



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

RANCANG BANGUN LENGAN ROBOT 3 DOF BERBASIS ATMEGA328 UNTUK MEMINDAHKAN BISKUIT

Hendry Erwantono

NRP 2213030033

Evi Siska Safarina

NRP 2213030049

Dosen Pembimbing

Ir. Hanny Boedinugroho, MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

DESIGN 3 DOF ROBOTIC ARM BASED ON ATMEGA328 TO REPLACE BISCUIT

Hendry Erwantono

NRP 2213030033

Evi Siska Safarina

NRP 2213030049

Advisor

Ir. Hanny Boedinugroho, M.T.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM

Faculty of Industrial Technology

Institut Teknologi Sepuluh Nopember

Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Rancang Bangun Lengan Robot Berbasis ATmega328 Untuk Memindahkan Biskuit”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Mahasiswa I



Hendry Erwantono
NRP 2213030033

Surabaya, 02 Juni 2016

Mahasiswa II



Evi Siska Safarina
NRP 2213030049

RANCANG BANGUN LENGAN ROBOT 3 DOF BERBASIS ATMEGA328 UNTUK MEMINDAHKAN BISKUIT

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik
Pada
Bidang Studi Komputer Kontrol
Program Studi D3 Teknik Elektro
Institut Teknologi Sepuluh Nopember

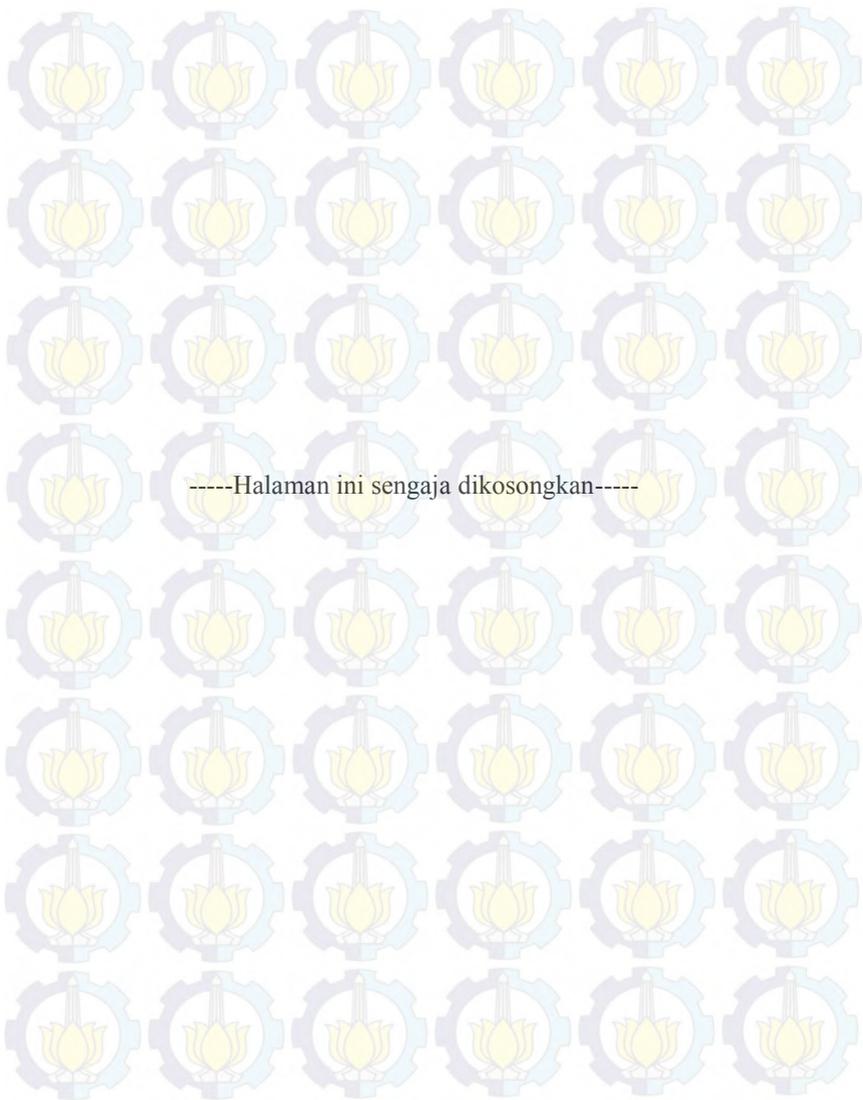


Menyetujui:

Dosen Pembimbing

Ir. Hanny Boedinugroho, MT.
NIP. 1961 07 06 1987 01 1001

SURABAYA
JUNI, 2016



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

RANCANG BANGUN LENGAN ROBOT 3 DOF BERBASIS ATMEGA328 UNTUK MEMINDAHKAN BISKUIT

Nama Mahasiswa : Hendry Erwantono
NRP : 2213030033
Nama Mahasiswa : Evi Siska Safarina
NRP : 2213030049
Dosen Pembimbing : Ir. Hanny Boedinugroho, MT.
NIP : 19610706 198701 1 001

ABSTRAK

Pada produksi biskuit, proses pemindahan biskuit dari garis antrian ke kaleng biskuit masih menggunakan tenaga manusia. Dari kasus tersebut, diperlukan suatu alat yang dapat memindahkan biskuit-biskuit dari garis antrian ke kemasan secara otomatis. Dengan demikian, dapat memacu jumlah produksi meski dengan jumlah pegawai yang sama.

Oleh karena itu, pada tugas akhir ini dibuat suatu lengan robot berbasis ATmega328 3 DOF untuk memindahkan biskuit-biskuit tersebut ke mika di dalam kaleng biskuit. Pada alat ini juga digunakan sensor pendeteksi benda yang terdiri dari LED merah yang memancarkan gelombang cahaya merah dan fotodiode yang berfungsi sebagai penerima gelombang cahaya merah. Metode yang digunakan untuk menentukan posisi akhir dari lengan robot adalah dengan memasukan sudut masing-masing motor *servo*. Jika posisi akhir lengan robot tidak sesuai, maka dilakukan pemrograman ulang untuk mengatur nilai sudut masing-masing motor *servo*. Jika program berjalan dengan benar, maka lengan robot dapat mengambil biskuit dari ujung *conveyor* ke mika biskuit. Dengan demikian, seorang pegawai dapat mengoperasikan beberapa lengan biskuit.

Hasil dari pengujian yang dilakukan, didapatkan persentase biskuit coklat muda yang tidak terambil adalah 67,857% dan biskuit coklat tua yang tidak terambil adalah 53,333%. Dengan demikian, sistem yang dibuat kurang sesuai karena lebih dari 10% biskuit yang tidak terambil dari ujung *conveyor*.

Kata Kunci : ATmega328, motor *servo*, Arduino, 3 DOF

DESIGN 3 DOF ROBOTIC ARM BASED ON ATMEGA328 TO REPLACE BISCUIT

Name : Hendry Erwantono
NRP : 2213030033
Name : Evi Siska Safarina
NRP : 2213030049
Advisor : Ir. Hany Boedinugroho, M.T.
NIP : 19610706 198701 1 001

ABSTRACT

In the production of biscuits, removal process of biscuits from the waiting line to the biscuit cans still use a human power. From these cases, we need a device to remove automatically the biscuits from the queue to a packaging line . Thus, it can boost the production with the same number of employee.

Therefore, this final project created a robot arm based ATmega328 3 DOF to move the biscuits into the mica of biscuit cans. In this device, also used object detection sensor that consists of red LEDs to emit red light and a photodiode as a receiver red light waves. The method used to determine the final position of the robot arm by including the angle of each servo motor. If the final position of the robot arm is not appropriate, then reprogramming to set the angle of each servo motor. If the program runs correctly, the robot arm can take a biscuit from the end of the conveyor to mica of biscuits cans. Therefore, an employee can operate multiple arm biscuits.

The results of test performed, the percentage of brown biscuits is not fetched about 67.857% the dark brown biscuits that are not fetched is 53.333%. Therefore, the system is less suitable for more than 10% of biscuits that are not picked up from the end of the conveyor.

Keywords : ATmega328, motor servo, Arduino, 3 DOF

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN LENGAN ROBOT 3 DOF BERBASIS ATMEGA328 UNTUK MEMINDAHKAN BISKUIT

Dalam Tugas Akhir ini, rancang bangun lengan robot 3 DOF berbasis ATmega328 untuk memindahkan biskuit dari kedua *conveyor* ke kaleng biskuit berdasarkan rasa pada biskuit.

Kami mengucapkan terimakasih kepada semua pihak yang telah mendukung dalam pembuatan buku tugas akhir ini, termasuk dosen pembimbing kami, yaitu Ir. Hanny Boedinugroho, MT. yang telah menyempatkan waktunya untuk membimbing kami dalam menyelesaikan tugas akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 02 Juni 2016

Penulis

DAFTAR ISI

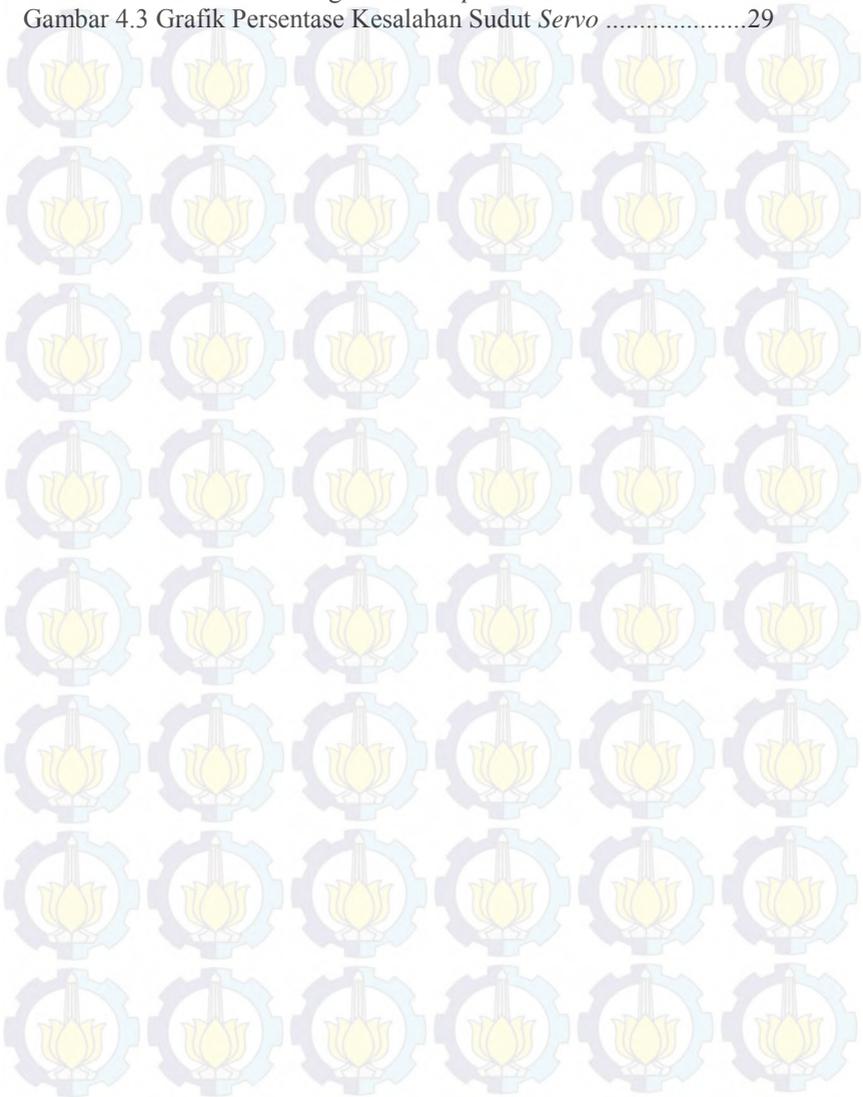
	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah.....	1
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan.....	2
1.7 Relevansi.....	3
BAB II TEORI DASAR.....	5
2.1 Motor <i>Servo</i>	5
2.2 Arduino Uno.....	6
2.2.1 Pemrograman Arduino IDE.....	7
2.2.2 Sinyal PWM pada Arduino Uno.....	7
2.3 LED Merah dan Fotodiode sebagai Sensor Pendeteksi Biskuit pada Conveyor.....	8
2.4 <i>3 Degrees of Freedom</i>	9
BAB III PERANCANGAN DAN PEMBUATAN ALAT.....	11
3.1 Pembuatan Perangkat Elektronik.....	12
3.1.1 Sistem Minimum ATmega328 Sebagai Arduino Uno.....	12
3.1.2 Pengkabelan Sistem Minimum ATmega328 dengan Motor <i>Servo</i>	12

