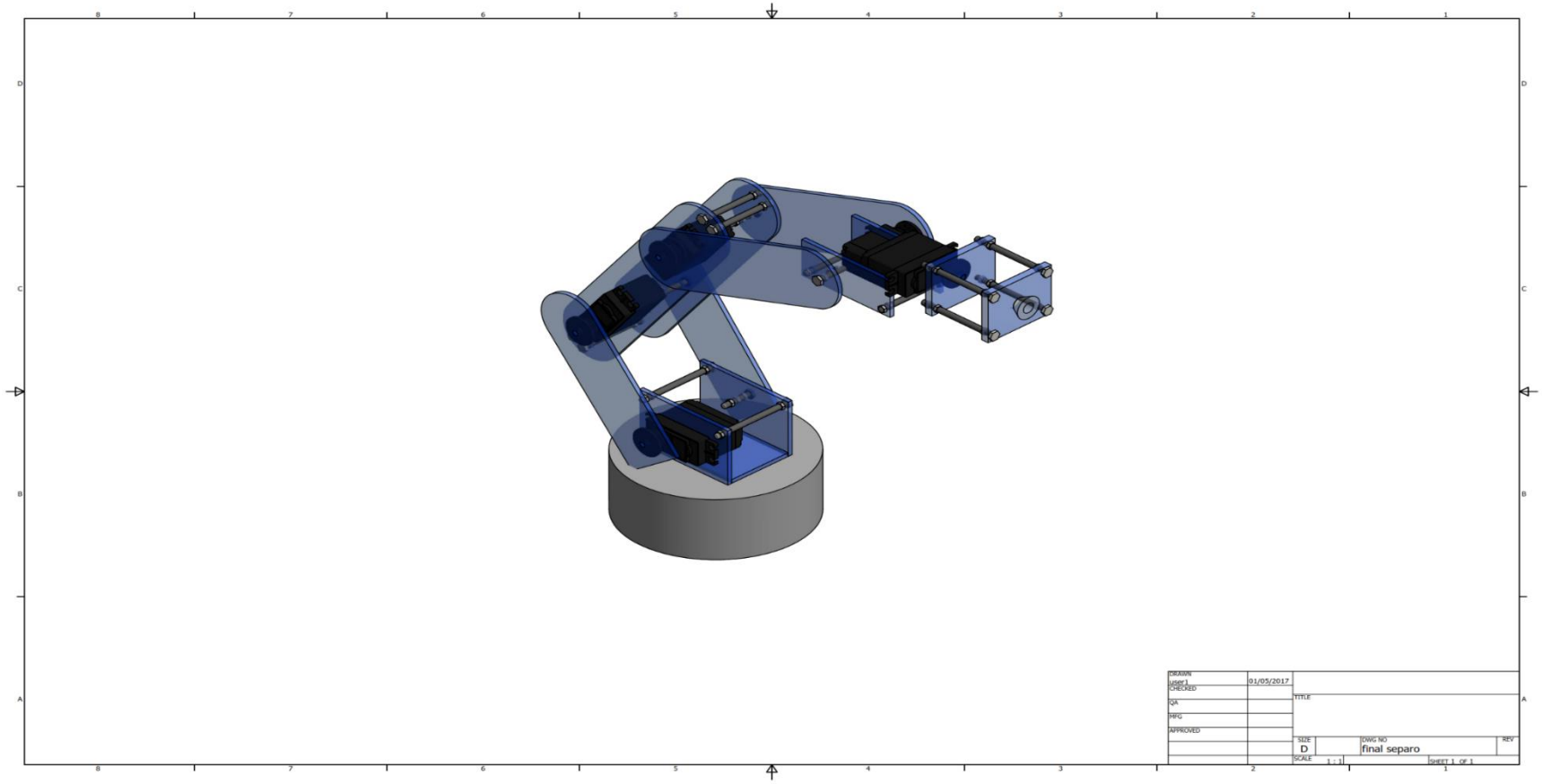
Figure 2-1. Block Diagram



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

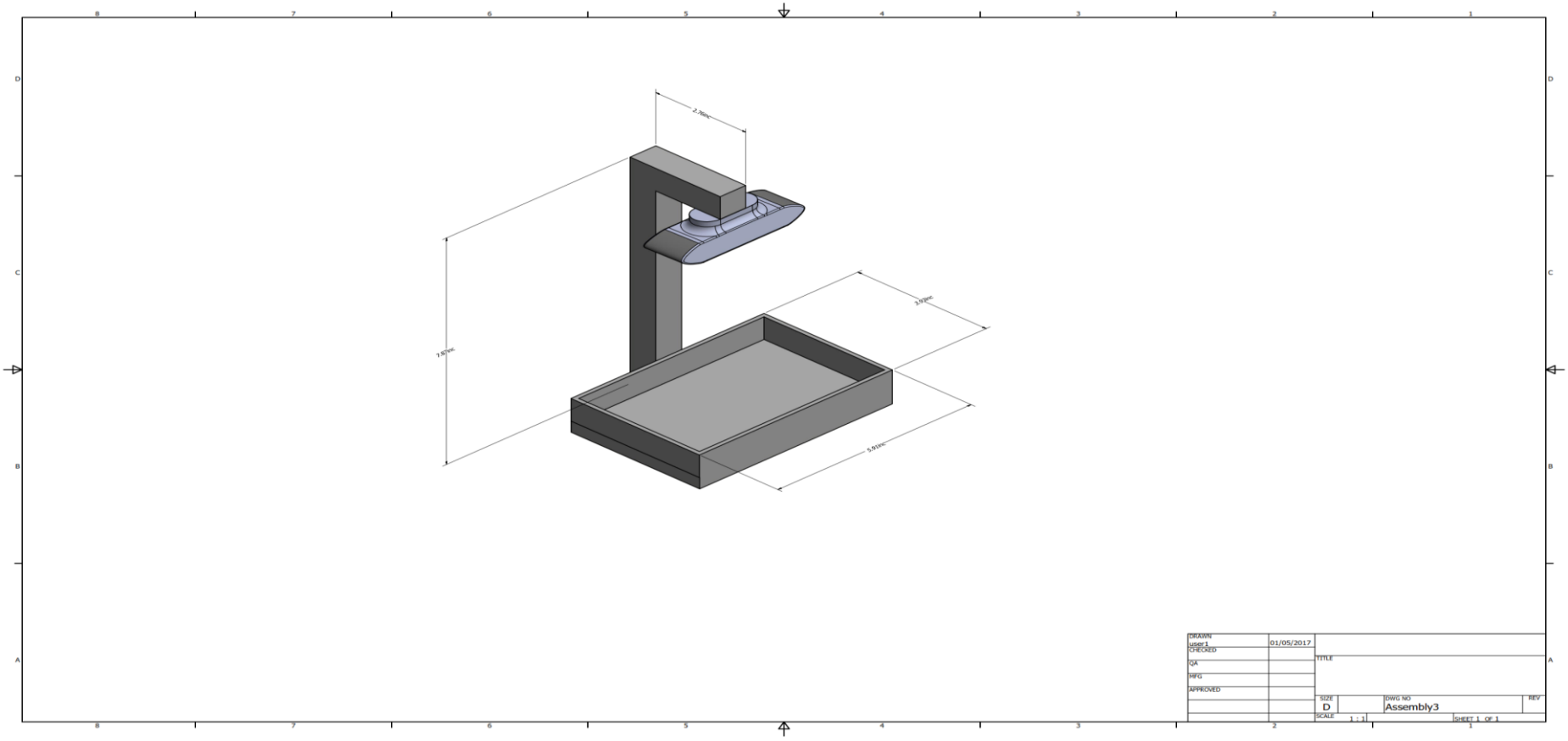
LAMPIRAN 3

Manipulator Robot Desain



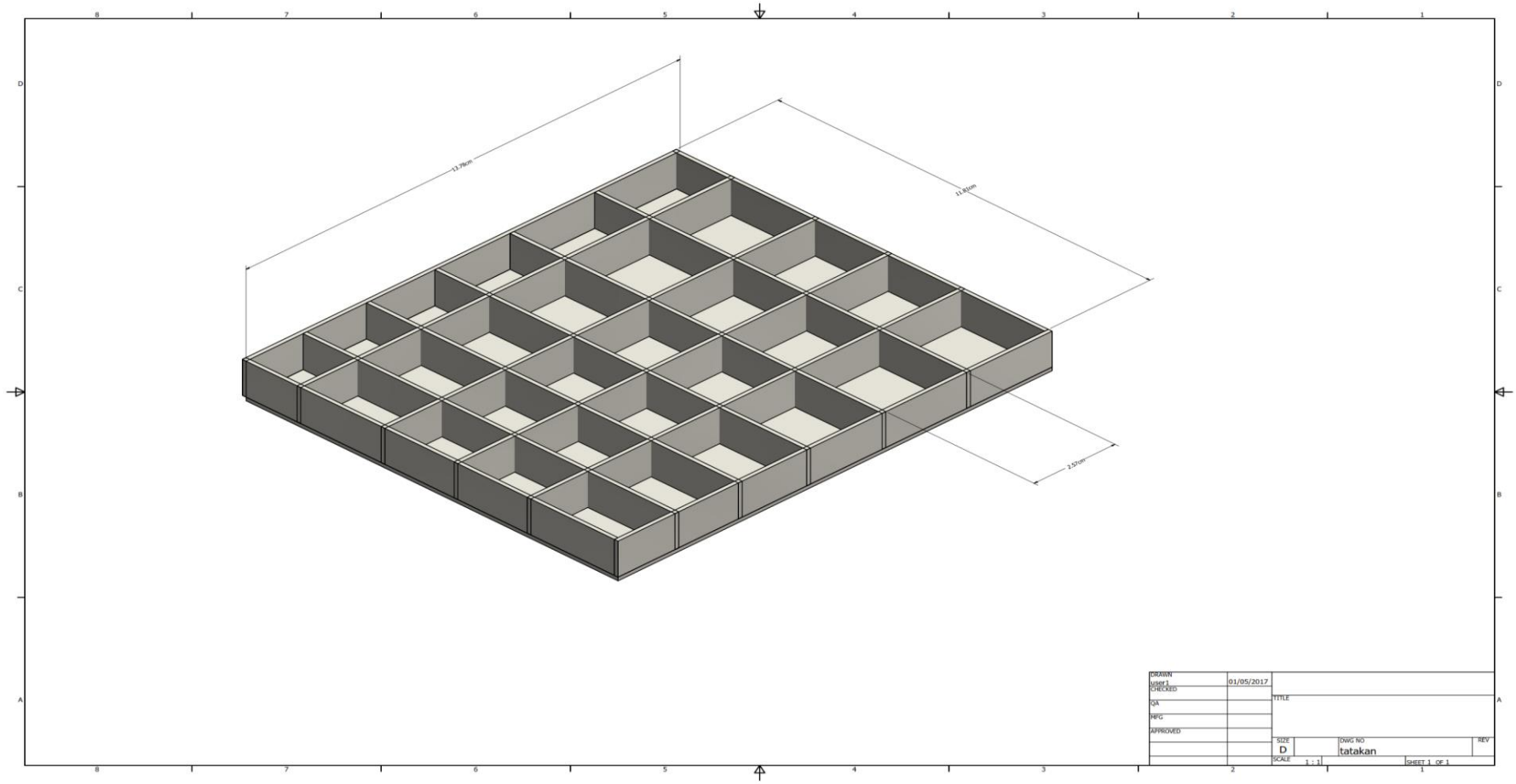
LAMPIRAN 4

Stand Kamera Desain



LAMPIRAN 5

Tatakan Telur Desain



LAMPIRAN 6

Source Code Image Processing dan Logika Fuzzy

```
#include <stdio.h>
#include <opencv\cvaux.h>
#include <opencv\cxmisc.h>
#include <opencv\highgui.h>
#include <stdlib.h>
#include <ctype.h>
#include "tserial.h"
#include "bot_control.h"
#include <Windows.h>
#include <math.h>

using namespace cv;
using namespace std;

CvBGCodeBookModel* model = 0;
const int NCHANNELS = 3;
bool ch[NCHANNELS] = { true, true, true };

#include <stdio.h>
#include <conio.h>

int i, j;
float Height, Width, A, kuadrat_A, kuadrat_B, B, luas, keliling;
float x, y, u_panjang[3], u_lebar[3], u[3][3], validasi, pembilang, penyebut, output;
//float z1 = 1.5, z2 = 1.5, z3 = 1.5, z4 = 1.5, z5 = 1.5, a1 = 8.75, a2 = 6.5, a3 = 4.5, a4 = 2.5, a5 = 0.75;
//rumus Z = A x T / 2, rumus a =(nilai minimum Membership function+(A x T / 2))//
//float z1 = 2, z2 = 1, z3 = 1.25, z4 = 0.75, z5 = 2, a1 = 8.39, a2 = 6.25, a3 = 5.3, a4 = 4, a5 = 1.06;
float z1 = 7.75, z2 = 1, z3 = 1.25, z4 = 0.75, z5 = 2, a1 = 8.39, a2 = 6.25, a3 = 5.3, a4 = 4, a5 = 1.06;

char karakter;
```

```
serial comm;

void fuzzyfikasi_panjang()
{
	x = Height;		//u_panjang[0] = kecil;	u_panjang[1] = sedang; u_panjang[2] = besar;
	// a=13		b=15	c=17
	if (x < 4.7)
	{
		u_panjang[0] = 1;			//(4.8 - x) / 0.8;
		u_panjang[1] = 0;
		u_panjang[2] = 0;
	}
	else if (x >= 4.7 && x < 5)
	{
		u_panjang[0] = (5 - x) / 0.5;		//(c-x)/(c-b)
		u_panjang[1] = (x - 4.7) / 0.3; //(x-a)/(b-a)
		u_panjang[2] = 0;
	}
	else if (x >= 5 && x < 5.3)
	{
		u_panjang[0] = 0;
		u_panjang[1] = (5.3 - x) / 0.3;
		u_panjang[2] = (x - 5) / 0.5;
	}
	else
	{
		u_panjang[0] = 0;
		u_panjang[1] = 0;
		u_panjang[2] = 1;
	}
}

void fuzzyfikasi_lebar()
{
	y = Width;
	if (y < 3.7)
```

```
	{
		u_lebar[0] = 1;		// (3.8 - y) / 0.8;
		u_lebar[1] = 0;
		u_lebar[2] = 0;
	}
	else if (y >= 3.7 && y < 4)
	{
		u_lebar[0] = (4 - y) / 0.5;
		u_lebar[1] = (y - 3.7) / 0.3;
		u_lebar[2] = 0;
	}
	else if (y >= 4 && y <= 4.3)
	{
		u_lebar[0] = 0;
		u_lebar[1] = (4.3 - y) / 0.3;
		u_lebar[2] = (y - 4) / 0.5;
	}
	else
	{
		u_lebar[0] = 0;
		u_lebar[1] = 0;
		u_lebar[2] = 1;
	}
}


void fuzzyfikasi()
{
	fuzzyfikasi_panjang();
	fuzzyfikasi_lebar();
}

void rule()
{
	for (i = 0; i < 3; i++)
	{
```

```
			for (j = 0; j < 3; j++)
			{
				if (u_panjang[i] < u_lebar[j])
				{
					u[i][j] = u_panjang[i];
				}
				else
				{
					u[i][j] = u_lebar[j];
				}
			}
		}
}

void defuzzyfikasi()
{
		pembilang =
			(u[0][0] * a5*z5) + (u[0][1] * a4*z4) + (u[0][2] * a3*z3) +
			(u[1][0] * a4*z4) + (u[1][1] * a3*z3) + (u[1][2] * a2*z2) +
			(u[2][0] * a3*z3) + (u[2][1] * a2*z2) + (u[2][2] * a1*z1);

		penyebut =
			(u[0][0] * z5) + (u[0][1] * z4) + (u[0][2] * z3) +
			(u[1][0] * z4) + (u[1][1] * z3) + (u[1][2] * z2) +
			(u[2][0] * z3) + (u[2][1] * z2) + (u[2][2] * z1);

		validasi = pembilang / penyebut;
		output = validasi;
}


int main(int argc, char** argv)
{

start:
		IplImage* rawImage = 0, *yuvImage = 0, *EqualHist;
```

```
