

**PERANCANGAN EXOSKELETON MOTION CAPTURE SYSTEM  
SEBAGAI PANDUAN GERAKAN TARI PADA ROBOT HUMANOID**

**KRSI**

**SKRIPSI**

Diajukan Untuk Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Sarjana Teknik



**DISUSUN OLEH :**

**TANSHUDA ALFAUZI**

**NIM. 105060300111018-63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2014**

## PENGANTAR

*Alhamdulillah*, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala petunjuk dan nikmat-Nya lah skripsi ini dapat diselesaikan. Skripsi berjudul “Perancangan *Exoskeleton Motion Capture System* Sebagai Panduan Gerakan Tari Pada Robot *Humanoid* KRSI” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Allah SWT atas rahmat dan hidayah yang telah diberikan,
- Rasulullah Muhammad SAW, semoga shalawat serta salam selalu tercurah kepada beliau,
- Ayah dan Ibu atas segala inspirasi, nasehat, kasih sayang, perhatian dan kesabarannya didalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
- Kakak dan keluarga besar atas inspirasinya,
- Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Moch. Rif'an, ST., MT. selaku Ketua Prodi Strata Satu Jurusan Teknik Elektro Universitas Brawijaya serta sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan,
- Alm. Bapak Ir. Chairuzzaini, selaku dosen Penasehat akademik pertama penulis, atas didikan, inspirasi, dan pelajaran berharganya,
- Bapak Ir. Sigit Kusmaryanto, M.Eng. selaku dosen Penasehat Akademik,
- Ibu Ir. Nurussa'adah, MT. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya serta sebagai Dosen Pembimbing I atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan,

- Bapak Ir. Nanang Sulistiyanto, MT. sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan,
- Ibu Sapriesty Nainy Sari, ST., MT. sebagai fasilitator program hibah penelitian mahasiswa Universitas Brawijaya 2014,
- Seluruh dosen pengajar Teknik Elektro Universitas Brawijaya,
- Staff Recording Jurusan Teknik Elektro,
- Teman – teman Magnet angkatan 2010,
- Rekan seperjuangan dalam skripsi, Basori, Abu, Anas, Veri, Viki, Rita, Agwin, Ari, Mas Alvian, Erwan, Zaratul, Firman, Hakiki, Luthfi, Rize, Dito, Ayu, Nunik, Erny, Aya, Maman, terima kasih atas segala bantuan yang telah diberikan,
- Rekan-rekan penerus “Divisi Humanoid” KRSI & KRSBI Tim Robot Teknik Elektro Universitas Brawijaya, Tegar, Emon, Bustanul, Ronny, Firman, Septian, Gabriel, Ferdi, Liza, dan Anggia, terima kasih atas kerja sama, niat, semangat, pengorbanan serta pengalaman berharga selama perjalanan tim tahun ini, dan juga segala bantuan yang telah diberikan,
- Seluruh Keluarga Besar Tim Robot Teknik Elektro Universitas Brawijaya atas segala bantuan alat, bahan, dan masukan – masukan yang telah diberikan,
- Seluruh Keluarga Besar Laboratorium Sistem Digital Teknik Elektro Universitas Brawijaya atas kerjasamanya dan pengalamannya selama masa jabatan asisten laboratorium,
- Rekan-rekan PKM-KC “SUSI” Universitas Brawijaya, Surya, Hanif, Wia, dan Dean, atas pengalaman yang berharga serta inspirasinya,
- Laboratorium Mekatronika dan Robotika, alias Markas Besar Tim Robot Universitas Brawijaya, serta GOR Pertamina Universitas Brawijaya, yang selama ini telah menyediakan tempat tinggal sementara dan sarana bagi penulis dalam mengerjakan skripsi ini,
- Seluruh teman-teman serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu, terimakasih banyak atas bantuan dan dukungannya.



Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, Juli 2014

Penulis



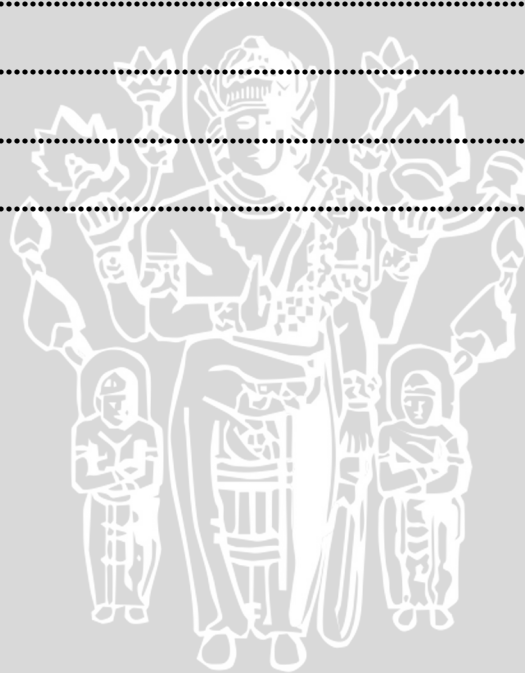
## DAFTAR ISI

|                                                    |            |
|----------------------------------------------------|------------|
| <b>PENGANTAR .....</b>                             | <b>i</b>   |
| <b>DAFTAR ISI.....</b>                             | <b>iv</b>  |
| <b>DAFTAR GAMBAR.....</b>                          | <b>vii</b> |
| <b>DAFTAR TABEL .....</b>                          | <b>ix</b>  |
| <b>ABSTRAK .....</b>                               | <b>x</b>   |
| <b>PENDAHULUAN.....</b>                            | <b>1</b>   |
| 1.1 Latar Belakang .....                           | 1          |
| 1.2 Rumusan Masalah .....                          | 2          |
| 1.3 Batasan Masalah .....                          | 2          |
| 1.4 Tujuan .....                                   | 3          |
| 1.5 Sistematika Penulisan .....                    | 3          |
| <b>TINJAUAN PUSTAKA.....</b>                       | <b>5</b>   |
| 2.1 Kontes Robot Seni Indonesia (KRSI) .....       | 5          |
| 2.2 Robot Humanoid.....                            | 6          |
| 2.3 <i>Exoskeleton Motion Capture System</i> ..... | 8          |
| 2.4 Potensiometer.....                             | 11         |
| 2.5 <i>Microcontroller ATmega16</i> .....          | 14         |
| 2.5.1 <i>ADC Microcontroller</i> .....             | 15         |
| 2.5.2 <i>Pulse Width Modulation (PWM)</i> .....    | 16         |
| 2.5.3 Komunikasi Serial.....                       | 17         |
| 2.6 Motor DC Servo.....                            | 18         |
| 2.7 USB to TTL .....                               | 21         |
| <b>METODE PENELITIAN .....</b>                     | <b>22</b>  |