3.1.3	Pengkabelan Sistem Minimum ATmega328 dengan Fotodiode dan LED Sebagai Sensor Pendeteksi Benda ...	13
3.1.4	Rangkaian Sensor Pendeteksi Benda dengan Transistor NPN TIP122 Sebagai Sakelar dan Motor 12 Volt Searah	13
3.2	Pembuatan Perangkat Mekanik.....	14
3.2.1	Pembuatan Lengan Robot.....	14
3.2.2	Pembuatan <i>Conveyor</i>	17
3.3	Pembuatan Perangkat Lunak.....	19
3.3.1	Pembuatan <i>Flowchart</i> Program.....	19
3.3.2	Pendefinisian Pin dan Pustaka	21
3.3.3	Segmen Program untuk Pengaturan Awal	21
3.3.4	Segmen Program untuk Menjalankan <i>Servo</i>	22
3.3.5	Segmen Program <i>Void Loop</i>	23
3.3.6	Segmen Program <i>Digital Input</i> dan <i>Output</i> , Kondisi, Perbandingan, dan Penjumlahan	23
BAB IV PENGUKURAN DAN PENGUJIAN		25
4.1	Pengukuran dan Pengujian Sensor Pendeteksi Biskuit dengan Fotodiode, LED Merah dan <i>Conveyor</i>	25
4.2	Pengukuran dan Pengujian Sistem Minimum ATmega328 dengan Motor <i>Servo</i>	27
4.3	Pengujian Keseluruhan Alat	29
BAB V PENUTUP		31
DAFTAR PUSTAKA		33
LAMPIRAN A		A-1
A.1	<i>Listing</i> Program.....	A-1
LAMPIRAN B.....		B-1
B.1.	Mg946r.....	B-1
B.2.	Sg90.....	B-2
B.3.	ATmega328	B-3
DAFTAR RIWAYAT HIDUP		C-1

DAFTAR GAMBAR

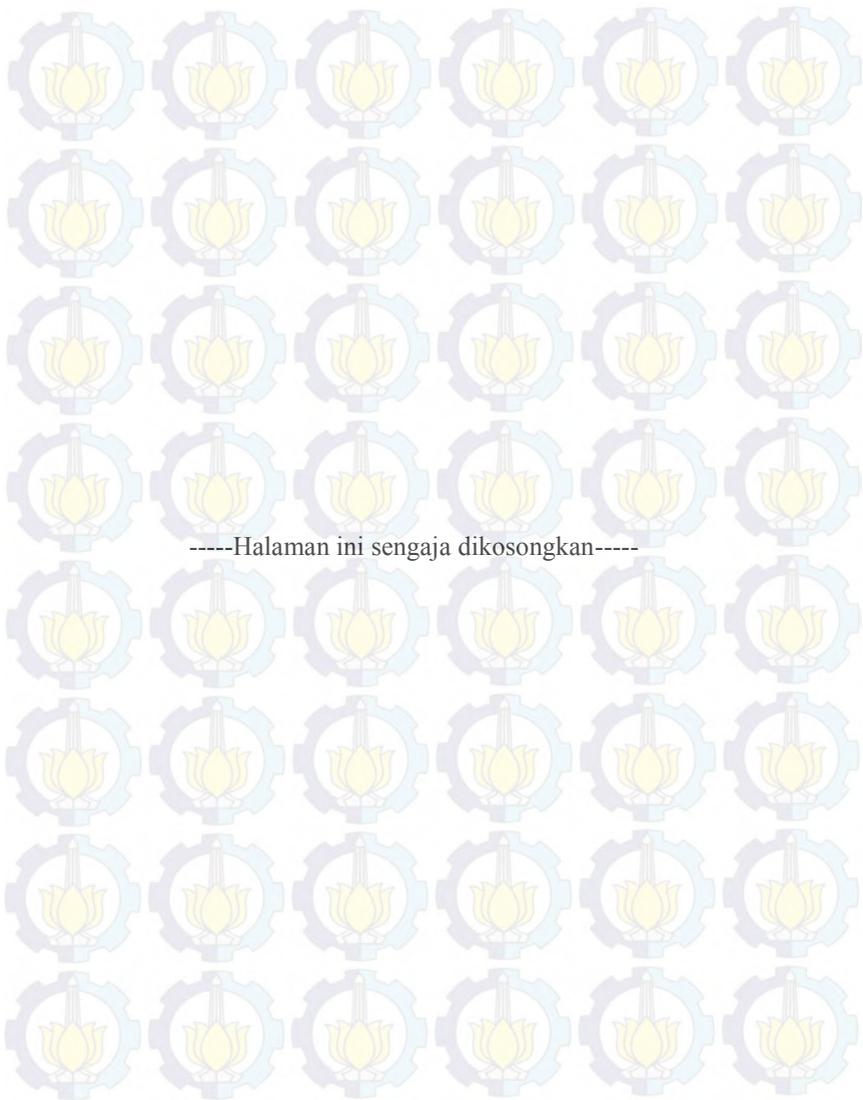
	HALAMAN
Gambar 2.1 Bentuk Fisik <i>Servo</i> Mikro	5
Gambar 2.2 Pulsa Kendali Motor <i>Servo</i>	6
Gambar 2.3 Bentuk Fisik Arduino Uno	6
Gambar 2.4 Contoh Arduino IDE 1.5.3	7
Gambar 2.5 Sinyal PWM dengan Nilai yang Diberikan	8
Gambar 2.6 Konstruksi Fotodiode dan LED sebagai Sensor Pendeteksi Biskuit	9
Gambar 2.7 Ilustrasi Lengan Robot 3 Sendi	9
Gambar 3.1 Blok Fungsional Sistem Secara Keseluruhan	11
Gambar 3.2 Pengkabelan Sistem Minimum ATmega328 sebagai Arduino Uno.....	12
Gambar 3.3 Pengkabelan Sistem Minimum ATmega328 dengan Motor <i>Servo</i> (dari kiri ke kanan: motor <i>servo</i> penjepit, ujung, lengan atas, lengan bawah, dan basis).....	13
Gambar 3.4 Pengkabelan Sistem Minimum ATmega328 dengan Sensor Pendeteksi Benda.....	13
Gambar 3.5 Rangkaian Transistor NPN TIP122 dan Motor 12 volt Searah.....	14
Gambar 3.6 Rancangan Bagian Dasar Lengan Robot.....	14
Gambar 3.7 Bagian Lengan Bawah Robot.....	15
Gambar 3.8 Bagian Lengan Atas Robot.....	15
Gambar 3.9 Bagian Peralatan Lengan Robot.....	16
Gambar 3.10 Bagian Tambahan Lengan Robot	16
Gambar 3.11 Hasil Rancangan Keseluruhan Bagian	17
Gambar 3.12 Bagian Bingkai <i>Conveyor</i>	18
Gambar 3.13 Bagian <i>Driver Pulley</i> dan <i>Tail Pulley Conveyor</i>	18
Gambar 3.14 <i>Flowchart</i> Keseluruhan Program	20
Gambar 3.15 Program Definisi Pustaka dan Tipe Data	21
Gambar 3.16 Segmen Program untuk Pengaturan Awal.....	21
Gambar 3.17 Segmen Program untuk Menjalankan Motor <i>Servo</i>	22
Gambar 3.18 Segmen Program untuk Melakukan Pengulangan.....	23
Gambar 3.19 Segmen Program Pengulangan dengan Kondisi dan Syarat Matematis	23

Gambar 4.1 Gambar Keseluruhan Alat.....25
Gambar 4.2 Grafik Data Pengukuran *Output Servo*.....28
Gambar 4.3 Grafik Persentase Kesalahan Sudut *Servo*29



DAFTAR TABEL

	HALAMAN
Tabel 4.1 Nilai ADC Sensor 1	26
Tabel 4.2 Nilai ADC Sensor 2	26
Tabel 4.3 Data Pengukuran <i>Output Servo</i>	27
Tabel 4.4 Persentase Kesalahan Sudut <i>Servo</i>	28
Tabel 4.5 Pengujian Jumlah Biskuit.....	30
Tabel 4.6 Hasil Pengujian ke-1	30
Tabel 4.7 Hasil Pengujian ke-2	30



BAB I

PENDAHULUAN

1.1 Latar Belakang

Kapasitas produksi industri biskuit di Indonesia terus menunjukkan peningkatan setiap tahunnya, yaitu mencapai 296.074 ton per tahun pada 2005, kemudian meningkat menjadi 299.035 ton per tahun pada 2006. Sampai dengan 2008 kapasitas terus bertambah hingga mencapai 314.106 ton per tahun. Sedangkan pada 2009 ini kapasitas produksi diperkirakan akan meningkat lagi menjadi 326.670 ton per tahun. Peningkatan kapasitas produksi yang terus bertambah setiap tahun tidak terlepas dari begitu banyaknya produsen dalam industri ini sehingga peningkatan terjadi karena adanya pengembangan kapasitas produksi dari produsen lama maupun beroperasinya pabrik-pabrik baru dalam industri biskuit dalam berbagai skala usaha.^[1]

Proses pengemasan biskuit dalam jumlah yang sangat besar juga akan membutuhkan jumlah pegawai yang banyak. Hal ini akan membuka peluang terbukanya lapangan pekerjaan. Namun, di sisi lain dengan jumlah pegawai yang sama akan dapat meningkatkan jumlah produksi jika memanfaatkan teknologi sehingga dapat memanfaatkan tenaga manusia untuk proses yang benar-benar tidak dapat dilakukan oleh mesin dan memperlakukan manusia selayaknya dalam proses pengemasan.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan adalah menentukan pengendalian sudut 3 motor *servo* yang dapat bergerak secara bersamaan dalam satu eksekusi perintah untuk menentukan posisi akhir berdasarkan sudut motor *servo*.

1.3 Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah berat dan bentuk biskuit yang dapat dipindahkan oleh lengan robot, lokasi garis antrian biskuit dan kaleng yang menjadi kemasan biskuit. Dalam hal ini bentuk biskuit yang dimaksud adalah *Cookies Good Time*.

¹ Bersumber dari website <http://www.datacon.co.id/MieInstan-2009Biskuit.html> (di akses pada tanggal 21 Oktober 2015)

1.4 Tujuan

Tujuan menuliskan tugas akhir ini adalah proses pembuatan lengan robot dapat membantu proses pemindahan biskuit dari garis antrian ke mika kemasan dengan menerapkan peralatan yang mudah dikendalikan.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap persiapan akan dipelajari mengenai konsep pengendalian motor *servo* arus searah dengan bahasa pemrograman Arduino. Pada tahap perencanaan dan pembuatan alat, akan dilakukan pembuatan kerangka lengan robot beserta rangkaian elektronik yang menjadi masukan dan keluaran dari sistem, setelah pembuatan kerangka lengan robot, maka proses selanjutnya adalah pembuatan jalur mekanik agar semua masukan dan keluaran saling terhubung. Kemudian akan dilakukan pembuatan program untuk mengendalikan sudut motor *servo* dari ATmega328 dengan bahasa pemrograman Arduino. Kemudian pembuatan program untuk sensor pendeteksi adanya biskuit dalam garis antrian. Setelah dilakukan perencanaan dan pembuatan alat, pengujian yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Arduino, Motor *Servo*, dan 3 DOF.

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas tentang penjelasan dari metodologi yang digunakan dan implementasinya pada lengan robot yang digunakan untuk memindahkan biskuit

Bab IV Pengukuran dan Pengujian

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian aktivitas motor pada lengan robot, *power supply* dan sensor pendeteksi biskuit. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan memberikan manfaat berupa penerapannya pada dunia industri khususnya industri biskuit untuk proses pengemasan guna meningkatkan jumlah produksi dengan jumlah tenaga kerja yang sama dan diharapkan pengembangannya dalam dunia pendidikan untuk menutupi kekurangan agar alat dapat berfungsi optimal.

BAB II TEORI DASAR

Bab ini akan dibahas mengenai teori – teori yang berkaitan dengan peralatan yang akan dirancang. Teori yang mendukung penyelesaian tugas akhir ini diantaranya adalah mengenai; motor *servo*, Arduino Uno, LED merah, fotodiode, *Power Supply* 12 Volt searah dan *3 Degrees Of Freedom*.