IplImage* ImaskCodeBook = 0, *ImaskCodeBookCC = 0;
CvCapture* capture = 0;
int nframes = 0;
int nframesToLearnBG = 100;
int waktu0 = 0, waktu1 = 0, waktu2 = 0, waktu3 = 0, waktu4 = 0,waktu_global=0;
char key;

CvMemStorage* memStorage = NULL;
IplImage* contoursImage = 0;
CvSeq* contours = 0;
CvRect box;

comm.startDevice("COM3", 9600);

model = cvCreateBGCodeBookModel();
model->modMin[0] = 5;
model->modMin[1] = model->modMin[2] = 5;
model->modMax[0] = 10;
model->modMax[1] = model->modMax[2] = 10;
model->cbBounds[0] = model->cbBounds[1] = model->cbBounds[2] = 10;

capture = cvCaptureFromCAM(1);

for (;;)
{
	rawImage = cvQueryFrame(capture);
	nframes++;
	//printf("nframes = %d \n", nframes);
	if (nframes == 1 && rawImage)
	{
		yuvImage = cvCloneImage(rawImage);
		ImaskCodeBook = cvCreateImage(cvGetSize(rawImage), IPL_DEPTH_8U, 1);
		ImaskCodeBookCC = cvCreateImage(cvGetSize(rawImage), IPL_DEPTH_8U, 1);
		cvSet(ImaskCodeBook, cvScalar(255));
		//printf("nframes = %d", nframes);
	}
```

```
		if (rawImage)
		{
			cvCvtColor(rawImage, yuvImage, CV_BGR2YCrCb);
			if (nframes < nframesToLearnBG) cvBGCodeBookUpdate(model, yuvImage);

			if (nframes == nframesToLearnBG) cvBGCodeBookClearStale(model, model->t / 2);

			if (nframes > nframesToLearnBG)
			{
				cvBGCodeBookDiff(model, yuvImage, ImaskCodeBook);
				cvCopy(ImaskCodeBook, ImaskCodeBookCC);
				cvSegmentFGMask(ImaskCodeBookCC);
				if (contoursImage == NULL)
				{
					contoursImage = cvCreateImage(cvGetSize(yuvImage), IPL_DEPTH_8U, 1);
					memStorage = cvCreateMemStorage(0);
				}
				else
				{
					cvClearMemStorage(memStorage);
				}
				cvCopy(ImaskCodeBookCC, contoursImage);
				cvFindContours(contoursImage, memStorage, &contours);
				cvZero(contoursImage);
				if (contours)
				{
					cvDrawContours(rawImage, contours, cvScalarAll(100), cvScalarAll(255), 100);
					box = cvBoundingRect(contours, 1);
					cvRectangle(rawImage, cvPoint(box.x, box.y),cvPoint(box.x + box.width, box.y +
box.height), cvScalar(255, 0, 0), 3, 8, 0);
					Height = (box.height*0.026458) - 3.5;    //sk/k = 3.15        //s =3.25            //b =
3.5          //sb = 3.7
					Width = (box.width*0.026458) - 3.1;             //sk/k = 2.85        //s =2.95
      //b = 3.1           //sb = 3.3
					printf("Height = %.1f \n", Height);
					printf("Width = %.1f \n\n", Width);
```

```
// Menghitung luas dan keliling ellips A = jari jari terpendek, B jari-jari terpanjang//
/*A = Height / 2;
B = Width / 2;
kuadrat_A = pow(0.5*A, 2);
kuadrat_B = pow(0.5*B, 2);
luas = 3.14*A*B;
keliling = 3.14*(sqrtl(2*(kuadrat_A + kuadrat_B)));

printf("Height = %f \n", Height);
printf("Width = %f \n", Width);
printf("Hasil Luas = %f \n", luas);
printf("Hasil Keliling= %f \n\n", keliling);*/
waktu_global++;

if (waktu_global >= 50)
{
	fuzzyfikasi();
	rule();
	defuzzyfikasi();

	printf("Hasil = %.2f \n\n", output);

	if (output >= 0 && output < 3)
	{
		printf("Sangat Kecil \n\n");
		printf("confirm? Y/N \n");
		scanf("%c", &karakter);
		if (karakter == 'y' || karakter == 'Y')
			{
				comm.send_data('N');
				comm.stopDevice();
				printf(" DATA TERKIRIM \n");
				Sleep(20000);
				goto start;
			}
```

```
	if (karakter == 'n' || karakter == 'N')
		{
			comm.stopDevice();
			Sleep(500);
			goto start;
		}
}

else if (output >= 2.5 && output < 4.5)
{