|                                             |                                                                     |           |
|---------------------------------------------|---------------------------------------------------------------------|-----------|
| 3.1                                         | Studi Literatur .....                                               | 22        |
| 3.2                                         | Penentuan Spesifikasi Alat .....                                    | 22        |
| 3.3                                         | Perancangan dan Pembuatan Alat.....                                 | 23        |
| 3.3.1                                       | Perancangan dan Pembuatan Mekanik .....                             | 23        |
| 3.3.2                                       | Perancangan dan Pembuatan Perangkat Keras ( <i>Hardware</i> ) ..... | 23        |
| 3.3.3                                       | Perancangan dan Pembuatan Perangkat Lunak ( <i>Software</i> ) ..... | 23        |
| 3.4                                         | Pengujian Alat.....                                                 | 23        |
| 3.4.1                                       | Pengujian Sub Sistem .....                                          | 24        |
| 3.4.2                                       | Pengujian Keseluruhan Sistem .....                                  | 24        |
| 3.5                                         | Pengambilan Kesimpulan .....                                        | 24        |
| <b>PERANCANGAN DAN PEMBUATAN ALAT .....</b> |                                                                     | <b>25</b> |
| 4.1                                         | Perancangan Mekanik.....                                            | 25        |
| 4.1.1                                       | Perancangan Derajat Kebebasan Sistem.....                           | 25        |
| 4.1.2                                       | Perancangan Mekanik <i>Exoskeleton</i> .....                        | 27        |
| 4.1.3                                       | Perancangan Mekanik Robot <i>Humanoid</i> .....                     | 29        |
| 4.2                                         | Perancangan <i>Hardware</i> .....                                   | 30        |
| 4.2.1                                       | Perancangan Rangkaian Antarmuka Potensiometer .....                 | 32        |
| 4.2.2                                       | Perancangan Rangkaian <i>Mainboard Microcontroller</i> .....        | 32        |
| 4.2.3                                       | Perancangan Rangkaian ATmega16 SMD .....                            | 33        |
| 4.3                                         | Perancangan Perangkat Lunak.....                                    | 34        |
| 4.3.1                                       | Perancangan Perangkat Lunak Utama .....                             | 34        |
| 4.3.2                                       | Perancangan Perangkat Lunak Konversi ADC.....                       | 37        |
| 4.3.3                                       | Perancangan Perangkat Lunak Konversi Nilai ADC ke Sudut.....        | 37        |
| 4.3.4                                       | Perancangan Perangkat Lunak Konversi Sudut ke PWM .....             | 38        |
| <b>PENGUJIAN DAN ANALISIS .....</b>         |                                                                     | <b>42</b> |
| 5.1                                         | Pengujian <i>Potensiometer</i> .....                                | 42        |



|                                   |                                            |           |
|-----------------------------------|--------------------------------------------|-----------|
| 5.2                               | Pengujian <i>Serial Terminal</i> .....     | 45        |
| 5.3                               | Pengujian ADC <i>microcontroller</i> ..... | 46        |
| 5.4                               | Pengujian sinyal PWM .....                 | 49        |
| 5.5                               | Pengujian motor DC <i>servo</i> .....      | 52        |
| 5.6                               | Pengujian Keseluruhan Sistem .....         | 54        |
| <b>KESIMPULAN DAN SARAN .....</b> |                                            | <b>58</b> |
| 6.1                               | Kesimpulan .....                           | 58        |
| 6.2                               | Saran .....                                | 59        |
| <b>DAFTAR PUSTAKA.....</b>        |                                            | <b>60</b> |
| <b>LAMPIRAN I.....</b>            |                                            | <b>61</b> |
| <b>LAMPIRAN II.....</b>           |                                            | <b>63</b> |
| <b>LAMPIRAN III.....</b>          |                                            | <b>70</b> |
| <b>LAMPIRAN IV .....</b>          |                                            | <b>71</b> |