2.1 Motor Servo

Motor *servo* adalah sebuah motor DC kecil yang diberi roda gigi dan potensiometer sehingga dapat menempatkan tanduk *servo* pada posisi yang dikehendaki. Motor *servo* ini jelas menggunakan sistem *close loop* untuk motor *servo* itu sendiri sehingga posisi tanduk yang dikehendaki dapat dipertahankan.[²]

Secara putaran terdapat dua jenis motor *servo*, yaitu motor *servo* standar dan motor *servo continuous*. Motor *servo* standar sering dipakai pada sistem robotika, misalnya untuk membuat Robot Arm (Robot Lengan) sedangkan motor *servo continuous* sering dipakai untuk *Mobile Robot*. Pada bagian *servo* tertulis tipe *servo* yang bersangkutan. Contoh motor *servo* bisa dilihat pada Gambar 2.1

Penggunaan motor *servo* di bidang robotika tentu ada alasannya. Pertama adalah motor *servo* memiliki putaran yang lambat dengan torsi yang kuat (berkat adanya sistem roda gigi). Hal ini cocok dengan bidang robotika.



Gambar 2.1 Bentuk Fisik Servo Mikro

² Bersumber dari Annisa Nadia. 2015

Operasional motor *servo* dikendalikan oleh sebuah sinyal berperiode ± 20 ms, dimana lebar pulsa antara 1 ms dan 2 ms menyatakan akhir dari jangkauan sudut maksimum. Jika motor *servo* diberikan pulsa dengan lebar 1,5 ms maka posisi yang dihasilkan adalah 90° . Jika diberikan pulsa kurang dari 1,5 ms maka posisi mendekati 0° . Jika diberikan pulsa lebih dari 1,5 ms maka posisi mendekati 180° .^[3] Bentuk dari pulsa kendali motor *servo* dapat dilihat pada Gambar 2.2



Gambar 2.2 Pulsa Kendali Motor *Servo*

2.2 Arduino Uno

Arduino merupakan rangkaian mikrokontroler dalam satu papan yang bersifat *open source*. Arduino adalah kombinasi dari perangkat keras, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih.



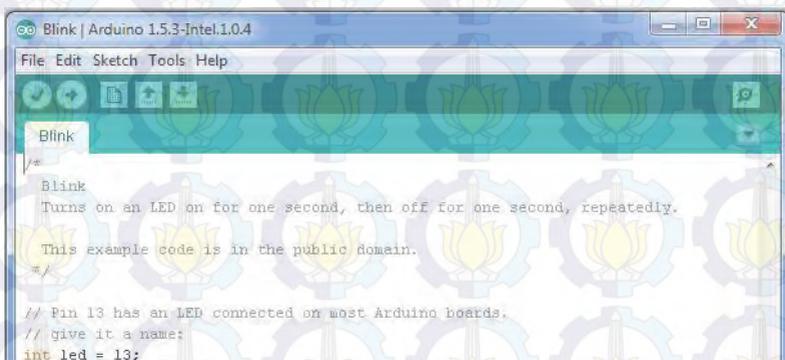
Gambar 2.3 Bentuk Fisik Arduino Uno

³ Bersumber dari <http://elektronika-dasar.web.id/motor-servo/> (di akses pada tanggal 21 Oktober 2015)

Arduino Uno seperti pada Gambar 2.3 merupakan papan mikrokontroler berbasis ATmega328 yang mempunyai 14 *pin* masukan dan keluaran digital (termasuk 6 diantaranya dapat berfungsi sebagai keluaran *Pulse Width Modulation*), 6 masukan *analog*, sebuah kristal 16 MHz, dan sebuah tombol *reset*. Arduino Uno beroperasi pada tegangan 5 volt.

2.2.1 Pemrograman Arduino IDE

IDE adalah sebuah aplikasi untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori mikrokontroler. Ada banyak proyek dan alat-alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino. Selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk dapat terhubung dengan Arduino. Pada Gambar 2.4 merupakan contoh dari IDE 1.5.3



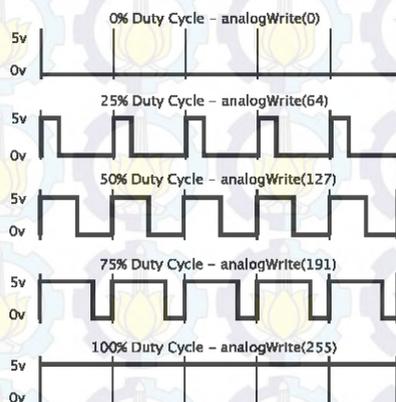
Gambar 2.4 Contoh Arduino IDE 1.5.3

2.2.2 Sinyal PWM pada Arduino Uno

PWM pada Arduino adalah metode mengatur suatu keluaran sinyal *digital* dengan frekuensi tertentu. Pada Arduino Uno frekuensi yang bekerja maksimal adalah 500 Hz. PWM pada Arduino Uno dituliskan dalam bentuk desimal dengan jangkauan angka biner 8 bit. Dengan demikian, jika dituliskan dalam desimal, maka bernilai 0 untuk minimum atau 255 pada keluaran maksimum. Selanjutnya, jika suatu pin keluaran PWM diberi nilai 0, maka pin tersebut akan memberikan keluaran 5 volt sebanyak 0 ketukan dalam 1 detik. Jika suatu pin PWM diberi nilai 255, maka pin tersebut akan memberikan keluaran 5 volt sebanyak 500 ketukan dalam 1 detik. Untuk

mengatur pin PWM agar memberikan keluaran 5 volt sebanyak 250 ketukan tiap 1 detik maka pin PWM diberi nilai 255 dibagi 2 yaitu 127 atau 128. Hal ini dikarenakan sistem biner tidak mengenal koma, misal 127,5 sehingga dibuat nilai pendekatan yaitu 127 atau 128. Salah satu contoh penggunaan sinyal PWM adalah untuk memberikan sinyal aksi kepada motor *servo*. Gambar 2.5 mengilustrasikan sinyal keluaran yang dibangkitkan oleh suatu pin PWM.

Pada Arduino IDE, ATmega328 dapat membangkitkan sinyal PWM dengan menuliskan *analogWrite* (bilangan PWM) dengan terlebih dahulu mendefinisikan pin yang dimaksud. Pin-pin PWM tersebut diantaranya adalah pin 5, 11, 12, 15, 16, dan 17 ATmega328 atau pin 3, 5, 6, 9, 10, dan 11 pada Arduino IDE.



Gambar 2.5 Sinyal PWM dengan Nilai yang Diberikan

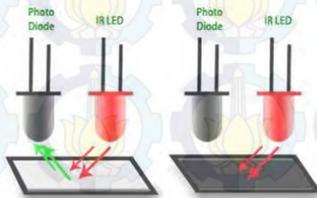
2.3 LED Merah dan Fotodiode sebagai Sensor Pendeteksi Biskuit pada Conveyor

LED (light-emitting diode) adalah suatu semikonduktor yang memancarkan cahaya monokromatik. Warna yang dihasilkan bergantung pada bahan semikonduktor yang dipakai.^[4] Fotodiode adalah jenis dioda yang berfungsi mendeteksi cahaya. Berbeda dengan diode biasa, komponen elektronika ini akan mengubah cahaya menjadi besaran listrik.^[5]

⁴ Bersumber dari https://id.wikipedia.org/wiki/Diode_pancaran_cahaya (di akses pada tanggal 21 Oktober 2015)

⁵ Bersumber dari <https://id.wikipedia.org/wiki/Fotodiode> (di akses pada tanggal 21 Oktober 2015)

Konsep kerja tersebut diaplikasikan pada *conveyor*. Pada saat tidak terdapat biskuit (terang), maka fotodiode mengirimkan logika 1. Pada saat terdapat biskuit (gelap), fotodiode mengirimkan logika 0. Gambar 2.6 merupakan konstruksi fotodiode dan LED sebagai sensor pendeteksi biskuit.

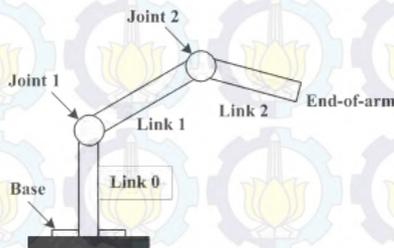


Gambar 2.6 Konstruksi Fotodiode dan LED sebagai Sensor Pendeteksi Biskuit

2.4 3 Degrees of Freedom

Jumlah dari sendi suatu manipulator akan menentukan jumlah *degrees of freedom*. *Degrees of freedom* adalah jumlah kebebasan sudut dari sebuah lengan robot yang dapat bergerak. Manipulator robot dikonstruksi dengan serangkaian sendi dan batang hubung.

Sendi lengan robot mirip dengan sendi dalam tubuh manusia. Namun, setiap sendi lengan robot hanya memiliki 1 derajat kebebasan sehingga hampir semua kasus, jumlah sendi yang dimiliki lengan robot sama dengan jumlah derajat kebebasannya. Setiap sendi dihubungkan dengan dua batang hubung, yaitu batang hubung masukan dan keluaran. Gambar 2.7 merupakan Ilustrasi lengan robot 3 sendi



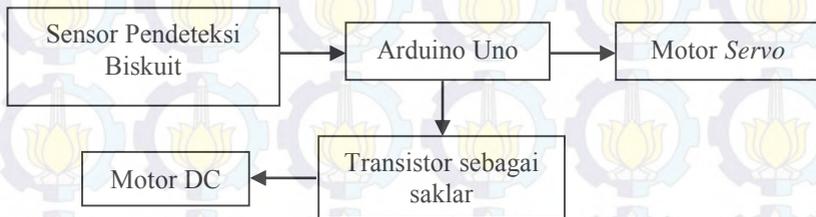
Gambar 2.7 Ilustrasi Lengan Robot 3 Sendi

Pada ujung pergelangan lengan robot terdapat peralatan terkait dengan tugas yang harus dilakukan oleh lengan robot yang disebut sebagai ujung lengan. Ujung lengan dapat berupa penjepit atau peralatan untuk melakukan proses.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Bab ini berisi tahapan yang dilakukan dalam perencanaan dan pembuatan tugas akhir. Penjelasan diawali dengan penjelasan blok fungsional sistem secara keseluruhan, kemudian perancangan perangkat keras dan diakhiri dengan perangkat lunak.

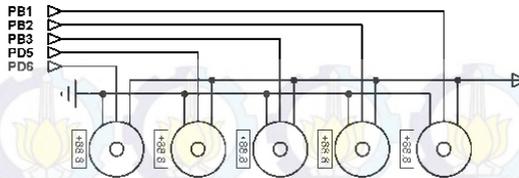


Gambar 3.1 Blok Fungsional Sistem Secara Keseluruhan

Untuk mendapatkan lengan robot yang berjalan sesuai harapan diperlukan bagian pokok, yaitu:

1. *Motor servo*, berfungsi sebagai aktuator lengan robot dengan mendapat sinyal PWM dari pengendali.
2. Sistem minimum *ATMega328*, mikrokontroler yang berfungsi sebagai pengendali, pengolah sinyal masuk dan keluar. Pin yang digunakan adalah pin *digital input*, *PWM*, *analog digital converter* sebagai *digital input*.
3. Sensor pendeteksi benda, sensor ini terdiri dari LED merah yang memancarkan gelombang cahaya merah dan fotodiode yang berfungsi sebagai penerima gelombang cahaya merah.
4. *Conveyor*, berfungsi untuk menyalurkan biskuit menuju ke lengan robot.

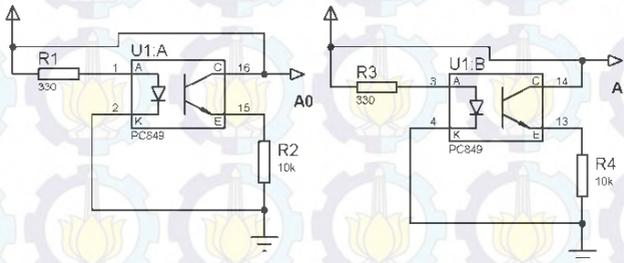
Adapun penjelasan dari blok fungsional pada Gambar 3.1 dan juga tahapan untuk menggerakkan lengan robot oleh motor *servo* hingga mencapai derajat tertentu. Penjelasan dari masing-masing bagian dijelaskan pada sub bab pembuatan perangkat elektronik, pembuatan perangkat mekanik, dan pembuatan perangkat lunak.