	printf("Kecil \n\n");
	printf("confirm? Y/N \n");
	scanf("%c", &karakter);
	if (karakter == 'y' || karakter == 'Y')
		{
			for (int i = 1; i <= 2; i++)
			{
				comm.send_data('M');
				if (i == 2)
				{
					comm.stopDevice();
					printf(" DATA TERKIRIM \n");
				}
			}
			Sleep(20000);
			goto start;
		}
	if (karakter == 'n' || karakter == 'N')
		{
			comm.stopDevice();
			Sleep(500);
			goto start;
		}
}
else if (output >= 4 && output < 6.5)
{
```

```
		printf("sedang \n\n");
		printf("confirm? Y/N \n");
		scanf("%c", &karakter);
		if (karakter == 'y' || karakter == 'Y')
			{
				for (int i = 1; i <= 3; i++)
				{
					comm.send_data('L');
					if (i == 3)
					{
						comm.stopDevice();
						printf(" DATA TERKIRIM \n");
					}
				}
				Sleep(20000);
				goto start;
			}
		if (karakter == 'n' || karakter == 'N')
			{
				comm.stopDevice();
				Sleep(500);
				goto start;
			}
	}
	else if (output >= 6.5 && output < 8)
	{
		printf("Besar \n\n");
		printf("confirm? Y/N \n");
		scanf("%c", &karakter);
		if (karakter == 'y' || karakter == 'Y')
			{
				for (int i = 1; i <= 4; i++)
				{
					comm.send_data('K');
					if (i == 4)
					{
```

```
                                        comm.stopDevice();
                                        printf(" DATA TERKIRIM \n");
                                }
                        }
                        Sleep(20000);
                        goto start;
                }
        if (karakter == 'n' || karakter == 'N')
                {
                        comm.stopDevice();
                        Sleep(500);
                        goto start;
                }
}
else if (output >= 7.5 && output < 9)
{
        printf("sangat Besar \n\n");
        printf("confirm? Y/N \n");
        scanf("%c", &karakter);
        if (karakter == 'y' || karakter == 'Y')
                {
                        for (int i = 1; i <= 5; i++)
                        {
                                comm.send_data('J');
                                if (i == 5)
                                {
                                        comm.stopDevice();
                                        printf(" DATA TERKIRIM \n");
                                }
                        }
                        Sleep(20000);
                        goto start;
                }
        if (karakter == 'n' || karakter == 'N')
                {
                        comm.stopDevice();
```

```
                                                        Sleep(500);
                                                        goto start;
                                                }
                                        }
                                }
                        }
                }

                cvShowImage("Raw", rawImage);
                cvShowImage("ForegroundCodeBook", ImaskCodeBook);
                cvShowImage("CodeBook_ConnectComp", ImaskCodeBookCC);
        }
        key = cvWaitKey(10);
        if (char(key) == 'q')
        {
                comm.stopDevice();
                goto start;
        }
        if (char(key) == 27){ break; }
}
cvReleaseCapture(&capture);
cvDestroyWindow("Raw");
cvDestroyWindow("ForegroundCodeBook");
cvDestroyWindow("CodeBook_ConnectComp");
return 0;
```