## DAFTAR GAMBAR

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Gambar 2.1 Robot <i>Humanoid</i> dalam perlombaan KRSI 2014 .....         | 6  |
| Gambar 2.2 Derajat Kebebasan Aldebaran Nao .....                          | 8  |
| Gambar 2.3 Exoskeleton motion capture system .....                        | 10 |
| Gambar 2.4 Potensiometer putar .....                                      | 11 |
| Gambar 2.5 Potensiometer sebagai pembagi tegangan .....                   | 12 |
| Gambar 2.6 Analisis rangkaian potensiometer .....                         | 13 |
| Gambar 2.7 <i>Microcontroller</i> AVR ATmega16 .....                      | 15 |
| Gambar 2.8 <i>Fast PWM mode</i> .....                                     | 16 |
| Gambar 2.9 Format frame data serial UART .....                            | 17 |
| Gambar 2.10 Diagram motor DC <i>Servo</i> .....                           | 18 |
| Gambar 2.11 Bagian internal motor DC <i>servo</i> .....                   | 19 |
| Gambar 2.12 Sinyal control motor DC <i>servo</i> .....                    | 20 |
| Gambar 2.13 Rangkaian USB to TTL CP2102 .....                             | 21 |
| Gambar 4.1 Alur perancangan sistem .....                                  | 25 |
| Gambar 4.2 Desain derajat kebebasan sistem .....                          | 26 |
| Gambar 4.3 Desain mekanik <i>exoskeleton</i> .....                        | 28 |
| Gambar 4.4 Desain bagian belakang robot <i>humanoid</i> .....             | 29 |
| Gambar 4.5 Desain bagian depan robot <i>humanoid</i> .....                | 30 |
| Gambar 4.6 Diagram blok <i>hardware</i> sistem secara keseluruhan .....   | 31 |
| Gambar 4.7 Skematik Rangkaian Antarmuka Potensiometer .....               | 32 |
| Gambar 4.8 Skematik Rangkaian <i>Mainboard Microcontroller</i> .....      | 33 |
| Gambar 4.9 Skematik Rangkaian ATmega16 SMD .....                          | 34 |
| Gambar 4.10 Diagram Alir Program Utama .....                              | 35 |
| Gambar 4.11 Diagram konversi program .....                                | 36 |
| Gambar 4.12 Diagram alir program konversi ADC .....                       | 37 |
| Gambar 4.13 Diagram alir program konversi nilai ADC ke sudut .....        | 38 |
| Gambar 4.14 Diagram alir program konversi sudut ke nilai TOP .....        | 39 |
| Gambar 4.15 Sinyal kontrol motor DC <i>servo</i> .....                    | 39 |
| Gambar 4.16 Sinyal kontrol motor DC <i>servo</i> jamak pada pin I/O ..... | 40 |
| Gambar 5.1 Diagram Blok Pengujian Potensiometer .....                     | 42 |