Gambar 3.3 Pengkabelan Sistem Minimum ATmega328 dengan Motor *Servo* (dari kiri ke kanan: motor *servo* penjepit, ujung, lengan atas, lengan bawah, dan basis).

3.1.3 Pengkabelan Sistem Minimum ATmega328 dengan Fotodiode dan LED Sebagai Sensor Pendeteksi Benda

Rangkaian sensor pendeteksi benda seperti pada Gambar 3.4 ini terdiri dari resistor 330 Ω , 10k Ω , fotodiode, dan LED merah. LED dan fotodiode dipasang berseberangan dengan jarak yang cukup. Jarak tersebut berfungsi agar biskuit dapat terjatuh di antara LED dan fotodiode. Jika fotodiode dan LED dihalangi oleh biskuit maka rangkaian akan mengirimkan logika 0 pada ATmega328. Jika fotodiode dan LED tidak terhalangi oleh biskuit, maka logika 1 akan dikirimkan ke ATmega328.

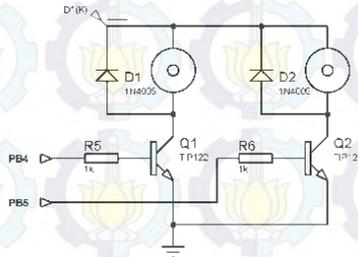


Gambar 3.4 Pengkabelan Sistem Minimum ATmega328 dengan Sensor Pendeteksi Benda

3.1.4 Rangkaian Sensor Pendeteksi Benda dengan Transistor NPN TIP122 Sebagai Sakelar dan Motor 12 Volt Searah

Rangkaian transistor sebagai sakelar pada Gambar 3.5 ini berfungsi untuk mengaktifkan motor DC 12 volt sehingga *conveyor* dapat bergerak. Rangkaian ini terdiri dari resistor 1k Ω , transistor TIP122, dan dioda 1N4004. Jika rangkaian ini diberi logika 1 oleh ATmega328, maka motor aktif sehingga *conveyor* bergerak. Jika rangkaian ini diberi logika 0 oleh ATmega328, maka motor aktif sehingga *conveyor* berhenti. Selain terhubung

dengan ATmega328, rangkaian ini juga terhubung dengan sumber tegangan 12 volt untuk mencatu motor DC.



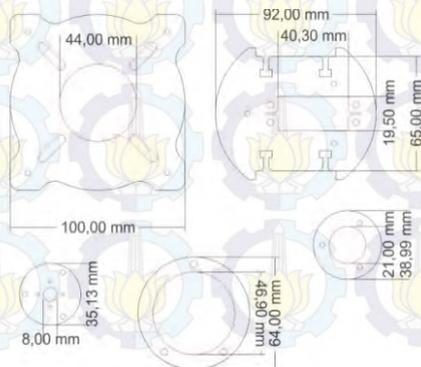
Gambar 3.5 Rangkaian Transistor NPN TIP122 dan Motor 12 volt Searah

3.2 Pembuatan Perangkat Mekanik

Dalam perangkat mekanik, terdapat beberapa bagian yang harus dibuat untuk dapat bekerja dengan baik. Terdapat 2 bagian besar dalam perangkat mekaniknya yang terdiri dari 1 buah lengan robot dan 2 buah *conveyor*. Satu lengan robot ini terdiri dari beberapa sub bagian yaitu plat dasar, dasar, lengan bawah, lengan atas, ujung, penjepit, dan tambahan. Berikut tahapan-tahapan pembuatan perangkat mekaniknya:

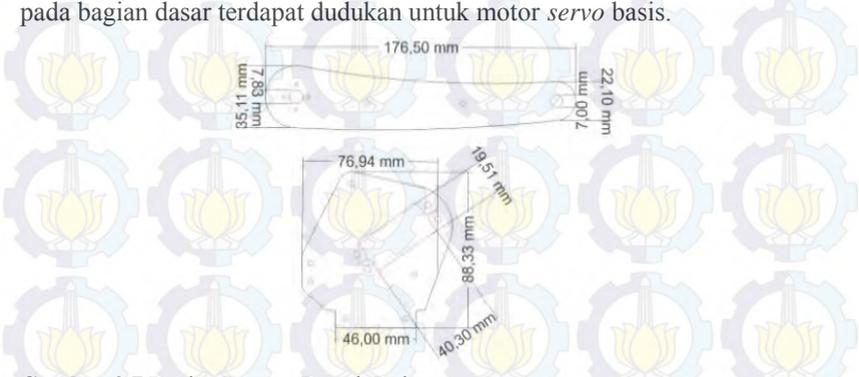
3.2.1 Pembuatan Lengan Robot

Lengan robot terdiri dari 5 bagian yaitu bagian dasar, lengan bawah, lengan atas, tambahan, dan peralatan berupa penjepit. Semua bahan yang digunakan untuk membuat lengan robot ini terbuat dari bahan akrilik. Sedangkan untuk penghubung digunakan mur dan baut berdiameter 3 mm.



Gambar 3.6 Rancangan Bagian Dasar Lengan Robot

Gambar 3.6 adalah rancangan bagian dasar lengan robot yang berfungsi sebagai penghubung antara lengan bawah dengan tanah. Selain itu, pada bagian dasar terdapat dudukan untuk motor *servo* basis.



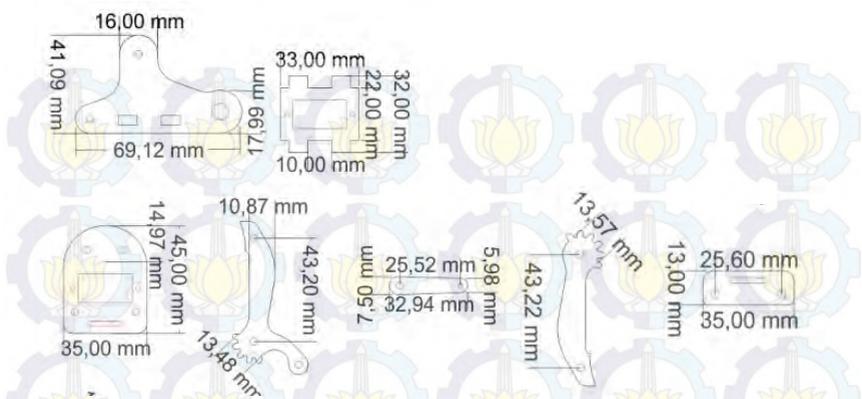
Gambar 3.7 Bagian Lengan Bawah Robot

Gambar 3.7 ini secara langsung terhubung dengan motor *servo*. Lengan bawah ini berfungsi sebagai *link* atau penghubung antara basis dengan lengan atas robot.



Gambar 3.8 Bagian Lengan Atas Robot

Gambar 3.8 adalah bagian lengan atas robot yang secara tidak langsung terhubung oleh motor *servo*, tetapi dihubungkan dengan sebuah lempeng akrilik yang berfungsi sebagai otot. Otot ini berfungsi sebagai penghubung antara motor *servo* dengan lengan atas sehingga dapat menarik dan mendorong lengan atas robot.



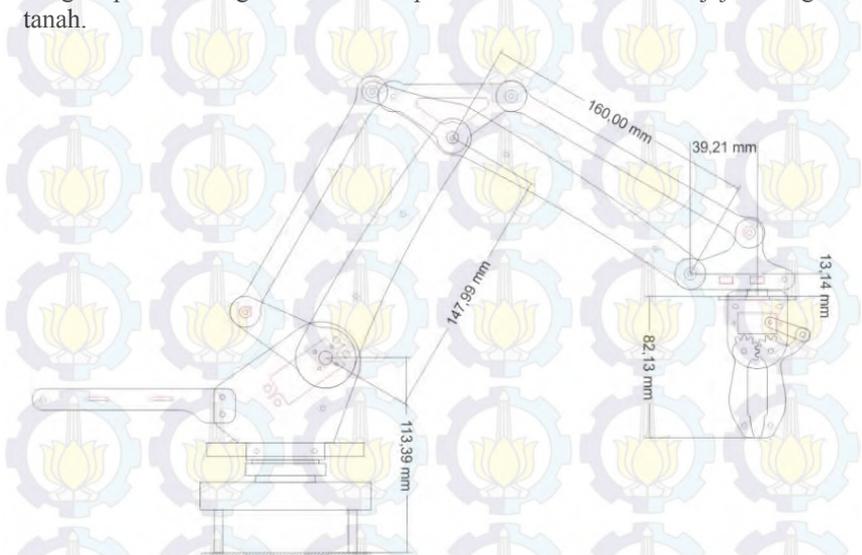
Gambar 3.9 Bagian Peralatan Lengan Robot

Bagian selanjutnya yaitu Gambar 3.9, berupa ujung lengan robot yang berfungsi sebagaiudukan peralatan lengan robot. Dalam hal ini yang dimaksud dengan peralatan adalah penjepit yang digerakkan oleh sebuah motor *mikroservo*. Berbeda dengan gerakan *servo-servo* sebelumnya. *Mikroservo* yang terletak pada bagian peralatan ini hanya membuka dan menutup *gripper*, sedangkan *servo* yang lain bergerak secara rotasi antara sudut 0° - 180° .



Gambar 3.10 Bagian Tambahan Lengan Robot

Gambar 3.10 ini berfungsi sebagaiudukan sistem minimum ATmega328 dan otot yang berfungsi untuk menghubungkan bagian dasar dengan peralatan agar kedudukan peralatan selalu dalam sejajar dengan tanah.

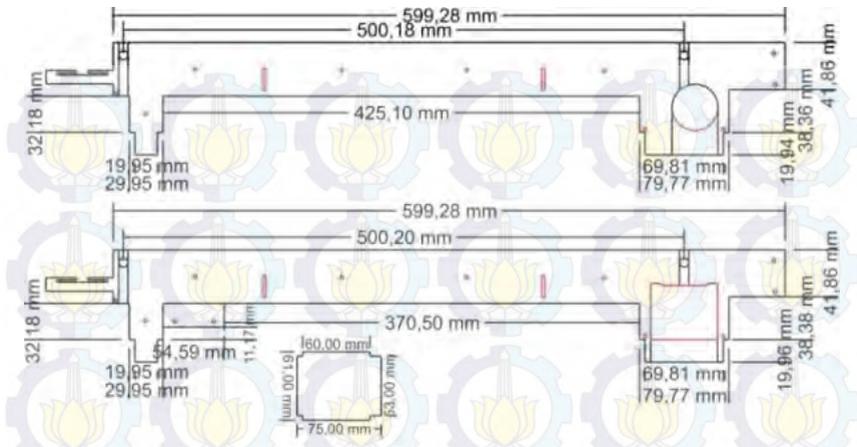


Gambar 3.11 Hasil Rancangan Keseluruhan Bagian

Gambar 3.11 merupakan Hasil rancangan keseluruhan bagian yang dirakit menjadi satu, dimana diantaranya disusun dari beberapa bagian, yaitu bagian dasar lengan robot, bagian lengan bawah robot, bagian lengan atas robot, bagian peralatan lengan robot, dan bagian tambahan lengan robot.

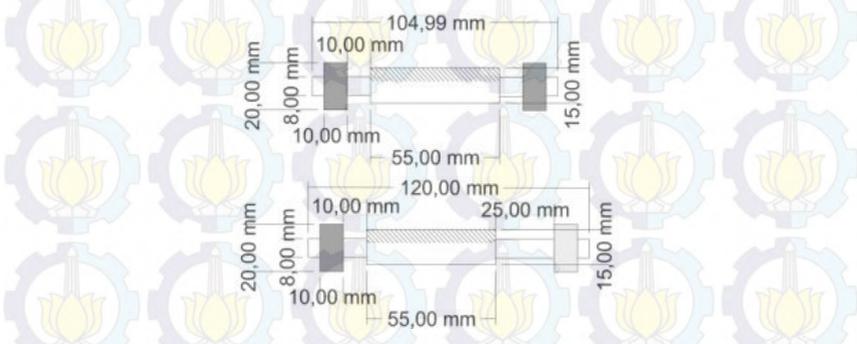
3.2.2 Pembuatan Conveyor

Conveyor dalam sistem ini berfungsi untuk menyalurkan biskuit dari bagian produksi ke lengan robot. Conveyor pada Gambar 3.12 dibuat dari bahan akrilik dengan ketebalan 2 mm. Di antara 2 lempeng akrilik terdapat belt atau sabuk yang berfungsi untuk menghubungkan driver pulley dan tail pulley.