LAMPIRAN 7

Source Code Manipulator Robot

```
#include <mega2560.h>
#include <delay.h>
#include <stdio.h>
#include <alcd.h>
#include <math.h>
#define relay  ((PINL & (1<<1))!=0)
#define proximity ((PINC & (1<<7))!=0)
#define servo1 PORTC.0
#define servo2 PORTC.4
#define servo3 PORTC.1
#define servo4 PORTC.3
#define servo5 PORTC.2
#define servo6 PORTC.5
#define t_atas   ((PINL & (1<<6))!=0)
#define t_bawah  ((PINL & (1<<5))!=0)
#define t_masuk  ((PINL & (1<<4))!=0)
#define t_keluar ((PINL & (1<<7))!=0)
#define LS ((PINL & (1<<2))!=0)

#define ADC_VREF_TYPE 0x20 //20
```

```
#ifndef RXB8
#define RXB8 1
#endif
#ifndef TXB8
#define TXB8 0
#endif
#ifndef UPE
#define UPE 2
#endif
#ifndef DOR
#define DOR 3
#endif
#ifndef FE
#define FE 4
#endif
#ifndef UDRE
#define UDRE 5
#endif
#ifndef RXC
#define RXC 7
#endif


#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
```

```
 #define DATA_OVERRUN (1<<DOR)
 #define DATA_REGISTER_EMPTY (1<<UDRE)
define RX_COMPLETE (1<<RXC)

 // USART0 Receiver buffer
 #define RX_BUFFER_SIZE0 8
 char rx_buffer0[RX_BUFFER_SIZE0];

 #if RX_BUFFER_SIZE0 <= 256
 unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;
 #else
 unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;
 #endif

 char buffer[30],buffer1[30];
 unsigned char data=0,data1=0;

 unsigned int x,servo_data0, servo_data1,servo_data2,servo_data3,servo_data4,servo_data5;
 int counter_sk=0,counter_k=0,counter_s=0,counter_b=0,counter_sb=0;

 unsigned char read_adc(unsigned char adc_input);
 void standby(); void standby_ambil(); void sdh_ambil(); void standby_sensor(); void standby_drop();
 void sk_1(); void sk_2(); void sk_3(); void sk_4(); void sk_5();
 void k_1(); void k_2(); void k_3(); void k_4(); void k_5();
```

```
void s_1(); void s_2(); void s_3(); void s_4(); void s_5();
void b_1(); void b_2(); void b_3(); void b_4(); void b_5();
void sb_1(); void sb_2(); void sb_3(); void sb_4(); void sb_5()

// ==================== MOTION ================= //

void motion_1()
{
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("MOTION 1");                              delay_ms(500);
      while(1)
      {
            lcd_gotoxy(0,0);lcd_putsf("STANDBY");
            standby();                                                              delay_ms(1000);
            for(servo_data1=115;servo_data1>=34;servo_data1--) { delay_ms(20); }     delay_ms(500);

      tengah:
            PORTL |=(1<<1);     /*vacuum_on 1*/                                     delay_ms(500);

            standby_ambil();                                                        delay_ms(1000);
            for(servo_data2=135 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}    delay_ms(1000);

            lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("AMBIL");
            sdh_ambil();                                                            delay_ms(1000);
            for(servo_data1=34;servo_data1<=67;servo_data1++) { delay_ms(20); }      delay_ms(500);
```

```
            standby_sensor();                                                   delay_ms(1500);
            data=read_adc(0);
            data1=read_adc(1);
            lcd_gotoxy(0,1);sprintf(buffer1,"NILAI = %d %d",data1,data); lcd_puts(buffer1);

      if(counter_sk == 0 && data>=10 && data<=255 && !proximity || counter_sk == 0 && data1>=10 && data1<=255 && !
proximity)
                {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat kecil 1");
            counter_sk = counter_sk + 1;                                        delay_ms(1000);
            standby_drop();                                                     delay_ms(1000);
            sk_5();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 1");                goto exit;
        }
      if(counter_sk == 1 && data>=10 && data<=255 && !proximity || counter_sk == 1 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat kecil 2");
            counter_sk = counter_sk + 1;                                        delay_ms(1000);
            standby_drop();                                                     delay_ms(1000);
            sk_4();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 2");                goto exit;
        }
```

```
      if(counter_sk == 2 && data>=10 && data<=255 && !proximity || counter_sk == 2 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat kecil 3");
            counter_sk = counter_sk + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sk_3();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 3");            goto exit;
        }
      if(counter_sk == 3 && data>=10 && data<=255 && !proximity || counter_sk == 3 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat kecil 4");
            counter_sk = counter_sk + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sk_2();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 4");            goto exit;
        }
      if(counter_sk == 4 && data>=10 && data<=255 && !proximity || counter_sk == 4 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat kecil 5");
```

```
            counter_sk = counter_sk + 1;                                        delay_ms(1000);
            standby_drop();                                                     delay_ms(1000);
            sk_1();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 5");                goto exit;
        }
      else if(counter_sk >= 5)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("TIDAK MUAT");                delay_ms(2000); standby_ambil();
            delay_ms(1000); for(servo_data2=120 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}    delay_ms(1000);
            PORTL &=~(1<<1);   /* vacuum_off 0*/
goto exit;
        }
                goto tengah;
      }
      exit :
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("KELUAR");                   delay_ms(500);
 }


 void motion_2()
 {
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("MOTION 2");                          delay_ms(500);
      while (1)
      {
            lcd_gotoxy(0,0);lcd_putsf("STANDBY");
            standby();                                                          delay_ms(1000);
            for(servo_data1=115;servo_data1>=34;servo_data1--) { delay_ms(20); } delay_ms(500);
```

```
tengah:
      PORTL |=(1<<1);     /*vacuum_on 1*/                                    delay_ms(500);

      standby_ambil();                                                       delay_ms(1000);
      for(servo_data2=135 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}   delay_ms(1000);

      lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("AMBIL");
      sdh_ambil();                                                           delay_ms(1000);
      for(servo_data1=34;servo_data1<=67;servo_data1++) { delay_ms(20); }     delay_ms(500);
      standby_sensor();                                                      delay_ms(1500);
      data=read_adc(0);
      data1=read_adc(1);
      lcd_gotoxy(0,1);sprintf(buffer1,"NILAI = %d %d",data1,data); lcd_puts(buffer1);

   if(counter_k == 0 && data>=10 && data<=255 && !proximity || counter_k == 0 && data1>=10 && data1<=255 && !
proximity)
            {
      lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
      lcd_clear();lcd_gotoxy(0,1); lcd_putsf("kecil 1");
      counter_k = counter_k + 1;                                          delay_ms(1000);
      standby_drop();                                                      delay_ms(1000);
      k_5();                                                              delay_ms(1000);
      lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 1");                 goto exit;
    }
```

```
      if(counter_k == 1 && data>=10 && data<=255 && !proximity || counter_k == 1 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("kecil 2");
            counter_k = counter_k + 1;                                          delay_ms(1000);
            standby_drop();                                                      delay_ms(1000);
            k_4();                                                              delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 2");                 goto exit;
        }
      if(counter_k == 2 && data>=10 && data<=255 && !proximity || counter_k == 2 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("kecil 3");
            counter_k = counter_k + 1;                                          delay_ms(1000);
            standby_drop();                                                      delay_ms(1000);
            k_3();                                                              delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 3");                 goto exit;
        }
      if(counter_k == 3 && data>=10 && data<=255 && !proximity || counter_k == 3 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("kecil 4");
```

```
            counter_k = counter_k + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            k_2();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 4");                  goto exit;
        }
      if(counter_k == 4 && data>=10 && data<=255 && !proximity || counter_k == 4 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("kecil 5");
            counter_k = counter_k + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            k_1();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 5");                  goto exit;
        }
      else if(counter_k >= 5)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("TIDAK MUAT");                     delay_ms(2000); standby_ambil();
            delay_ms(1000); for(servo_data2=120 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}   delay_ms(1000); PORTL
            &=~(1<<1);    /* vacuum_off 0*/
                      goto exit;
        }
                    goto tengah;
      }
      exit :
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("KELUAR");                     delay_ms(500);
```

```
}

void motion_3()
{
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("MOTION 3");                                  delay_ms(500);
      while (1)
      {
            lcd_gotoxy(0,0);lcd_putsf("STANDBY");
            standby();                                                                  delay_ms(1000);
            for(servo_data1=115;servo_data1>=34;servo_data1--) { delay_ms(20); }         delay_ms(500);

      tengah:
            PORTL |=(1<<1);      /*vacuum_on 1*/                                        delay_ms(500);

            standby_ambil();                                                            delay_ms(1000);
            for(servo_data2=135 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}        delay_ms(1000);

            lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("AMBIL");
            sdh_ambil();                                                                delay_ms(1000);
            for(servo_data1=34;servo_data1<=67;servo_data1++) { delay_ms(20); }          delay_ms(500);
            standby_sensor();                                                           delay_ms(1500);
            data=read_adc(0);
            data1=read_adc(1);
            lcd_gotoxy(0,1);sprintf(buffer1,"NILAI = %d %d",data1,data); lcd_puts(buffer1);
```

```
      if(counter_s == 0 && data>=10 && data<=255 && !proximity || counter_s == 0 && data1>=10 && data1<=255 && !
proximity)
                  {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sedang 1");
            counter_s = counter_s + 1;                                          delay_ms(1000);
            standby_drop();                                                      delay_ms(1000);
            s_5();                                                              delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 1");                 goto exit;
        }
      if(counter_s == 1 && data>=10 && data<=255 && !proximity || counter_s == 1 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sedang 2");