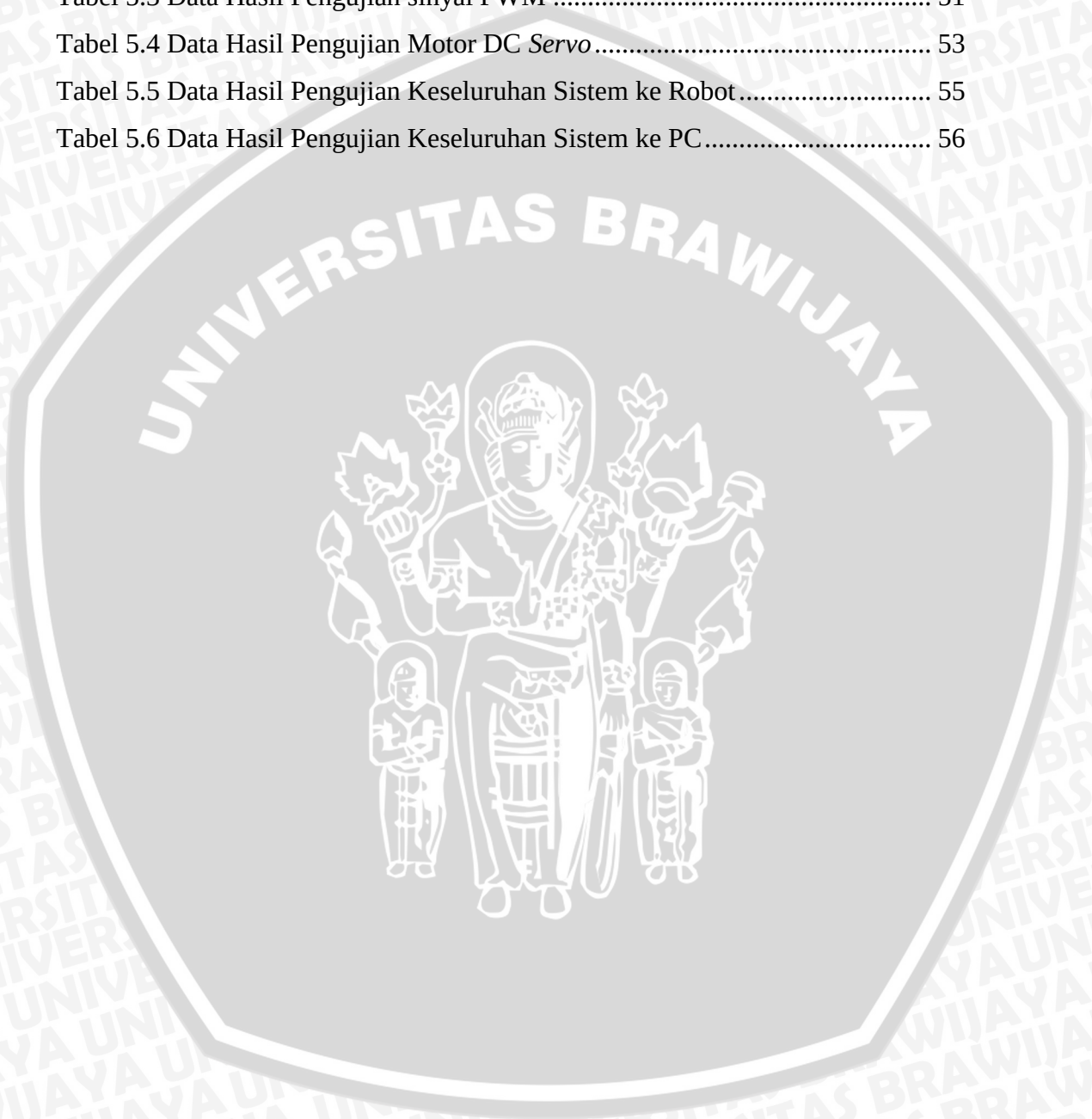


|                                                              |    |
|--------------------------------------------------------------|----|
| Gambar 5.2 Grafik Pengujian Potensiometer.....               | 45 |
| Gambar 5.3 Skema Pengujian USB to TTL.....                   | 46 |
| Gambar 5.4 Hasil Pengujian USB to TTL .....                  | 46 |
| Gambar 5.5 Skema Pengujian ADC.....                          | 47 |
| Gambar 5.6 Diagram alir program pengujian ADC.....           | 47 |
| Gambar 5.7 Grafik Pengujian ADC <i>Microcontroller</i> ..... | 49 |
| Gambar 5.8 Skema Pengujian Sinyal PWM .....                  | 50 |
| Gambar 5.9 Tampilan Pengukuran Sinyal PWM.....               | 50 |
| Gambar 5.10 Grafik Pengujian Sinyal PWM.....                 | 52 |
| Gambar 5.11 Skema Pengujian Motor DC <i>Servo</i> .....      | 53 |
| Gambar 5.12 Grafik Pengujian Motor DC <i>Servo</i> .....     | 54 |



## DAFTAR TABEL

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Tabel 5.1 Data Hasil Pengujian Potensiometer.....               | 43 |
| Tabel 5.2 Data Hasil Pengujian ADC .....                        | 48 |
| Tabel 5.3 Data Hasil Pengujian sinyal PWM .....                 | 51 |
| Tabel 5.4 Data Hasil Pengujian Motor DC <i>Servo</i> .....      | 53 |
| Tabel 5.5 Data Hasil Pengujian Keseluruhan Sistem ke Robot..... | 55 |
| Tabel 5.6 Data Hasil Pengujian Keseluruhan Sistem ke PC.....    | 56 |



## ABSTRAK

**Tanshuda Alfauzi**, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juli 2014, *Perancangan Exoskeleton Motion Capture System Sebagai Panduan Gerakan Tari Pada Robot Humanoid KRSI*, Dosen Pembimbing: Nurussa'adah, Ir., MT. dan Nanang Sulistiyanto, Ir., MT.