Gambar 3.12 Bagian Bingkai Conveyor

Driver pulley seperti pada Gambar 3.13 adalah poros yang terdapat roda gigi di salah satu ujungnya dan terhubung dengan motor sebagai penggerak. Sedangkan *tail pulley* adalah poros lain pada conveyor yang tidak terhubung dengan motor penggerak. Saat *driver pulley* digerakkan oleh motor maka sabuk akan bergerak dan *tail pulley* akan berputar.



Gambar 3.13 Bagian Driver Pulley dan Tail Pulley Conveyor

Secara umum, fungsi dari conveyor adalah mengubah gerak rotasi motor menjadi gerak linier pada sabuk conveyor sehingga barang dapat disalurkan dari ujung satu dengan ujung yang lain. Dalam kasus ini, yang disalurkan adalah biskuit dari bagian produksi ke lengan robot.

3.3 Pembuatan Perangkat Lunak

Dalam perangkat lunak, terdapat beberapa program yang harus dibuat untuk dapat menggerakkan motor *servo* dengan baik. Tahapan pembuatan tersebut adalah sebagai berikut:

3.3.1 Pembuatan *Flowchart* Program

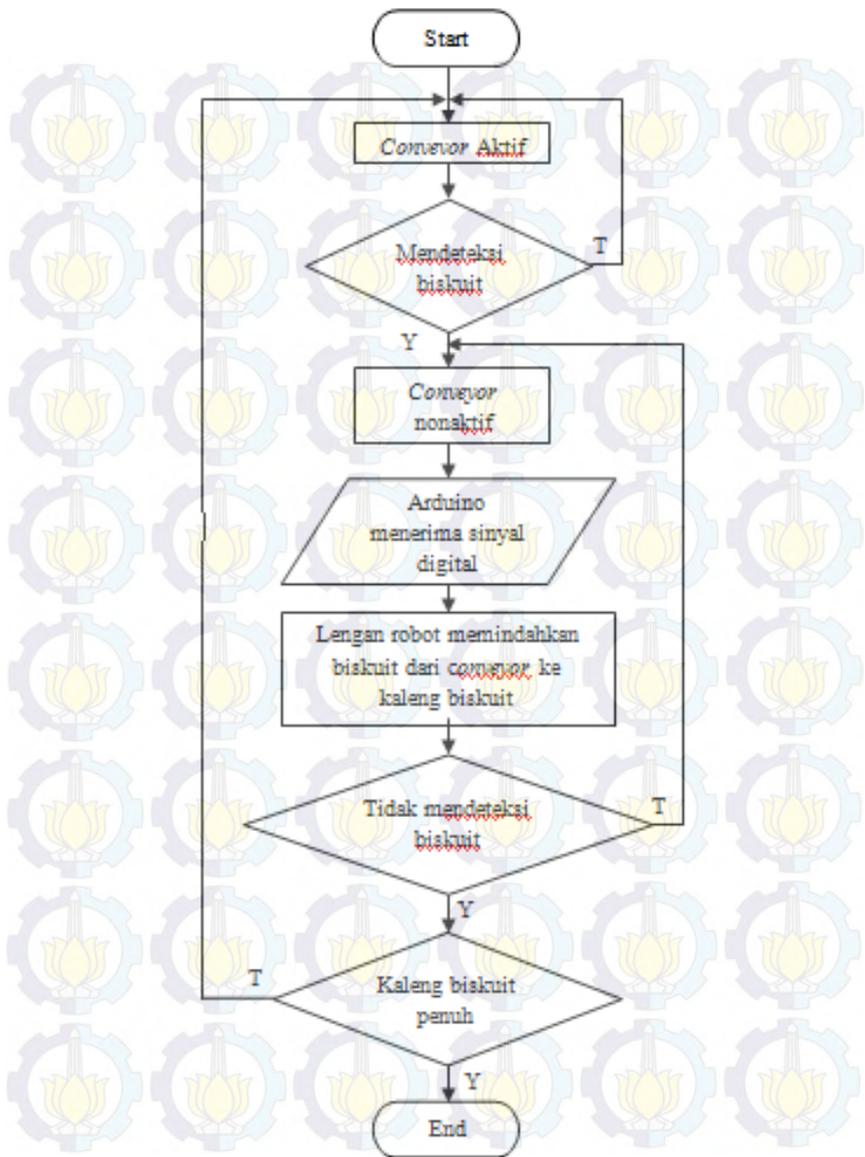
Flowchart merupakan bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Bagan ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung. *Flowchart* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan poses kegiatan menjadi lebih jelas.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- *START*: berisi instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- *READ*: berisi instruksi untuk membaca data dari suatu peralatan.
- *PROCESS*: berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- *WRITE*: berisi instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- *END*: mengakhiri kegiatan pengolahan

Flowchart program dari Tugas Akhir ini meliputi seluruh sistem jalannya alat ini. Sistem yang dimaksud adalah sistem umum secara keseluruhan.

Pada Gambar 3.14 dijelaskan bahwa pada saat program dijalankan, *conveyor* aktif. *Conveyor* akan memindahkan biskuit. Jika pendeteksi biskuit tidak mendeteksi biskuit, maka *conveyor* akan kembali aktif. Jika sensor pendeteksi biskuit mendeteksi biskuit, maka Arduino menerima sinyal digital berupa logika 0. Setelah mendapat logika 0, *conveyor* nonaktif dan lengan robot diberi sinyal oleh Arduino berupa sinyal PWM untuk mengaktifkan motor *servo*. Setelah lengan robot memindahkan biskuit, sensor pendeteksi biskuit akan mengirimkan logika 1 ke Arduino dan *conveyor* kembali aktif. Jika sensor pendeteksi biskuit tetap mengirimkan logika 0, maka lengan biskuit akan tetap memindahkan biskuit. Program akan berulang sehingga lengan robot memindahkan biskuit coklat muda sebanyak 14 kali dan coklat tua sebanyak 15 kali. Kemudian program berhenti.



Gambar 3.14 Flowchart Keseluruhan Program

3.3.2 Pendefinisian Pin dan Pustaka

```
#include <Servo.h>
Servo servo1;
int a;
int pinOutput = 13;
int pinInput = A1;
```

Gambar 3.15 Program Definisi Pustaka dan Tipe Data

Pada Gambar 3.15 dapat dijelaskan bahwa dalam rancangan program yang dibuat, dimasukkan pustaka untuk kendali motor *servo* yang telah disediakan oleh Arduino IDE. Setelah memasukan pustaka *servo*, diberi nama setiap *servo* yang akan digunakan dalam pemrograman, dalam hal ini *servo* diberi nama *servo1*.

Kemudian, variabel *a* berupa tipe data *integer* yang dapat memuat 16-bit dimulai dari -32,768 hingga 32,767. Variabel ini berfungsi untuk menyimpan jumlah biskuit yang telah dimasukan dalam salah satu lubang berlabel A.

Definisi *pinOutput* pada segmen program tersebut untuk mendefinisikan sebuah variabel bernama *pinOutput* yang berada pada pin 13 dan *pinInput* yang berada pada pin A1 Arduino.

3.3.3 Segmen Program untuk Pengaturan Awal

```
void setup() {
  a=0;
  pinMode(pinOutput, OUTPUT);
  pinMode(pinInput, INPUT);
}
```

Gambar 3.16 Segmen Program untuk Pengaturan Awal

Void pada Arduino berfungsi sebagai kata kunci untuk membuat sub program dari program utama. Jika fungsi *void* digunakan sebagai *void setup*, maka sub program tersebut berfungsi sebagai program awal dari Arduino dan tidak diulang saat program dieksekusi.

Pada Gambar 3.16, terdapat variabel *a* yang dinyatakan sebagai 0. Hal ini berarti bahwa pada awal program, *a* didefinisikan sebagai angka bulat 0. Kemudian fungsi *pinMode* berfungsi untuk merepresentasikan fungsi pin

yang akan dipanggil. Jika pada potongan program Gambar 3.16 tertulis *pinMode* (*pinOutput*, OUTPUT), maka fungsi tersebut menyatakan bahwa pin yang dinamakan sebagai *pinOutput* berfungsi sebagai OUTPUT. Sebagaimana sebelumnya *pinOutput* adalah nama untuk pin 13 dalam potongan program ini.

Pada Gambar 3.16 terdapat fungsi *pinMode* (*pinInput*, INPUT) yang berarti pin yang dinamakan sebagai *pinInput* difungsikan sebagai *input* pada program ini. Sebagaimana sebelumnya, yang dimaksud dengan *pinInput* adalah pin A1 pada Arduino atau pin C1 pada ATmega328.

3.3.4 Segmen Program untuk Menjalankan Servo

```
void menjalankan_servo1() {  
    servo1.attach(9);  
    servo1.write(120);  
}  
  
void menjalankan_servo2() {  
    servo1.attach(9);  
    servo1.write(60);  
}
```

Gambar 3.17 Segmen Program untuk Menjalankan Motor Servo

Pada Gambar 3.17, terdapat fungsi *void* yang diikuti oleh nama subprogram yaitu *menjalankan_servo1* dan *menjalankan_servo2*. Dari masing-masing subprogram terdapat fungsi *attach* dan *write* yang hanya dapat dikenali apabila memasukan pustaka *servo* dalam program.

Maksud dari fungsi *attach* adalah menyatakan bahwa *servo1* dihubungkan dengan pin 9 Arduino. Dengan demikian, pada pin 9 akan menghasilkan gelombang PWM.

Maksud dari fungsi *write* yang dihubungkan dengan nama motor *servo* adalah memanipulasi gelombang PWM agar motor *servo* dapat bergerak sebesar sudut yang dituliskan dalam konstanta dalam kurung. Jika dalam Gambar 3.17 terdapat konstanta 60° dan 120°, maka lebar pulsa sinyal PWM yang dihasilkan ketika menghasilkan sudut 60° adalah 1,375 milidetik dan 1,675 milidetik untuk menghasilkan sudut 120°. Hal ini sesuai dengan kaidah motor *servo* secara umum, yakni menghasilkan 0° apabila diberi PWM dengan lebar pulsa 1 milidetik dan 180° apabila diberi PWM dengan lebar pulsa 2 milidetik.

3.3.5 Segmen Program *Void Loop*

Void loop adalah subprogram yang akan dijalankan berulang-ulang oleh *processor*. *Void loop* dibuat setelah menuliskan *void setup* pada satu program utama.

```
void loop() {  
    menjalankan_servo1();  
    delay(1000);  
    menjalankan_servo2();  
    delay(1000);  
}
```

Gambar 3.18 Segmen Program untuk Melakukan Pengulangan

Seperti pada contoh Gambar 3.18, maka program yang akan diulang adalah subprogram *menjalankan_servo1* kemudian ditunda 1 detik dan dilanjutkan ke subprogram *menjalankan_servo2* ditunda 1 detik demikian seterusnya.