            counter_s = counter_s + 1;                                          delay_ms(1000);
            standby_drop();                                                      delay_ms(1000);
            s_4();                                                              delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 2");                 goto exit;
        }
      if(counter_s == 2 && data>=10 && data<=255 && !proximity || counter_s == 2 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sedang 3");
```

```
            counter_s = counter_s + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            s_3();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 3");                  goto exit;
        }
      if(counter_s == 3 && data>=10 && data<=255 && !proximity || counter_s == 3 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sedang 4");
            counter_s = counter_s + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            s_2();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 4");                  goto exit;
        }
      if(counter_s == 4 && data>=10 && data<=255 && !proximity || counter_s == 4 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sedang 5");
            counter_s = counter_s + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            s_1();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 5");                  goto exit;
        }
```

```
      else if(counter_s >= 5)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("TIDAK MUAT");                    delay_ms(2000); standby_ambil();
            delay_ms(1000); for(servo_data2=120 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}  delay_ms(1000); PORTL
            &=~(1<<1);   /* vacuum_off 0*/
            goto exit;
        }
                goto tengah;
      }
      exit :
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("KELUAR");                       delay_ms(500);
}

void motion_4()
{
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("MOTION 4");                              delay_ms(500);
      while (1)
      {
            lcd_gotoxy(0,0);lcd_putsf("STANDBY");
            standby();                                                              delay_ms(1000);
            for(servo_data1=115;servo_data1>=34;servo_data1--) { delay_ms(20); }     delay_ms(500);

      tengah:
            PORTL |=(1<<1);      /*vacuum_on 1*/                                    delay_ms(500);

            standby_ambil();                                                        delay_ms(1000);
```

```
            for(servo_data2=135 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}    delay_ms(1000);

            lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("AMBIL");
            sdh_ambil();                                                            delay_ms(1000);
            for(servo_data1=34;servo_data1<=67;servo_data1++) { delay_ms(20); }      delay_ms(500);
            standby_sensor();                                                       delay_ms(1500);
            data=read_adc(0);
            data1=read_adc(1);
            lcd_gotoxy(0,1);sprintf(buffer1,"NILAI = %d %d",data1,data); lcd_puts(buffer1);

      if(counter_b == 0 && data>=10 && data<=255 && !proximity || counter_b == 0 && data1>=10 && data1<=255 && !
proximity)
                  {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("besar 1");
            counter_b = counter_b + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            b_5();                                                               delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 1");                  goto exit;
        }
      if(counter_b == 1 && data>=10 && data<=255 && !proximity || counter_b == 1 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("besar 2");
```

```
            counter_b = counter_b + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            b_4();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 2");                  goto exit;
        }
      if(counter_b == 2 && data>=10 && data<=255 && !proximity || counter_b == 2 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("besar 3");
            counter_b = counter_b + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            b_3();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 3");                  goto exit;
        }
      if(counter_b == 3 && data>=10 && data<=255 && !proximity || counter_b == 3 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("besar 4");
            counter_b = counter_b + 1;                                          delay_ms(1000);
            standby_drop();                                                       delay_ms(1000);
            b_2();                                                             delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 4");                  goto exit;
        }
```

```
      if(counter_b == 4 && data>=10 && data<=255 && !proximity || counter_b == 4 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("besar 5");
            counter_b = counter_b + 1;                                          delay_ms(1000);
            standby_drop();                                                      delay_ms(1000);
            b_1();                                                               delay_ms(1000);lcd_clear();lcd_gotoxy(0,0);
            lcd_putsf("selesai 5");                  goto exit;
        }
      else if(counter_b >= 5)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("TIDAK MUAT");                  delay_ms(2000); standby_ambil();
            delay_ms(1000); for(servo_data2=120 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}    delay_ms(1000);
            PORTL &=~(1<<1);   /* vacuum_off 0*/
            goto exit;
        }
                goto tengah;
      }
      exit :
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("KELUAR");                    delay_ms(500);
}

void motion_5()
{
    lcd_clear();lcd_gotoxy(0,0);lcd_putsf("MOTION 5");                           delay_ms(500);
      while (1)
```

```
{
      lcd_gotoxy(0,0);lcd_putsf("STANDBY");
      standby();                                                          delay_ms(1000);
      for(servo_data1=115;servo_data1>=34;servo_data1--) { delay_ms(20); }    delay_ms(500);

tengah:
      PORTL |=(1<<1);     /*vacuum_on 1*/                                 delay_ms(500);

      standby_ambil();                                                    delay_ms(1000);
      for(servo_data2=135 ;servo_data2>=100 ;servo_data2--) {delay_ms(30);}   delay_ms(1000);

      lcd_clear(); lcd_gotoxy(0,0); lcd_putsf("AMBIL");
      sdh_ambil();                                                        delay_ms(1000);
      for(servo_data1=34;servo_data1<=67;servo_data1++) { delay_ms(20); }     delay_ms(500);
      standby_sensor();                                                   delay_ms(1500);
      data=read_adc(0);
      data1=read_adc(1);
      lcd_gotoxy(0,1);sprintf(buffer1,"NILAI = %d %d",data1,data); lcd_puts(buffer1);

if(counter_sb == 0 && data>=10 && data<=255 && !proximity || counter_sb == 0 && data1>=10 && data1<=255 && !
proximity)
          {
      lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
      lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat besar 1");
      counter_sb = counter_sb + 1;                                        delay_ms(1000);
```

```
            standby_drop();                                                 delay_ms(1000);
            sb_5();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 1");            goto exit;
        }
      if(counter_sb == 1 && data>=10 && data<=255 && !proximity || counter_sb == 1 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat besar 2");
            counter_sb = counter_sb + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sb_4();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 2");            goto exit;
        }
      if(counter_sb == 2 && data>=10 && data<=255 && !proximity || counter_sb == 2 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat besar 3");
            counter_sb = counter_sb + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sb_3();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 3");            goto exit;
        }
      if(counter_sb == 3 && data>=10 && data<=255 && !proximity || counter_sb == 3 && data1>=10 && data1<=255 && !
```

```
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat besar 4");
            counter_sb = counter_sb + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sb_2();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 4");            goto exit;
        }
      if(counter_sb == 4 && data>=10 && data<=255 && !proximity || counter_sb == 4 && data1>=10 && data1<=255 && !
proximity)
        {
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("telur");
            lcd_clear();lcd_gotoxy(0,1); lcd_putsf("sangat besar 5");
            counter_sb = counter_sb + 1;                                    delay_ms(1000);
            standby_drop();                                                 delay_ms(1000);
            sb_1();                                                         delay_ms(1000);
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("selesai 5");            goto exit;
        }
      else if(counter_sb >= 5)
        {
            lcd_clear();lcd_gotoxy(0,0);  lcd_putsf("TIDAK  MUAT");                    delay_ms(2000);  standby_ambil();
            delay_ms(1000);  for(servo_data2=120  ;servo_data2>=100  ;servo_data2--)  {delay_ms(30);}     delay_ms(1000);  PORTL
            &=~(1<<1);    /* vacuum_off 0*/
            goto exit;
        }
```

```
                    goto tengah;
      }
      exit :
            lcd_clear();lcd_gotoxy(0,0); lcd_putsf("KELUAR");                    delay_ms(500);
}


// ================ MOTION AWAL ================== //
void standby ()
{
    //orientasi robot standby hadap tatakan telur
    //servo_data2=120;
    servo_data0=40;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(400);
    servo_data3=50;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data4=180;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    delay_ms(100);
    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data5=80;        // arm vacuum 'kanan 180 kiri 40'
    delay_ms(500);
    servo_data1=115;    //base 'kanan 40 kiri 180'
}

void standby_ambil()
{
```

```
    servo_data1=33;    //base 'kanan 40 kiri 180'
    delay_ms(500);
    servo_data0=115;        // no.2 atas dari base 'kiri 40 kanan 180
    delay_ms(300);
    servo_data3=90;      // no.3 atas dari base 'kiri 180 kanan 40'
    servo_data2=135;      // no.1 atas dari base 'kiri 180 kanan 40'
    delay_ms(150);
    servo_data4=148;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    delay_ms(300);
    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'
}

void sdh_ambil()
{
    servo_data4=40;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180' servo_data3=90;      //
    no.3 atas dari base 'kiri 180 kanan 40' servo_data2=150;      // no.1 atas dari base 'kiri 180
    kanan 40' servo_data0=160;        // no.2 atas dari base 'kiri 40 kanan 180 servo_data1=33;
    //base 'kanan 40 kiri 180'
    delay_ms(500);
    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'
}

void standby_sensor()
{
    servo_data0=105;        // no.2 atas dari base 'kanan 40 kiri 180
    //for(servo_data0=160;servo_data0>=100;servo_data1--) { delay_ms(30); }    //delay_ms(500);
```

```
    delay_ms(200);
    //for(servo_data2=150;servo_data2>=120;servo_data1--) { delay_ms(30); }    //delay_ms(500);
    servo_data2=122;      // no.1 atas dari base 'kanan 180 kiri 40'
    delay_ms(300);
    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data4=151;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    servo_data1=67;    //base 'kanan 40 kiri 180'
    servo_data5=80;                              //delay_ms(300);

}

void standby_drop()
{