Kontes Robot Seni Indonesia (KRSI) merupakan suatu ajang kompetisi perancangan dan pembuatan Robot yang disertai dengan unsur-unsur seni dan budaya. Divisi KRSI mengambil tema robot penari untuk menampilkan seni budaya yang diinginkan sesuai dengan tema setiap tahun. Robot KRSI berjenis *humanoid*, harus mampu menari. Sistem perekam gerakan yang bisa langsung dipasang pada penari profesional atau orang yang mengerti bagaimana gerakan yang artistik dibutuhkan dengan tujuan mendapatkan gerakan Robot yang mendekati aslinya. Jadi, robot *humanoid* akan bergerak mengikuti gerakan yang kita lakukan, kemudian kita bisa mengetahui posisi sendi dari gerakan kita. Maka dari itu, dirancanglah suatu sistem *exoskeleton motion capture*, sensor sudut diletakkan di sendi yang ingin dukur untuk mengetahui pergerakan dari sendi tersebut.

Berdasarkan pengujian yang dilakukan, sensor potensiometer memiliki sudut efektif sebesar  $265^{\circ}$ , dengan batas bawah  $15^{\circ}$  dan batas atas  $280^{\circ}$ , kesalahan terbesar 0,05 V dan kesalahan rata-rata 0,02 V. ADC *microcontroller* mengubah data masukan tegangan dengan rentang 0 - 5,08 V menjadi data digital 8-bit dengan rentang nilai antara 0 - 255, memiliki persen *error* rata-rata 0,71%, dengan persen *error* tertinggi adalah 4,38%, dan persen *error* terendah adalah 0%. Sinyal PWM untuk mengontrol motor DC *servo* mampu dibangkitkan oleh *microcontroller* dengan lebar  $T_{ON}$ , sesuai kebutuhan *servo*, dengan rentang antara 0,6 - 2,5 ms, dengan persen *error* terbesar 0,71% dan persen *error* rata-rata 0,13%. Motor DC *servo* dapat bergerak searah dan berlawanan arah jarum jam, dengan rentang sudut antara  $0^{\circ}$ - $140^{\circ}$ , sinyal PWM yang dapat diterima oleh *servo* adalah sinyal dengan rentang  $T_{ON}$  antara 0,8 ms - 2,2 ms, dengan persen *error* rata-rata 0,5%, persen *error* maksimal 1,67% dan persen *error* minimal 0%. Data dari *microcontroller* ke PC melalui USB to TTL UART berhasil dikirim dengan tingkat keberhasilan 100%. Sendi *exoskeleton* berhasil menggerakkan sendi robot *humanoid* dengan kesalahan rata-rata  $0.59^{\circ}$ , kesalahan terbesar  $1,5^{\circ}$ , dan kesalahan terkecil  $0^{\circ}$ . Sementara sudut yang ditampilkan ke PC memiliki kesalahan rata-rata  $1,13^{\circ}$ , kesalahan terbesar  $4^{\circ}$ , dan kesalahan terkecil  $0^{\circ}$ .

Kata kunci : KRSI, Robot Humanoid, Exoskeleton Motion Capture, Potensiometer, Microcontroller



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Pelaksanaan kontes robot telah berlangsung setiap tahun selama lebih dari satu dekade di Indonesia, salah satunya adalah Kontes Robot Indonesia yang diadakan oleh Direktorat Jenderal Pendidikan Tinggi (Dirjen DIKTI), yang terdiri dari 4 kategori, yaitu Kontes Robot ABU Indonesia (KRAI), Kontes Robot Pemadam Api Indonesia (KRPAI), Kontes Robot Sepak Bola Indonesia (KRSBI), dan Kontes Robot Seni Indonesia (KRSI) yang diselenggarakan setiap tahunnya.

Kontes Robot Seni tari Indonesia (KRSI) merupakan suatu ajang kompetisi perancangan dan pembuatan robot yang disertai dengan unsur-unsur seni dan budaya bangsa yang telah terkenal di bumi pertiwi. Robot tersebut harus dirancang dan dibuat sendiri, dengan menggunakan sensor, aktuator, serta mikroprosesor yang ada dan harus diprogram sesuai dengan tema kontes yang berbeda setiap tahunnya. Robot yang digunakan dalam perlombaan KRSI adalah robot berjenis *humanoid*, yaitu robot yang memiliki bentuk dan karakteristik seperti manusia. Robot diharuskan untuk dapat menari sambil berjalan di arena yang sudah ditentukan selama musik pengiring dimainkan.