3.3.6 Segmen Program *Digital Input dan Output, Kondisi, Perbandingan, dan Penjumlahan*

```
void loop() {  
    kondisi_input = digitalRead(A1);  
    if ((a << 3) && (kondisi_input == LOW)){  
        digitalWrite(pinOutput, LOW);  
        a=a+1;  
    }  
    else if ((a == 3) && (kondisi_input == LOW)){  
        digitalWrite(pinOutput, LOW);  
    }  
    else {  
        digitalWrite(pinOutput, HIGH);  
    }  
}
```

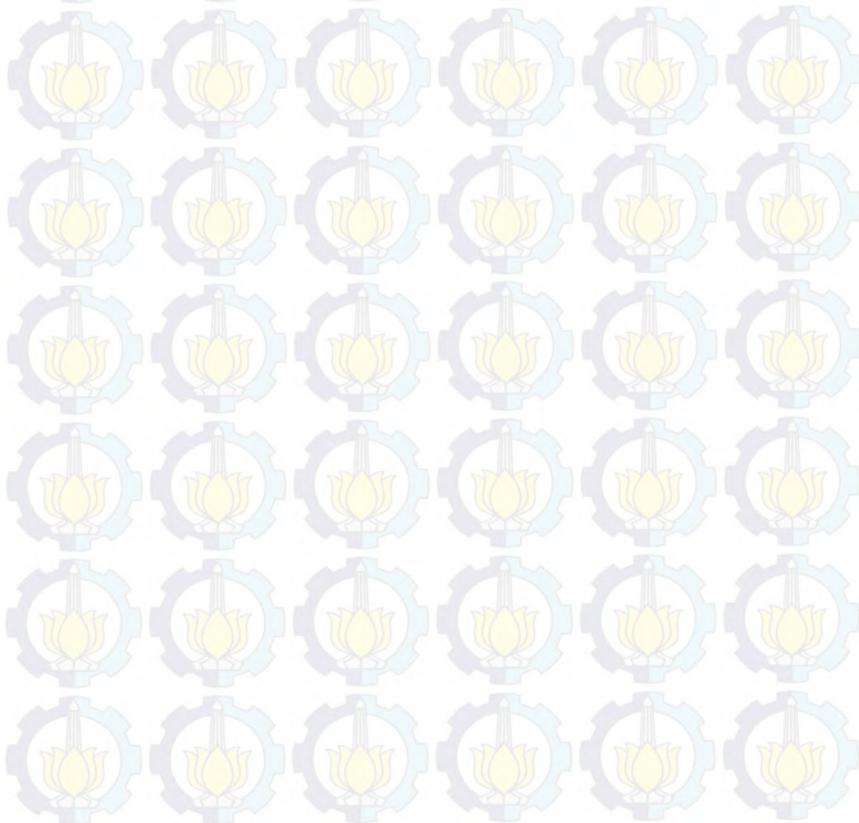
Gambar 3.19 Segmen Program Pengulangan dengan Kondisi dan Syarat Matematis

Pada Gambar 3.19 terdapat subprogram pengulangan yang berisi *digital input* dan *output*. Maksud dari potongan program tersebut adalah variabel *kondisi_input* membaca logika pin A1, 0 atau 1. Setelah terbaca, terdapat syarat yang dipenuhi untuk menjalankan perintah. Pada syarat pertama, jika kondisi *a* kurang dari 3 dan kondisi *_input* adalah logika 0, maka *digitalWrite*

berfungsi untuk menyatakan bahwa pin yang bernama *pinOutput* berlogika 0 dan variabel *a* ditambahkan 1. Selama kondisi *a* kurang dari 3, maka fungsi ini akan dijalankan sedemikian hingga variabel ditambahkan dengan konstanta 1.

Jika variabel *a* sama dengan 3 dan kondisi *_input* berlogika 0, maka yang dijalankan oleh program adalah menyatakan logika 0 pada pin bernama *pinOutput* dan tidak lagi menambahkan variabel *a* dengan konstanta 1.

Fungsi *else* pada program bermaksud sebagai kondisi diluar syarat yang telah dituliskan sebelumnya yaitu kondisi *_input* berlogika 1 sehingga pin bernama *pinOutput* dinyatakan sebagai logika 1.



BAB IV

PENGUKURAN DAN PENGUJIAN

Untuk mengetahui bahwa alat telah bekerja dengan benar maka perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat. Alat yang telah selesai dibuat, dapat dilihat pada Gambar 4.1, alat tersebut secara umum terdapat 2 buah *conveyor*, lengan robot dengan 3 sudut kebebasan, dan biskuit (dalam hal ini, digunakan 1 kaleng *Cookies Good Time*.)



Gambar 4.1 Gambar Keseluruhan Alat

4.1 Pengukuran dan Pengujian Sensor Pendeteksi Biskuit dengan Fotodiode, LED Merah dan Conveyor

Tujuan dari pengukuran dan pengujian sensor pendeteksi benda adalah mengetahui kerja dari fotodiode dan LED merah.

Cara pengambilan data yang digunakan adalah memprogram Arduino dan membacanya dengan komunikasi serial. Hasil dari pengambilan data nilai ADC sensor 1 dapat dilihat pada Tabel 4.1, dan untuk Hasil dari pengambilan data nilai ADC sensor 2 dapat dilihat pada Tabel 4.2.

Dalam program, apabila sensor mendeteksi biskuit maka *conveyor* berhenti. Sedangkan jika tidak ada biskuit, maka *conveyor* aktif.

Tabel 4.1 Nilai ADC Sensor 1

No.	ADC Terdapat Biskuit	Volt Terdapat Biskuit	ADC Tanpa Biskuit	Volt Tanpa Biskuit
1	12	0.0576	991	4.7568
2	17	0.0816	992	4.7616
3	7	0.0336	993	4.7664
4	28	0.1344	995	4.776
5	17	0.0816	996	4.7808
6	10	0.048	994	4.7712
7	11	0.0528	992	4.7616
8	21	0.1008	993	4.7664
9	7	0.0336	995	4.776
10	13	0.0624	993	4.7664
Rata-Rata	14,3	0.06864	9934	4.76832

Tabel 4.2 Nilai ADC Sensor 2

No.	ADC Terdapat Biskuit	Volt Terdapat Biskuit	ADC Tanpa Biskuit	Volt Tanpa Biskuit
1	11	0.0528	1001	4.8048
2	20	0.096	1002	4.8096
3	9	0.0432	1003	4.8144
4	11	0.0528	1000	4.8
5	19	0.0912	1001	4.8048
6	10	0.048	1001	4.8048
7	15	0.072	1000	4.8
8	18	0.0864	1000	4.8
9	6	0.0288	1002	4.8096
10	13	0.0624	1002	4.8096
Rata-Rata	13,2	0.06336	10012	4.80576

Nilai ADC pada Arduino merupakan hasil dari pengolahan nilai biner 10 bit menjadi 1023 nilai desimal dan ADC Arduino hanya dapat membaca tegangan 0-5 VDC. Dari keterangan tersebut dapat diketahui bahwa tiap 1 nilai desimal ADC bernilai 0,0048 volt DC. Sehingga ketika tidak terdapat biskuit, nilai tertinggi dari salah satu sensor adalah

$$\begin{aligned} \text{Nilai Desimal ADC} \times 0,0048 &= \text{tegangan sensor} & (1) \\ 28 \times 0,0048 &= 0,1344 \text{ volt} \end{aligned}$$

Sedangkan nilai terendah pada saat fotodiode terhalang oleh biskuit adalah

$$\begin{aligned} \text{Nilai Desimal ADC} \times 0,0048 &= \text{tegangan sensor} & (2) \\ 991 \times 0,0048 &= 4,7568 \text{ volt} \end{aligned}$$

Dari pengambilan data tersebut, dihasilkan *conveyor* aktif meski tegangan yang dihasilkan dari sensor hanya 4,7568 volt dan *conveyor non* aktif pada saat tegangan sensor 0,1344 volt.

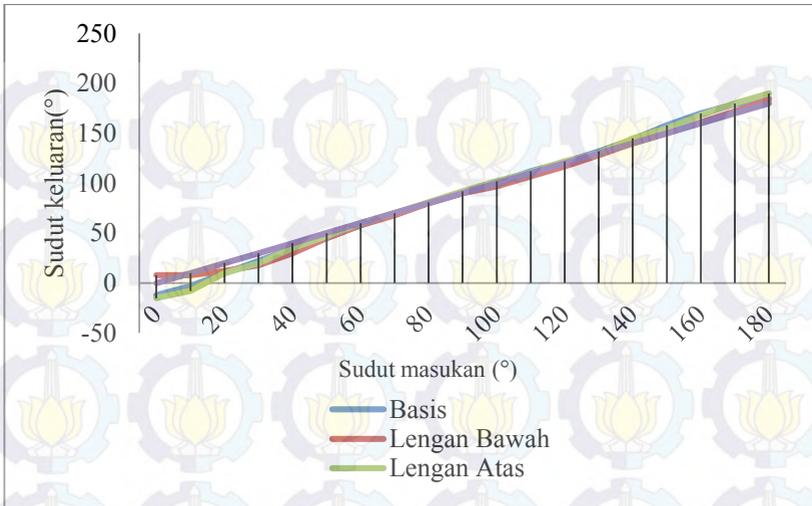
4.2 Pengukuran dan Pengujian Sistem Minimum ATmega328 dengan Motor Servo

Tujuan dari pengambilan data dari sudut motor *servo* adalah untuk mengetahui kesalahan sudut yang dihasilkan dari masing-masing motor *servo*.

Untuk melakukan pengukuran gerakan motor *servo*, maka dilakukan pemrograman melalui Arduino dengan *interval* sudut sebesar 10° tiap motor *servo* hingga 180° dan diukur dengan busur. Hasil dari pengambilan data pengukuran *Output Servo* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Data Pengukuran *Output Servo*

No.	Input (derajat)	Basis (derajat)	Lengan Bawah (derajat)	Lengan Atas (derajat)
1.	0	-12	8	-15
2.	10	-3	8	-8
3.	20	10	12	10
4.	30	22	18	20
5.	40	35	30	35
6.	50	45	45	47
7.	60	58	58	60
8.	70	70	68	70
9.	80	80	80	81
10.	90	90	90	92
11.	100	100	97	102
12.	110	112	107	111
13.	120	121	117	122
14.	130	132	128	130
15.	140	144	140	145
16.	150	158	150	155
17.	160	170	161	168
18.	170	179	172	180
19.	180	185	184	190



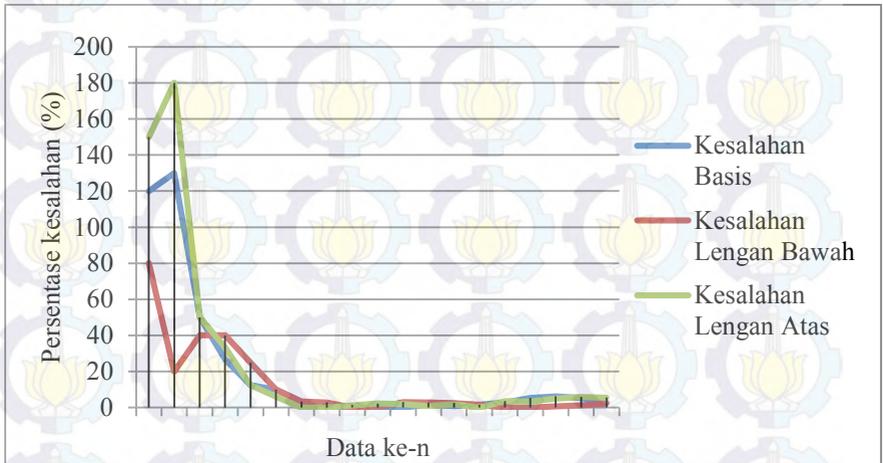
Gambar 4.2 Grafik Data Pengukuran *Output Servo*

Pada percobaan yang dilakukan, didapat bahwa masing-masing motor *servo* memiliki kesalahan dalam menghasilkan sudut. Kesalahan tersebut dapat dilihat pada Gambar 4.2.

Tabel 4.4 Persentase Kesalahan Sudut *Servo*

No.	Kesalahan Basis (%)	Kesalahan Lengan Bawah (%)	Kesalahan Lengan Atas (%)
1.	120	80	150
2.	130	20	180
3.	50	40	50
4.	26.66666667	40	33.33333333
5.	12.5	25	12.5
6.	10	10	6
7.	3.333333333	3.333333333	0
8.	0	2.857142857	0
9.	0	0	1.25
10.	0	0	2.222222222
11.	0	3	2
12.	1.818181818	2.727272727	0.909090909
13.	0.833333333	2.5	1.666666667
14.	1.538461538	1.538461538	0
15.	2.857142857	0	3.571428571

16.	5.333333333	0	3.333333333
17.	6.25	0.625	5
18.	5.294117647	1.176470588	5.882352941
19.	2.777777778	2.222222222	5.555555556
Jumlah	379,202	234,979	432,281



Gambar 4.3 Grafik Persentase Kesalahan Sudut *Servo*

Dari Tabel 4.4, dapat dibuat rata-rata kesalahan dari masing-masing servo berdasarkan rumus berikut

$$\text{Rata - rata kesalahan} = \frac{\text{Jumlah data}}{\text{banyak data}} \times 100\% \quad (3)$$

Rata-rata kesalahan dari *servo* basis = 19,95801833%

Rata-rata kesalahan dari *servo* lengan bawah = 12,36736333%

Rata-rata kesalahan dari *servo* lengan atas = 22,751672%.

Kemudian persentase kesalahan sudut *servo* dinyatakan dalam bentuk grafik seperti pada Gambar 4.3.