    servo_data4=45;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    delay_ms(1500);
    servo_data3=50;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=160;      // no.1 atas dari base 'kanan 180 kiri 40'
    delay_ms(250);
    servo_data0=150;        // no.2 atas dari base 'kiri 40 kanan 180
    delay_ms(450);
    for(servo_data1=67;servo_data1<=115;servo_data1++) { delay_ms(20); }    delay_ms(450);
    //servo_data1=135;    //base 'kanan 40 kiri 180'
    servo_data5=80;
}
```

```
// ================== MOTION SANGAT KECIL ================== //


void sk_1()

{

    for(servo_data1=115;servo_data1<=130;servo_data1++) { delay_ms(20); }     delay_ms(1000); servo_data4=50;        //
    arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70) servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);



    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=140;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kiri 40 kanan 180 delay_ms(150);



    servo_data3=85;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);



    servo_data3=68;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=125;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);



    servo_data3=90;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=115;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(500);



    servo_data3=100;      // no.3 atas dari base 'kanan 180 kiri 40'

    delay_ms(200);

    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=100;        // no.2 atas dari base 'kanan 40 kiri 180

    //delay_ms(500);
```

```
    //servo_data1=153;    //base 'kanan 40 kiri 180'
    delay_ms(1500);
    for(servo_data4=70;servo_data4<=130;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                      delay_ms(1500);

}

void sk_2()
{
    for(servo_data1=115;servo_data1<=135;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=90;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=115;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=125;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    //servo_data1=157;    //base 'kanan 40 kiri 180' servo_data5=140;        // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);
    for(servo_data4=50;servo_data4<=120;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                      delay_ms(1500);
}

void sk_3()
{
    for(servo_data1=115;servo_data1<=137;servo_data1++) { delay_ms(20); }     delay_ms(1000);
```

```
    servo_data4=50;         // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)

    delay_ms(500);

    servo_data3=68;       // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=125;       // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=125;         // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    //servo_data1=160;    //base 'kanan 40 kiri 180' servo_data5=140;         // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);

    for(servo_data4=50;servo_data4<=107;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                     delay_ms(1500);

}


void sk_4()

{

    //servo_data1=170;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1<=143;servo_data1++) {
    delay_ms(20); }     delay_ms(1000); servo_data4=70;         // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);

    servo_data3=85;       // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=130;       // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=125;         // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    servo_data5=140;         // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);

    for(servo_data4=70;servo_data4<=145;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                     delay_ms(1500);

}
```

```
void sk_5()

{

    //for(servo_data1=135;servo_data1<=200;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    for(servo_data1=115;servo_data1<=150;servo_data1++) { delay_ms(20); }     delay_ms(1000); delay_ms(500);

    servo_data4=70;         // arm no.2 dari ujung vacum 'kanan 40 kiri 180'

    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=140;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=125;        // no.2 atas dari base 'kiri 40 kanan 180

    delay_ms(500);

    servo_data5=80;

    for(servo_data4=70;servo_data4<=130;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}




// ================ motion kecil ============================== //


void k_1()

{

    for(servo_data1=115;servo_data1<=127;servo_data1++) { delay_ms(20); }     delay_ms(1000); servo_data4=50;        //
    arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70) servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);
```

```
    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=140;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kiri 40 kanan 180 delay_ms(150);


    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=135;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);


    servo_data3=63;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=120;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);


    servo_data3=90;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=120;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=130;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(500);


    servo_data3=100;      // no.3 atas dari base 'kanan 180 kiri 40'

    delay_ms(200);

    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=100;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(1500);

    //servo_data1=153;    //base 'kanan 40 kiri 180'

    //servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    //delay_ms(1500);

    for(servo_data4=70;servo_data4<=135;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}
```

```
void k_2()
{
    for(servo_data1=115;servo_data1<=128;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data4=50;         // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=90;       // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=120;       // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=130;         // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    //servo_data1=152;    //base 'kanan 40 kiri 180' servo_data5=140;         // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);
    for(servo_data4=50;servo_data4<=133;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void k_3()
{
    for(servo_data1=115;servo_data1<=130;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data4=50;         // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=63;       // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=130;       // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=120;         // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
```

```
    //servo_data1=155;    //base 'kanan 40 kiri 180' servo_data5=140;        // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);

    for(servo_data4=50;servo_data4<=127;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}


void k_4()

{

    //servo_data1=160;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1<=135;servo_data1++) {
    delay_ms(20); }     delay_ms(1000); servo_data4=60;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);

    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=135;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=125;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);

    for(servo_data4=70;servo_data4<=150;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}


void k_5()

{

    //servo_data1=170;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1<=145;servo_data1++) {
    delay_ms(20); }     delay_ms(1000); servo_data4=70;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
```

```
    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=140;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kiri 40 kanan 180

    delay_ms(500);

    servo_data5=80;

    for(servo_data4=70;servo_data4<=147;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);


}


// ================= motion sedang ============================== //


void s_1()

{

    //for(servo_data1=135;servo_data1<=137;servo_data1++) { delay_ms(20); }     delay_ms(1000);

    servo_data1=119;

    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)

    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);


    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=143;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=128;        // no.2 atas dari base
    'kiri 40 kanan 180 delay_ms(150);


    servo_data3=75;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=137;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=130;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);
```

```
    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40' servo_data0=120;        //
    no.2 atas dari base 'kanan 40 kiri 180 delay_ms(150);


    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=130;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(500);


    servo_data3=100;      // no.3 atas dari base 'kanan 180 kiri 40'

    delay_ms(200);


    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=100;        // no.2 atas dari base 'kanan 40 kiri 180

    //delay_ms(500);

    //servo_data1=147;    //base 'kanan 40 kiri 180'

    delay_ms(1500);

    for(servo_data4=70;servo_data4<=140;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}


void s_2()

{

    //for(servo_data1=135;servo_data1<=137;servo_data1++) { delay_ms(20); }     delay_ms(1000);

    servo_data1=120;

    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
```

```
    delay_ms(500);
    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=130;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    //servo_data1=147;    //base 'kanan 40 kiri 180' servo_data5=140;        // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);
    for(servo_data4=50;servo_data4<=127;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void s_3()
{
    //for(servo_data1=135;servo_data1<=137;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data1=123;
    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=120;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    //servo_data1=147;    //base 'kanan 40 kiri 180' servo_data5=140;        // arm
    vacuum 'kanan 180 kiri 40' delay_ms(1500);
    for(servo_data4=50;servo_data4<=128;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}
```

```
void s_4()
{
    servo_data1=125;    //base 'kanan 40 kiri 180'
    //for(servo_data1=135;servo_data1<=137;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data4=60;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=75;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=137;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=130;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    servo_data5=80;        // arm vacuum 'kanan 180 kiri 40'
    delay_ms(1500);
    for(servo_data4=60;servo_data4<=150;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void s_5()
{
    servo_data1=128;    //base 'kanan 40 kiri 180'
    //for(servo_data1=135;servo_data1<=135;servo_data1++) { delay_ms(20); }     delay_ms(1000);
    servo_data4=73;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=143;      // no.1 atas dari base 'kanan 180 kiri 40'
```

```
    servo_data0=128;        // no.2 atas dari base 'kiri 40 kanan 180

    delay_ms(500);

    servo_data5=80;

    for(servo_data4=70;servo_data4<=153;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);


}


// ================ motion besar ================================ //


void b_1()

{

    servo_data1=112;    //base 'kanan 40 kiri 180'

    //for(servo_data1=135;servo_data1<=120;servo_data1++) { delay_ms(20); }     delay_ms(1000); servo_data4=50;        //
    arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70) servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);


    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=143;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kiri 40 kanan 180 delay_ms(150);