Setiap robot dinilai performanya oleh juri dengan beberapa parameter yang tercantum pada buku pedoman peraturan lomba yang diterbitkan oleh DIKTI setiap tahun. Secara garis besar, penilaian performa robot tersebut dapat dibagi menjadi dua parameter, yaitu parameter keindahan, dan kestabilan robot. Parameter kestabilan mewakili kemampuan robot untuk dapat menyelesaikan perlombaan sesuai dengan peraturan perlombaan yang telah diatur dalam *rule* perlombaan, seperti kemampuan robot untuk dapat berjalan melalui setiap zona hingga zona akhir, parameter kestabilan bersifat kuantitatif serta objektif. Sedangkan parameter keindahan mewakili aspek artistik dari robot, biasanya diukur dari keluwesan, keatraktifan, serta sinkronisasi tari robot dengan irama musik pengiring, bersifat kualitatif serta subjektif.

Pada saat pencarian gerakan tari, biasanya tim KRSI Teknik Elektro Universitas Brawijaya melihat contoh gerakan, memperkirakan sudut di setiap

sendi (biasanya kaki tidak diikutsertakan dalam gerakan tari), kemudian diproses oleh *microcontroller* untuk menggerakkan motor DC servo di setiap sendinya. Kekurangan metode seperti ini adalah proses trial yang tidak sebentar untuk mencari gerakan tari sesuai yang diinginkan.

Maka, pada skripsi ini akan dirancang *exoskeleton motion capture* untuk mempermudah dan memaksimalkan proses pemrograman gerakan tari pada robot *humanoid* KRSI. Dengan menggunakan sistem ini, diharapkan gerak tari robot sesuai yang diinginkan dengan cara yang lebih mudah. Pemrograman dapat dilakukan dengan memasang *exoskeleton* yang di setiap sendinya dilengkapi sensor posisi sudut, kemudian data diproses untuk menggerakkan sendi pada robot *humanoid* hingga gerakan robot sesuai yang diinginkan, kemudian data sudut pada setiap persendian dapat ditampilkan ke *Personal Computer*, untuk kemudian dimasukkan ke dalam program pergerakan otomatis robot. Data sudut pada setiap persendian mewakili satu *sequence* gerakan tari. Pemrograman gerak tari juga bisa dilakukan oleh penari yang lebih ahli dengan mengenakan *exoskeleton motion capture system* ini untuk mendapatkan gerakan tari yang lebih artistik.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat disusun rumusan masalah sebagai berikut :

- 1) Bagaimana merancang dan membuat robot *humanoid* KRSI yang dapat menari sesuai *sequence* gerakan dari *exoskeleton motion capture system*.
- 2) Bagaimana merancang dan membuat *exoskeleton motion capture system* untuk mengontrol robot *humanoid* KRSI.
- 3) Bagaimana menterjemahkan masukan dari *exoskeleton motion capture system* menjadi acuan sudut untuk posisi sendi robot *humanoid* KRSI.
- 4) Bagaimana menampilkan data sudut acuan sendi pada *Personal Computer* agar dapat disimpan menjadi informasi sudut untuk satu rangkaian gerakan.

## 1.3 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan penelitian akan diberikan batasan sebagai berikut :



- 1) Gerakan tari diwakili oleh bagian *upper body*, yaitu tangan dan kepala.
- 2) Pembahasan mekanik *exoskeleton* tidak secara mendalam.
- 3) Tidak membahas pembuatan dan pengolahan data pada USB to TTL.
- 4) Tidak membahas kecepatan dan kehalusan gerakan sistem.
- 5) Data sudut pengujian keseluruhan sistem diwakili oleh 1 DOF yang tidak terpasang pada robot *humanoid* dan *exoskeleton*.
- 6) Pengujian 8 DOF keseluruhan sistem ditampilkan pada demonstrasi *live* dan *video*.

#### 1.4 Tujuan

Tujuan penelitian ini adalah merancang dan membuat *exoskeleton motion capture system* sebagai panduan gerakan tari pada robot *humanoid* KRSI.