4.3 Pengujian Keseluruhan Alat

Pengujian keseluruhan alat ini bertujuan untuk mengetahui jumlah biskuit yang terambil, tidak terambil, dan hancur. Biskuit terdiri dari warna coklat muda dan tua yang diletakkan pada lubang berlabel A-H. Sebelum

melakukan pengujian perlu diketahui bahwa terdapat jumlah biskuit dalam sekali eksekusi program yang disajikan dalam tabel berikut,

Tabel 4.5 Pengujian Jumlah Biskuit

	A	B	C	D	E	F	G	H	Jumlah
Coklat muda	4	4	4	2	-	-	-	-	14
Coklat tua	-	-	-	-	4	4	4	3	15

Tabel 4.5 merupakan pengujian jumlah biskuit dalam satu kaleng apabila tidak terjadi kesalahan.

Pengujian dilakukan dengan menjalankan keseluruhan program sebanyak 2 kali. Dari pengujian pertama didapatkan pada Tabel 4.6 berikut ini,

Tabel 4.6 Hasil Pengujian ke-1

No.	Jenis Biskuit	Terambil	Tidak Terambil tidak hancur
1.	Coklat Muda	5	9
2.	Coklat Tua	6	9
JUMLAH		11	17

Kemudian pengujian kedua didapatkan pada Tabel 4.7 berikut ini,

Tabel 4.7 Hasil Pengujian ke-2

No.	Jenis Biskuit	Terambil	Tidak Terambil tidak hancur
1.	Coklat Muda	4	10
2.	Coklat Tua	8	7
JUMLAH		12	15

Berdasarkan pengujian yang dilakukan, terdapat 2 keadaan yang dialami oleh biskuit karena perlakuan lengan robot yaitu terambil dan tidak terambil. Sebab yang menjadi biskuit terambil atau tidak adalah posisi akhir dari biskuit setelah melalui *conveyor* ke kedudukan biskuit. Hanya pada posisi-posisi tertentu saja yang mengakibatkan biskuit dapat terambil.

Jika mengacu pada ketiga tabel diatas, maka dapat dibuat rata-rata biskuit tidak terambil dan persentase biskuit yang tidak terambil.

$$\frac{\text{Jumlah seharusnya} - \text{biskuit terambil}}{\text{Jumlah seharusnya}} \times 100\% \quad (4)$$

Persentase biskuit coklat muda yang tidak terambil = 67,857%

Persentase biskuit coklat tua yang tidak terambil = 53,333%

BAB V PENUTUP

Dari hasil pengujian motor *servo*, dapat diambil kesimpulan bahwa metode yang digunakan tersebut terdapat rata-rata kesalahan dari *servo* basis sebesar 19,95801833%, *servo* lengan bawah sebesar 12,36736333%, *servo* lengan atas sebesar 22,751672%. Dari pengujian sensor pendeteksi biskuit yang dilakukan, hasil dari sensor fotodioda tidak berupa tegangan 0 dan 5 volt sebagai logika 0 dan 1. Tetapi rata-rata sebesar 4,7568 volt saat logika 1 dan 0,1344 volt saat logika 0. Namun, hal ini tetap dapat dianggap logika 0 dan 1 oleh Arduino. Hal yang menyebabkan sensor tidak benar-benar 0 dan 5 volt adalah karena pada saat terhalang oleh biskuit, fotodioda masih mendapat cahaya dari lingkungan sekitar. Dari pengujian keseluruhan alat, didapatkan persentase biskuit yang tidak terambil diantaranya biskuit coklat muda sebesar 67,857% dan coklat tua sebesar 53,333%. Penyebab biskuit tidak terambil adalah posisi akhir dari biskuit setelah melalui *conveyor* tidak selalu sama dengan posisi akhir lengan robot.

Saran untuk penelitian selanjutnya adalah sebaiknya digunakan motor dengan akurasi tinggi untuk gerak sudut. Selain itu, terdapat beberapa metode dalam penentuan gerak lengan robot diantaranya *inverse kinematics*, *forward kinematics*, atau *Denavit Hartenberg* yang dapat digunakan pada lengan robot industri. Mengenai sensor pendeteksi biskuit, sebaiknya dapat menggunakan sensor yang dapat menentukan logika 1 atau 0 dengan tepat. Untuk posisi akhir biskuit setelah melalui *conveyor*, sebaiknya dibuatkan dudukan yang sesuai dengan ukuran biskuit sehingga posisi akhir dari biskuit dapat sama dengan posisi akhir lengan robot. Dengan demikian, semua biskuit dapat terambil oleh lengan robot.

DAFTAR PUSTAKA

- [1] _____, *Industri Biskuit di Indonesia Tumbuh 8,8% pada 2008 Juni 2009*, <http://www.datacon.co.id/MielInstan-2009Biskuit.html>, Diakses pada tanggal 21 Oktober 2015
- [2] Nadia, A., dan Afnan, S. , "Tangan Robot Peniru Gerak Tangan Manusia Berbasis Mikrokontroler", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.
- [3] _____, *Motor Servo*, <http://elektronika-dasar.web.id/motor-servo/>, Diakses pada tanggal 21 Oktober 2015.
- [4] _____, *Diode Pancaran Cahaya*, https://id.wikipedia.org/wiki/Diode_pancaran_cahaya, Diakses pada tanggal 21 Oktober 2015.
- [5] _____, *Fotodiode*, <https://id.wikipedia.org/wiki/Fotodiode>, Diakses pada tanggal 21 Oktober 2015.
- [6] _____, *UArm Assembly Instructions*, <http://www.docfoc.com/uarm-assembly-instructions-v123>, Diakses pada tanggal 11 November 2015.
- [7] Saleh, Mashad Uddin., dkk., 2013, *Design and Implementation Of a Simple, Low-Cost Robotic Arm*, www.ijareeie.com/upload/2013/october/2EDESIGN.pdf, Diakses pada tanggal 11 November 2015.
- [8] _____, *TowerPro MG946R Servo*, <http://www.servodatabase.com/servo/towerpro/mg946r>, Diakses pada tanggal 12 April 2016.
- [9] _____, *Servo Motor SG90*, abc-rc.pl/templates/images/files/.../1428085018-sg-90-tower.pdf, Diakses pada tanggal 12 April 2016.
- [10] _____, *ATmega48A/PA/88A/PA/168A/PA/328/P*, http://www.atmel.com/images/atmel-8271-8-bit-avr-microcontroller-atmega48a-48pa-88a-88pa-168a-168pa-328-328p_datasheet_summary.pdf, Diakses pada tanggal 12 April 2016.



LAMPIRAN A

A.1 Listing Program

```
#include <Servo.h>

Servo basis;
Servo lbawah;
Servo atas;
Servo ujung;
Servo penjepit;
int a,b,c,d,e,f,g,h;
int con1 = 13;
int con2 = 12;
int sen1 = A1;
int sen2 = A0;
int con1jb = 0;
int con2jb = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  a=0;
  b=0;
  c=0;
  d=0;
  e=0;
  f=0;
  g=0;
  h=0;
  pinMode(sen2, INPUT);
  pinMode(sen1, INPUT);
  pinMode(sen2, INPUT);
  pinMode(con1, OUTPUT);
  pinMode(con2, OUTPUT);
  digitalWrite(12, HIGH);
  digitalWrite(13, HIGH);
  diatas_conveyor();
}
```

```
void jepit()
{
  //ujung.attach(5);
  //ujung.write(10);
  penjepit.attach(6);
  penjepit.write(50);
}
```

```
void buka()
{
  ujung.attach(5);
  ujung.write(0);
  penjepit.attach(6);
  penjepit.write(70);
}
```

```
void diatas_conveyor()
{
  //delay(1000);
  basis.attach(9);
  basis.write(70);
  lbawah.attach(10);
  lbawah.write(80);
  latas.attach(11);
  latas.write(130);
  ujung.attach(5);
  ujung.write(10);
  //delay(1000);
}
```

```
void diatas_conveyor1()
{
  delay(1000);
  basis.attach(9);
  basis.write(70);
  lbawah.attach(10);
  lbawah.write(100);
  latas.attach(11);
}
```

```
latas.write(130);  
ujung.attach(5);  
ujung.write(10);  
delay(1000);  
}
```

```
void diatas_conveyor2()
```

```
{  
basis.attach(9);  
basis.write(60);  
lbawah.attach(10);  
lbawah.write(50);  
latas.attach(11);  
latas.write(120);  
ujung.attach(5);  
ujung.write(150);  
}
```

```
void posisiC1()
```

```
{  
didas_conveyor1();  
delay(1000);  
basis.attach(9);  
basis.write(76);  
lbawah.attach(10);  
lbawah.write(125);  
latas.attach(11);  
latas.write(180);  
ujung.attach(5);  
ujung.write(15);  
delay(1000);  
jepit();  
}
```

```
void posisiC2()
```

```
{  
delay(1000);  
basis.attach(9);  
basis.write(63);  
}
```

```
lbawah.attach(10);  
lbawah.write(70);  
latas.attach(11);  
latas.write(70);  
ujung.attach(5);  
ujung.write(150);  
delay(1000);  
jepit();  
}
```

```
void diatas_kaleng()
```

```
{  
  delay(1000);  
  basis.attach(9);  
  basis.write(150);  
  lbawah.attach(10);  
  lbawah.write(80);  
  latas.attach(11);  
  latas.write(130);  
  delay(1000);  
}
```

```
void kalengA()
```

```
{  
  buka();  
  diatas_conveyor();  
  delay(1000);  
  posisiC1();  
  delay(1000);  
  jepit();  
  delay(1000);  
  diatas_conveyor();  
  diatas_kaleng();  
  delay(1000);  
  basis.attach(9);  
  basis.write(153);  
  lbawah.attach(10);  
  lbawah.write(90);  
  latas.attach(11);  
}
```

```
latas.write(100);  
delay(1000);  
buka();  
}
```

```
void kalengB()  
{
```

```
  buka();  
  diatas_conveyor();  
  delay(1000);  
  posisiC1();  
  delay(1000);  
  jepit();  
  delay(1000);  
  diatas_conveyor();  
  diatas_kaleng();  
  delay(1000);  
  basis.attach(9);  
  basis.write(145);  
  lbawah.attach(10);  
  lbawah.write(90);  
  latas.attach(11);  
  latas.write(90);  
  delay(1000);  
  buka();  
}
```

```
void kalengC()  
{
```

```
  buka();  
  diatas_conveyor();  
  delay(1000);  
  posisiC1();  
  delay(1000);  
  jepit();  
  delay(1000);  
  diatas_conveyor();  
  diatas_kaleng();  
  delay(1000);  
}
```

```
basis.attach(9);
basis.write(140);
lbawah.attach(10);
lbawah.write(78);
latas.attach(11);
latas.write(84);
delay(1000);
buka();
}
```

```
void kalengD()
```

```
{
  buka();
  diatas_conveyor();
  delay(1000);
  posisiC1();
  delay(1000);
  jepit();
  delay(1000);
  diatas_conveyor();
  diatas_kaleng();
  delay(1000);
  basis.attach(9);
  basis.write(155);
  lbawah.attach(10);
  lbawah.write(83);
  latas.attach(11);
  latas.write(85);
  delay(1000);
  buka();
}
```

```
void kalengE()
```

```
{
  buka();
  diatas_conveyor();
  delay(1000);
  diatas_conveyor2();
  delay(1000);
}
```

```
posisiC2();
delay(1000);
jepit();
delay(300);
diatas_conveyor2();
delay(300);
diatas_conveyor();
diatas_kaleng();
delay(1000);
basis.attach(9);
basis.write(148);
lbawah.attach(10);
lbawah.write(73);
latas.attach(11);
latas.write(79);
delay(1000);
buka();
}
```

```
void kalengF()
```

```
{
  buka();
  diatas_conveyor();
  delay(1000);
  diatas_conveyor2();
  delay(1000);
  posisiC2();
  delay(1000);
  jepit();
  delay(300);
  diatas_conveyor2();
  delay(300);
  diatas_conveyor();
  diatas_kaleng();
  delay(1000);
  basis.attach(9);
  basis.write(167);
  lbawah.attach(10);
  lbawah.write(73);
}
```

```
latas.attach(11);  
latas.write(82);  
delay(1000);  
buka();  
}
```

```
void kalengG()  
{
```

```
  buka();  
  diatas_conveyor();  
  delay(1000);  
  diatas_conveyor2();  
  delay(1000);  
  posisiC2();  
  delay(1000);  
  jepit();  
  delay(300);  
  diatas_conveyor2();  
  delay(300);  
  diatas_conveyor();  
  diatas_kaleng();  
  delay(1000);  
  basis.attach(9);  
  basis.write(175);  
  lbawah.attach(10);  
  lbawah.write(85);  
  latas.attach(11);  
  latas.write(80);  
  delay(1000);  
  buka();  
}
```

```
void kalengH()  
{
```

```
  buka();  
  diatas_conveyor();  
  delay(1000);  
  diatas_conveyor2();  
  delay(1000);
```

```

posisiC2();
delay(300);
diatas_conveyor2();
delay(300);
jepit();
delay(1000);
diatas_conveyor();
diatas_kaleng();
delay(1000);
basis.attach(9);
basis.write(167);
lbawah.attach(10);
lbawah.write(90);
latas.attach(11);
latas.write(98);
delay(1000);
buka();
}