    servo_data3=75;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=137;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);


    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=115;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);
```

```
    servo_data3=85;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=125;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=130;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(500);


    servo_data3=100;      // no.3 atas dari base 'kanan 180 kiri 40'

    delay_ms(200);

    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=100;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(1500);

    //delay_ms(1500);

    for(servo_data4=70;servo_data4<=145;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}


void b_2()

{

    servo_data1=112;    //base 'kanan 40 kiri 180'

    delay_ms(1000);

    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)

    delay_ms(500);

    servo_data3=85;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=125;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=130;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);
```

```
    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'
    delay_ms(1500);
    for(servo_data4=50;servo_data4<=127;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void b_3()
{
    servo_data1=110;    //base 'kanan 40 kiri 180'
    delay_ms(1000);
    servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=115;        // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'
    delay_ms(1500);
    for(servo_data4=50;servo_data4<=133;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void b_4()
{
    servo_data1=110;    //base 'kanan 40 kiri 180'
```

```
    delay_ms(1000);
    servo_data4=60;         // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);
    servo_data3=75;       // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=137;       // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=125;         // no.2 atas dari base 'kanan 40 kiri 180
    delay_ms(500);
    servo_data5=140;         // arm vacuum 'kanan 180 kiri 40'
    delay_ms(1500);
    for(servo_data4=70;servo_data4<=153;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}

void b_5()
{
    servo_data1=107;    //base 'kanan 40 kiri 180'
    delay_ms(1000);
    servo_data4=73;         // arm no.2 dari ujung vacum 'kanan 40 kiri 180'
    servo_data3=65;       // no.3 atas dari base 'kanan 180 kiri 40'
    servo_data2=143;       // no.1 atas dari base 'kanan 180 kiri 40'
    servo_data0=125;         // no.2 atas dari base 'kiri 40 kanan 180
    delay_ms(500);
    servo_data5=80;
    for(servo_data4=70;servo_data4<=155;servo_data4++) { delay_ms(30); }      delay_ms(1500);
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
```

```
}



// ========================== motion sangat besar ================ //


void sb_1()

{

    //servo_data1=130;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1>=105;servo_data1--) {
    delay_ms(20); }     delay_ms(1000); servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);



    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=143;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=128;        // no.2 atas dari base
    'kiri 40 kanan 180 delay_ms(150);



    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=125;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);



    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=130;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=120;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(150);



    servo_data3=85;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=125;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=130;        // no.2 atas dari base
    'kanan 40 kiri 180 delay_ms(500);
```

```
    servo_data3=100;      // no.3 atas dari base 'kanan 180 kiri 40'

    delay_ms(200);

    servo_data2=100;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=100;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(1500);

    //delay_ms(1500);

    for(servo_data4=70;servo_data4<=140;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}


void sb_2()

{

    //servo_data1=127;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1>=105;servo_data1--) {
    delay_ms(20); }     delay_ms(1000); servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);

    servo_data3=85;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=125;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=130;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);

    for(servo_data4=50;servo_data4<=125;servo_data4++) { delay_ms(30); }      delay_ms(1500); PORTL &=~(1<<1);   /*
    vacuum_off 0*/                                      delay_ms(1500);

}


void sb_3()
```

```
{
    //servo_data1=123;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1>=102;servo_data1--) {
    delay_ms(20); }     delay_ms(1000); servo_data4=50;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);

    servo_data3=60;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=120;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);

    for(servo_data4=50;servo_data4<=120;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);

}


void sb_4()

{
    //servo_data1=120;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1>=98;servo_data1--) {
    delay_ms(20); }     delay_ms(1000); servo_data4=60;        // arm no.2 dari ujung vacum 'kiri 40 kanan 180' (70)
    delay_ms(500);

    servo_data3=80;      // no.3 atas dari base 'kanan 180 kiri 40'

    servo_data2=130;      // no.1 atas dari base 'kanan 180 kiri 40'

    servo_data0=125;        // no.2 atas dari base 'kanan 40 kiri 180

    delay_ms(500);

    servo_data5=140;        // arm vacuum 'kanan 180 kiri 40'

    delay_ms(1500);

    for(servo_data4=70;servo_data4<=153;servo_data4++) { delay_ms(30); }      delay_ms(1500);
```

```
    PORTL &=~(1<<1);   /* vacuum_off 0*/                                    delay_ms(1500);
}


void sb_5()
{
    //servo_data1=117;    //base 'kanan 40 kiri 180' for(servo_data1=115;servo_data1>=90;servo_data1--) {
    delay_ms(20); }     delay_ms(1000); servo_data4=73;        // arm no.2 dari ujung vacum 'kanan 40 kiri 180'

    servo_data3=65;      // no.3 atas dari base 'kanan 180 kiri 40' servo_data2=143;      //
    no.1 atas dari base 'kanan 180 kiri 40' servo_data0=128;        // no.2 atas dari base
    'kiri 40 kanan 180

    delay_ms(500);

    servo_data5=80;

    for(servo_data4=70;servo_data4<=147;servo_data4++) { delay_ms(30); }      delay_ms(1500);

    PORTL &=~(1<<1);   /* vacuum_off 0*/                                     delay_ms(1500);


}


// ========================== END MOTION ======================== //

// This flag is set on USART0 Receiver buffer overflow

bit rx_buffer_overflow0;


// USART0 Receiver interrupt service routine

interrupt [USART0_RXC] void usart0_rx_isr(void)

{

char status,data;

status=UCSR0A;
```

```
data=UDR0;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
   {
   rx_buffer0[rx_wr_index0++]=data;
#if RX_BUFFER_SIZE0 == 256
   // special case for receiver buffer size=256
   if (++rx_counter0 == 0) rx_buffer_overflow0=1;
#else
   if (rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
   if (++rx_counter0 == RX_BUFFER_SIZE0)
      { rx_counter0=0; rx_buffer_overflow0=1;
      }
#endif
   }
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART0 Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0++];
```

```
#if RX_BUFFER_SIZE0 != 256
if (rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#endif
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}
#pragma used-
#endif

// USART0 Transmitter buffer
#define TX_BUFFER_SIZE0 8
char tx_buffer0[TX_BUFFER_SIZE0];

#if TX_BUFFER_SIZE0 <= 256
unsigned char tx_wr_index0,tx_rd_index0,tx_counter0;
#else
unsigned int tx_wr_index0,tx_rd_index0,tx_counter0;
#endif

// USART0 Transmitter interrupt service routine
interrupt [USART0_TXC] void usart0_tx_isr(void)
{
if (tx_counter0)
```

```
                {
   --tx_counter0;
   UDR0=tx_buffer0[tx_rd_index0++];
#if TX_BUFFER_SIZE0 != 256
   if (tx_rd_index0 == TX_BUFFER_SIZE0) tx_rd_index0=0;
#endif
                }
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART0 Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter0 == TX_BUFFER_SIZE0);
#asm("cli")
if (tx_counter0 || ((UCSR0A & DATA_REGISTER_EMPTY)==0))
   {
   tx_buffer0[tx_wr_index0++]=c;
#if TX_BUFFER_SIZE0 != 256
   if (tx_wr_index0 == TX_BUFFER_SIZE0) tx_wr_index0=0;
#endif
   ++tx_counter0;
   }
```

```
else
                UDR0=c;
#asm("sei")
}
#pragma used-
#endif

unsigned char read_adc(unsigned char adc_input)
{
ADMUX=(adc_input & 0x1f) | (ADC_VREF_TYPE & 0xff);
if (adc_input & 0x20) ADCSRB |= 0x08;
else ADCSRB &= 0xf7;
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{

            TCNT0=39;
```

```
   x++;
    if(x>1250)
    {
        x=0;
    }
    else
    {
        if(x<servo_data0) { servo1=1;} else {servo1=0;}
        if(x<servo_data1) { servo2=1;} else {servo2=0;}
        if(x<servo_data2) { servo3=1;} else {servo3=0;}
        if(x<servo_data3) { servo4=1;} else {servo4=0;}
        if(x<servo_data4) { servo5=1;} else {servo5=0;}
        if(x<servo_data5) { servo6=1;} else {servo6=0;}
    }
}


void main(void)
{
#pragma optsize- CLKPR=0x80; CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif



PORTA=0x00; DDRA=0x00;
PORTB=0x00; DDRB=0b10000000;
PORTC=0b10000000; DDRC=0b01111111;
```

```
PORTD=0x00; DDRD=0x00;

PORTE=0x00; DDRE=0x00;

PORTF=0x00; DDRF=0x00;

PORTG=0x00; DDRG=0x00;

PORTH=0x00; DDRH=0x00;

PORTJ=0x00; DDRJ=0x00;

PORTK=0x00; DDRK=0x00;

PORTL=0b11110100; DDRL=0b00000001;

TCCR0A=0x00; TCCR0B=0x01; TCNT0=0x00;
OCR0A=0x00; OCR0B=0x00;

TCCR1A=0x00; TCCR1B=0x00; TCNT1H=0x00;
TCNT1L=0x00; ICR1H=0x00;
ICR1L=0x00;OCR1AH=0x00; OCR1AL=0x00;
OCR1BH=0x00; OCR1BL=0x00; OCR1CH=0x00;
OCR1CL=0x00;



ASSR=0x00; TCCR2A=0x00; TCCR2B=0x00;
TCNT2=0x00; OCR2A=0x00; OCR2B=0x00;



TCCR3A=0x00; TCCR3B=0x00; TCNT3H=0x00;
TCNT3L=0x00; ICR3H=0x00; ICR3L=0x00;
OCR3AH=0x00; OCR3AL=0x00; OCR3BH=0x00;
OCR3BL=0x00; OCR3CH=0x00; OCR3CL=0x00;



TCCR4A=0x00; TCCR4B=0x00; TCNT4H=0x00;
TCNT4L=0x00; ICR4H=0x00; ICR4L=0x00;
OCR4AH=0x00; OCR4AL=0x00; OCR4BH=0x00;
OCR4BL=0x00; OCR4CH=0x00; OCR4CL=0x00;



TCCR5A=0x00; TCCR5B=0x00; TCNT5H=0x00;
TCNT5L=0x00; ICR5H=0x00; ICR5L=0x00;
OCR5AH=0x00; OCR5AL=0x00; OCR5BH=0x00;
OCR5BL=0x00; OCR5CH=0x00; OCR5CL=0x00;
```

```
 EICRA=0x00; EICRB=0x00; EIMSK=0x00;


 PCMSK0=0x00;  PCMSK1=0x00;  PCMSK2=0x00;
 PCICR=0x00;


 TIMSK0=0x01;  TIMSK1=0x00;  TIMSK2=0x00;

 TIMSK3=0x00; TIMSK4=0x00; TIMSK5=0x00;


 UCSR0A=0x00; UCSR0B=0xD8; UCSR0C=0x06;

 UBRR0H=0x00;
 UBRR0L=0x67;
 UCSR1B=0x00;
 UCSR2B=0x00;UCSR3B=0x00;
 ACSR=0x80; ADCSRB=0x00;
 DIDR1=0x00;

DIDR0=0x00;

 DIDR2=0xFF; ADMUX=ADC_VREF_TYPE & 0xff;
 ADCSRA=0xA4;

 ADCSRB&=0xF8;

 SPCR=0x00;


 TWCR=0x00;

#asm("sei")


 lcd_init(16);
 lcd_putsf("===BISMILLAH===");
 delay_ms(500);

 lcd_clear();
```

```
while (1)
            {
            //cek_proxi();
      awal: standby();
         delay_ms(2000);
         lcd_gotoxy(0,0); lcd_putsf("PENYORTIR TELUR");
         lcd_gotoxy(3,1); sprintf(buffer,"%d %d %d %d %d",counter_sk,counter_k,counter_s,counter_b,counter_sb); lcd_puts
(buffer);
         PORTL &=~(1<<1);   /* vacuum_off 0*/
         //PORTL |=(1<<1);
        while(!LS)
        {
         if(getchar()=='N') {lcd_gotoxy(14,1); lcd_putsf("SK"); delay_ms(500); lcd_clear(); motion_1(); goto awal;}
         if(getchar()=='M') {lcd_gotoxy(15,1); lcd_putsf("K");  delay_ms(500); lcd_clear(); motion_2(); goto awal;}
         if(getchar()=='L') {lcd_gotoxy(15,1); lcd_putsf("S");  delay_ms(500); lcd_clear(); motion_3(); goto awal;}
         if(getchar()=='K') {lcd_gotoxy(15,1); lcd_putsf("B");  delay_ms(500); lcd_clear(); motion_4(); goto awal;}
         if(getchar()=='J') {lcd_gotoxy(14,1); lcd_putsf("SB"); delay_ms(500); lcd_clear(); motion_5(); goto awal;}
        } } };
```

LAMPIRAN 8

Source code *Graphical User Interface*

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Diagnostics;

namespace GUI_penyortir_telur
{
    public partial class Form1 : Form
    {
        double Height, Width;
        double validasi, pembilang, penyebut, output;
        double x, y;
        double[] u_panjang = new double[3];
        double[] u_lebar = new double[3];
        double[,] u = new double[3,3];
        float z1 = 7.75f; float z2 = 1; float z3 = 1.25f;
        float z4 = 0.75f; float z5 = 2; float a1 = 8.39f;
        float a2 = 6.25f; float a3 = 5.3f; float a4 = 4;
        float a5 = 1.06f;

        void fuzzyfikasi_panjang()
        {
            x = Height;           //u_panjang[0] = kecil;     u_panjang[1] = sedang; u_panjang[2] = besar;
            // a=13  b=15   c=17
            if (x < 4.7)
```

```
    {
        u_panjang[0] = 1;               //(4.8 - x) / 0.8;
        u_panjang[1] = 0;
        u_panjang[2] = 0;
    }
    else if (x >= 4.7 && x < 5)
    {
        u_panjang[0] = (5 - x) / 0.5;   //(c-x)/(c-b)
        u_panjang[1] = (x - 4.7) / 0.3; //(x-a)/(b-a)
        u_panjang[2] = 0;
    }
    else if (x >= 5 && x < 5.3)
    {
        u_panjang[0] = 0;
        u_panjang[1] = (5.3 - x) / 0.3;
        u_panjang[2] = (x - 5) / 0.5;
    }
    else
    {
        u_panjang[0] = 0;
        u_panjang[1] = 0;
        u_panjang[2] = 1;
    }
}

void fuzzyfikasi_lebar()
{
    y = Width;
    if (y < 3.7)
    {
        u_lebar[0] = 1;    // (3.8 - y) / 0.8;
        u_lebar[1] = 0;
        u_lebar[2] = 0;
    }
    else if (y >= 3.7 && y < 4)
    {
```

```
        u_lebar[0] = (4 - y) / 0.5;
        u_lebar[1] = (y - 3.7) / 0.3;
        u_lebar[2] = 0;
    }
    else if (y >= 4 && y <= 4.3)
    {
        u_lebar[0] = 0;
        u_lebar[1] = (4.3 - y) / 0.3;
        u_lebar[2] = (y - 4) / 0.5;
    }
    else
    {
        u_lebar[0] = 0;
        u_lebar[1] = 0;
        u_lebar[2] = 1;
    }
}


void fuzzyfikasi()
{
    fuzzyfikasi_panjang();
    fuzzyfikasi_lebar();
}

void rule()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (u_panjang[i] < u_lebar[j])
            {
                u[i,j] = u_panjang[i];
            }
            else
```

```
            {
                u[i,j] = u_lebar[j];
            }
        }
    }
}

void defuzzyfikasi()
{
    pembilang =
        (u[0,0] * a5 * z1) + (u[0,1] * a4 * z4) + (u[0,2] * a3 * z3) +
        (u[1,0] * a4 * z4) + (u[1,1] * a3 * z3) + (u[1,2] * a2 * z2) +
        (u[2,0] * a3 * z3) + (u[2,1] * a2 * z2) + (u[2,2] * a1 * z5);

    penyebut =
        (u[0,0] * z1) + (u[0,1] * z4) + (u[0,2] * z3) +
        (u[1,0] * z4) + (u[1,1] * z3) + (u[1,2] * z2) +
        (u[2,0] * z3) + (u[2,1] * z2) + (u[2,2] * z5);

    validasi = pembilang / penyebut;
    output = validasi;
}

public Form1()
{
    InitializeComponent();
}
private void radioButton1_Click(object sender, EventArgs e)
{
    progressBar1.Visible = false;
    comboBox1.Enabled = false;
    comboBox2.Enabled = false;
    button1.Enabled = false;
    comboBox1.Items.Clear();
    button6.Enabled = true;
    textBox3.Enabled = true;
```

```
    textBox4.Enabled = true;
    button4.Visible = false;
    textBox1.Visible = false;
    textBox2.Visible = false;
    label4.Visible = false;
    label5.Visible = false;
    button3.Visible = false;
    button5.Visible = false;
    label9.Visible = false;
    label10.Visible = false;
    if(button2.Enabled==true)
    {
        MessageBox.Show("COM Port Masih terbuka, silahkan tutup terlebih dahulu","PERINGATAN");
        button6.Enabled = false;
        radioButton2.Enabled = false;
        button1.Enabled = false;
    }
}
private void radioButton2_Click(object sender, EventArgs e)
{

    button6.Enabled = false;
    textBox3.Enabled = false;
    textBox4.Enabled = false;
    comboBox1.Enabled = true;
    comboBox2.Enabled = true;
    button1.Enabled = true;
    availableport();
}

void availableport()
{
    String[] ports = SerialPort.GetPortNames();
    comboBox1.Items.AddRange(ports);
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        if(comboBox1.Text=="")
        {
            label1.Text = "Pilih Port";
        }
        else if(comboBox2.Text=="")
        {
            label2.Text = "Masukkan BaudRate";
        }
        else
        {

            serialPort1.PortName = comboBox1.Text;
            serialPort1.BaudRate = Convert.ToInt32(comboBox2.Text);
            serialPort1.Open();
            progressBar1.Value = 100;
            button1.Enabled = false;
            button2.Enabled = true;
            button3.Enabled = true;
            button5.Enabled = true;
            button6.Enabled = false;
            button3.Visible = true;
            button5.Visible = true;
            textBox1.Enabled = true;
            textBox2.Enabled = true;
            textBox1.Visible = true;
            textBox2.Visible = true;
            progressBar1.Visible = true;
            label4.Visible = true;
            label5.Visible = true;
            label9.Visible = true;
            label10.Visible = true;
        }
```

```
    }
        catch(UnauthorizedAccessException)
    {
        label7.Visible = true;
        label7.Text = "Unauthorized Allocation";
    }
}

private void button2_Click(object sender, EventArgs e)
{
    serialPort1.Close();
    progressBar1.Value = 0;
    button1.Enabled = true;
    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
    button5.Enabled = false;
    button3.Visible = false;
    button5.Visible = false;
    textBox1.Enabled = false;
    textBox2.Enabled = false;
    textBox1.Visible = false;
    textBox2.Visible = false;
    progressBar1.Visible = false;
    label4.Visible = false;
    label5.Visible = false;
    button6.Enabled = true;
    radioButton2.Enabled = true;
}

private void button3_Click(object sender, EventArgs e)
{
    Height = double.Parse(textBox1.Text);
    Width = double.Parse(textBox2.Text);

        fuzzyfikasi();
```

```
rule();
defuzzyfikasi();
if (output >= 0 && output < 3)
{
    label6.Visible = true;
    //label6.Text = output.ToString();
    label6.Text = "kategori ";
    label8.Visible = true;
    label8.Text = "Sangat Kecil";
    button4.Visible = true;
    button4.Enabled = true;

}
else if (output >= 2.5 && output < 4.5)
{
    label6.Visible = true;
    label6.Text = "kategori ";
    label8.Visible = true;
    label8.Text = "Kecil";
    button4.Visible = true;
    button4.Enabled = true;

}
else if (output >= 4 && output < 6.5)
{
    label6.Visible = true;
    label6.Text = "kategori ";
    label8.Visible = true;
    label8.Text = "Sedang";
    button4.Visible = true;
    button4.Enabled = true;

}
else if (output >= 6.5 && output < 8)
{
    label6.Visible = true;
```

```
            label6.Text = "kategori ";
            label8.Visible = true;
            label8.Text = "Besar";
            button4.Visible = true;
            button4.Enabled = true;
        }

        else if (output >= 7.5 && output < 9)
        {
            label6.Visible = true;
            label6.Text = "kategori ";
            label8.Visible = true;
            label8.Text = "Sangat Besar";
            button4.Visible = true;
            button4.Enabled = true;

        }

}

private void button5_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    label8.Visible = false;
    label6.Visible = false;
    button4.Enabled = false;
}

private void button4_Click(object sender, EventArgs e)
{
    if (output >= 7.5 && output < 9)
    {
        for (int i = 1; i <= 5; i++)
        {
            serialPort1.WriteLine("J/");
```

```
                }
            }
            else if (output >= 6.5 && output < 8)
            {
                for (int i = 1; i <= 4; i++)
                {
                    serialPort1.WriteLine("K/");
                }

            }
            else if (output >= 4 && output < 6.5)
            {

                for (int i = 1; i <= 3; i++)
                {
                    serialPort1.Write("L/");
                }
            }
            else if (output >= 2.5 && output < 4.5)
            {
                for (int i = 1; i <= 2; i++)
                {
                    serialPort1.Write("M/");
                }

            }
            else if (output >= 0 && output < 3)
            {
                serialPort1.Write("N/");
            }
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if(MessageBox.Show("Apakah Anda Yakin Ingin Keluar","Keluar",MessageBoxButtons.YesNo,MessageBoxIcon.Question)
== System.Windows.Forms.DialogResult.Yes)
```

```
            {
                Application.Exit();
            }

        }

        private void clearToolStripMenuItem_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
            textBox2.Text = "";
            comboBox1.Text= "";
            comboBox2.Text = "";
            comboBox1.Items.Clear();
            serialPort1.Close();
            label8.Visible = false;
            label6.Visible = false;
            button2.Enabled = false;
            button3.Enabled = false;
            button4.Enabled = false;
            button5.Enabled = false;
            progressBar1.Value = 0;
            radioButton1.PerformClick();
        }

        private void aboutUsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            MessageBox.Show(" PROTOTYPE DAN IMPLEMENTASI PENYORTIR TELUR\nBERBASIS LOGIKA FUZZY PADA MANIPULATOR 6 DOF\n\n
Ver 1.0","About us");
        }

        private void button6_Click(object sender, EventArgs e)
        {
            OpenFileDialog data = new OpenFileDialog();
            if (data.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                textBox3.Text = data.FileName;
```

```
        textBox4.Text = data.SafeFileName;
        string filepath = data.FileName;
        Process.Start(filepath);
    }
}


private void textBox1_TextChanged(object sender, EventArgs e)
{
    if (Height >= 6 || Height <= 4)
    {
        label8.Visible = true;
        label8.Text = "PERHATIAN Input Panjang 4-6cm";
    }
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
if (Width >= 5 || Width <= 3)
    {
        label8.Visible = true;
        label8.Text = "PERHATIAN Input Lebar 3-5";
    }
}
```