```

```

void loop() {
// put your main code here, to run repeatedly:
con1jb = digitalRead(A1);
con2jb = digitalRead(A0);
if ((con1jb == LOW) && (con2jb == HIGH)){
Serial.println("con1 berhenti");
digitalWrite(con1, LOW);
digitalWrite(con2, LOW);
if (a<=3){
a=a+1;
Serial.println("A");
Serial.print(a);
diatas_kaleng();
delay(1000);
kalengA();
}
else if ((a>=3) && (b<=3)){
b=b+1;
Serial.println("B");
Serial.print(b);
}
}
}

```

```

    diatas_kaleng();
    delay(1000);
    kalengB();
}
else if ((a>=3) && (b>=3) && (c<=3)){
    c=c+1;
    Serial.println("C");
    Serial.print(c);
    diatas_kaleng();
    delay(1000);
    kalengC();
}
else if ((a>=3) && (b>=3) && (c>=3) && (d<=1)){
    d=d+1;
    Serial.println("D");
    Serial.print(d);
    diatas_kaleng();
    delay(1000);
    kalengD();
}
}
else if ((con2j2b == LOW) && (con1j2b == HIGH)){
    Serial.println("con2 berhenti");
    digitalWrite(con1, LOW);
    digitalWrite(con2, LOW);
    if (e<=3){
        e=e+1;
        Serial.println("E");
        Serial.print(e);
        diatas_kaleng();
        delay(1000);
        kalengE();
    }
else if ((a>=3) && (b>=3) && (c>=3) && (d>=2) && (e>=3) &&
(f<=3)){
    f=f+1;
    Serial.println("F");
    Serial.print(f);
    diatas_kaleng();
}

```




LAMPIRAN B

B.1. Mg946r

TowerPro MG946R Servo

Basic Information

Modulation:	Digital
Torque:	4.8V: 145.8 oz-in (10.50 kg-cm) 6.0V: 180.5 oz-in (13.00 kg-cm)
Speed:	4.8V: 0.20 sec/60° 6.0V: 0.17 sec/60°
Weight:	1.94 oz (55.0 g)
Dimensions:	Length: 1.60 in (40.7 mm) Width: 0.78 in (19.7 mm) Height: 1.69 in (42.9 mm)
Motor Type:	3-pole
Gear Type:	Metal
Rotation/Support:	? (add)

Additional Specifications

Rotational Range:	? (add)
Pulse Cycle:	1 ms
Pulse Width:	? (add)
Connector Type:	JR

It's gone. [Undo](#)



Brand:	Towerpro
Product Number:	? (add)
Suggested Retail:	? (add)
Street Price:	10.50 USD
Compare:	add

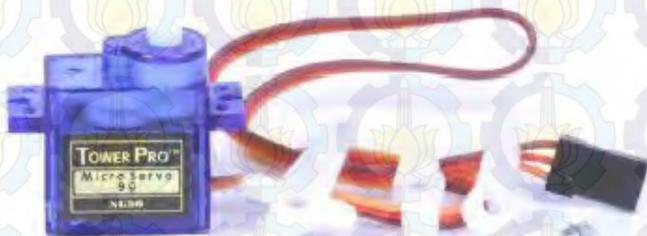
User Reviews

Number of Reviews:	5
Average Rating:	3.2 / 5.0

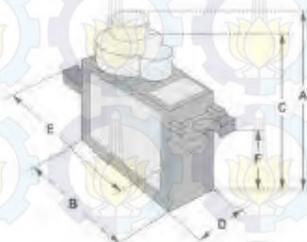
B.2. Sg90

SERVO MOTOR SG90

DATA SHEET



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear bar, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "180" (~1ms pulse) is all the way to the left.

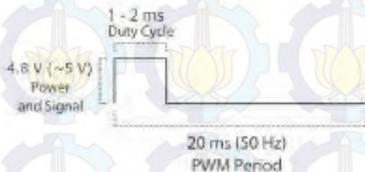
Dimensions & Specifications

A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

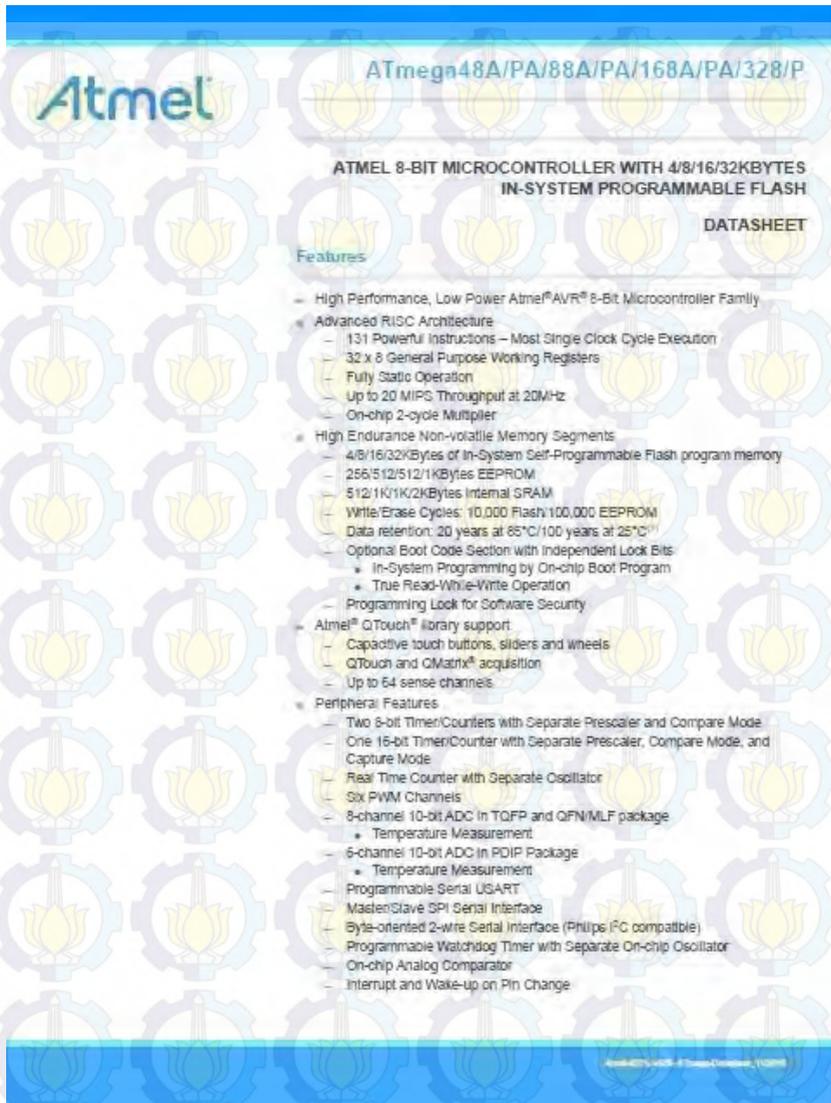
PWM=Orange (J1)

Vcc=Red (+)

Ground=Brown (-)



B.3. ATmega328



Atmel

ATmega48A/PA/88A/PA/168A/PA/328/P

ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH

DATASHEET

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller Family
- **Advanced RISC Architecture**
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- **High Endurance Non-volatile Memory Segments**
 - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
 - 256/512/512/1KBytes EEPROM
 - 512/1K/1K/2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - 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® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix® acquisition
 - Up to 64 sense channels
- **Peripheral Features**
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change

© 2006 Atmel Corporation. All rights reserved. Atmel, the Atmel logo, AVR, AVR logo, QTouch, and QMatrix are registered trademarks or trademarks of Atmel Corporation in the United States and other countries.



• 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

- 23 Programmable I/O Lines
- 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:

- 1.8 - 5.5V

• Temperature Range:

- -40°C to 85°C

• Speed Grade:

- 0 - 4MHz @ 1.8 - 5.5V, 0 - 10MHz @ 2.7 - 5.5V, 0 - 20MHz @ 4.5 - 5.5V

• Power Consumption at 1MHz, 1.8V, 25°C

- Active Mode: 0.2mA
- Power-down Mode: 0.1µA
- Power-save Mode: 0.75µA (including 32kHz RTC)


 ATmega48A/PA/88A/PA/168A/PA/328/P [DATASHEET]
 2

©2006 Atmel Corporation. All rights reserved.

1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

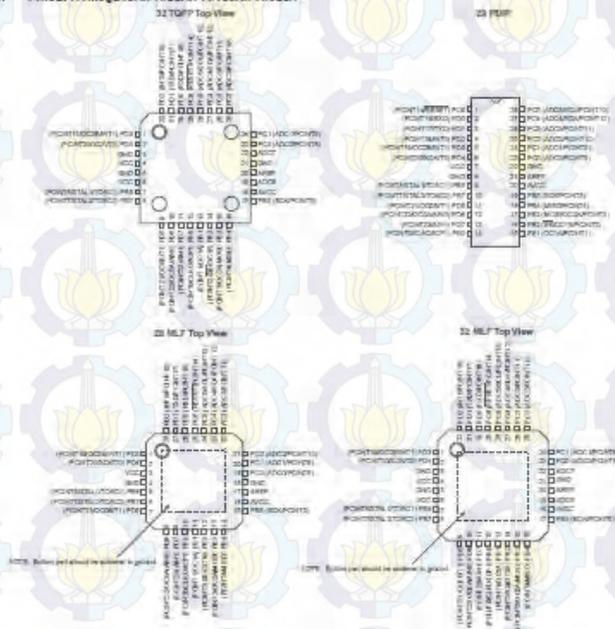


Table 1-1. 32UFPGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC3	PC1
B	PD3	PD4	PDD	PC5	PC2	PC0
C	GND	GND			AD07	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	A/D D	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tristated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the Inverting Oscillator amplifier and input to the Internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the Inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7...6 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 27.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tristated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 29-11 on page 305. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tristated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 86.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3D, and ADC7:6. It should be externally connected to V_{DD} even if the ADC is not used. If the ADC is used, it should be connected to V_{DD} through a low-pass filter. Note that PC6...4 use digital supply voltage, V_{DD}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.



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

DAFTAR RIWAYAT HIDUP



Nama : Hendry Erwantono
TTL : Gresik, 09 Juni 1994
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jalan Walet Raya FA 22 GKA
Gresik
Telp/HP : 085746407778
E-mail : hendry.erwantono@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SDN Randu Agung 3
2. 2007 – 2010 : SMPN 1 Gresik
3. 2010 – 2013 : SMAN 1 Gresik
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Petrowidada Gresik

PENGALAMAN ORGANISASI

1. Staff Departemen Dalam Negeri Periode 2014/2015 HIMAD3TEKTRO, FTI – ITS
2. Ketua Departemen Dalam Negeri Periode 2015/2016 HIMAD3TEKTRO, FTI – ITS
3. Koordinator Keamanan dan Perijinan *Industrial Automation and Robotic Competition* (IARC 2015)

DAFTAR RIWAYAT HIDUP



Nama : Evi Siska Safarina
TTL : Malang, 12 Juli 1994
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jl. Gondowangi RT. 06 RW.
01 Kec. Wagir Kab. Malang
Telp/HP : 08563332903
E-mail : evi.safarina@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SD Negeri Gondowangi 01
2. 2007 – 2010 : SMP Negeri 1 Wagir
3. 2010 – 2013 : MA Negeri 3 Malang
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Oriental Abadi Indonesia, Surabaya

PENGALAMAN ORGANISASI

1. Asisten Sekretaris Periode 2014/2015 HIMAD3TEKTRO, FTI – ITS
2. Sekretaris Umum Periode 2015/2016 HIMAD3TEKTRO, FTI – ITS
3. Anggota Sie Dana *Industrial Automation and Robotic Competition* (IARC 2014)
4. Sekretaris *Industrial Automation and Robotic Competition* (IARC 2015)