#### 1.5 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini sebagai berikut :

##### BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

##### BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

##### BAB III Metodologi

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

##### BAB IV Perancangan

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja dan realisasi alat.

##### BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-



hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

## **BAB VI Kesimpulan dan Saran**

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian dimasa yang akan datang.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Kontes Robot Seni Indonesia (KRSI)

Kontes Robot Seni tari Indonesia (KRSI) merupakan suatu ajang kompetisi perancangan, pembuatan dan pemrograman robot yang disertai dengan unsur-unsur seni dan budaya bangsa Indonesia khususnya seni tari yang telah terkenal di bumi pertiwi. KRSI pertamakali diadakan pada tahun 2009 dengan tema "*Robot Penari Jaipong*", tahun 2010 dengan tema "*Robot Penari Pendet*", tahun 2011 dengan tema "*Robot Penari Klono Topeng*", tahun 2012 mengangkat tema "*Robot Penari Piring*" dan tahun 2013 mengangkat tema "*Robot Penari hanuman Duto*". Untuk KRSI 2014, tema yang diangkat adalah "*Robot Penari Legong Keraton*". Kegiatan KRSI 2014 ini dilaksanakan bersamaan dengan pelaksanaan Kontes Robot Indonesia (KRI) tingkat Regional dan KRI tingkat Nasional pada tanggal 14-16 Juni 2014 yang dikoordinasi dan didanai oleh Direktorat Penelitian dan Pengabdian kepada Masyarakat, Direktorat Jenderal Pendidikan Tinggi Kementrian Pendidikan dan Kebudayaan bekerjasama dengan institusi Perguruan Tinggi yang ditunjuk. (Dikti, 2013: 5)

Tujuan dari kontes robot ini adalah untuk menumbuh kembangkan kreatifitas dan minat para mahasiswa dalam teknologi maju, khususnya teknologi robotika yang selain diperuntukkan bagi industri juga diharapkan dapat membantu kegiatan manusia sehari-hari serta seni budaya khususnya sen tari. Setiap tim peserta yang terdiri dari 3(tiga) mahasiswa dengan seorang dosen pembimbing, diwajibkan untuk membuat satu atau beberapa robot yang terkoordinasi untuk menampilkan seni budaya yang diinginkan sesuai tema kontes. (Dikti, 2013: 20)

Robot penari yang digunakan harus menyerupai struktur tubuh manusia dengan tinggi  $50 \pm 5$ cm diukur di posisi kepala. Robot harus memiliki bagian yang dapat disebut sebagai sistem kaki, tubuh, tangan dan kepala. Jumlah derajat kebebasan masing-masing sistem robot minimal 21 (dua puluh satu). Berat robot total maksimum adalah 30kg. Rentang kaki atau tangan robot maksimal tidak boleh lebih 60 cm diukur dari ujung jari tangan/kaki kanan ke kiri ketika

membuka tangan/kaki selebar-lebarnya. Lebar telapak kaki maksimum 150 cm<sup>2</sup> berbentuk elips, lingkaran atau persegi empat. (Dikti, 2013: 21)

Gerak tari harus diselaraskan dengan irama musik pengiring tari "Legong Keraton". Waktu yang disediakan untuk setiap unjuk kebolehan tari dalam lomba ini adalah tiga (3) menit sesuai dengan panjang atau durasi irama gamelan pengiring. Dalam waktu tiga(3) menit, musik pengiring akan berhenti sebanyak dua kali masing-masing selama 10-20 detik. Setiap tim pada setiap *game* diberikan kesempatan *retry*. Setiap *retry* akan dikenakan hukuman pengurangan nilai (*penalty*). (Dikti, 2013:, Hal 25) Gambar 2.1 menunjukkan robot *humanoid* dalam perlombaan KRSI 2014.



Gambar 2.1 Robot *Humanoid* dalam perlombaan KRSI 2014

## 2.2 Robot Humanoid

Dalam budaya populer, istilah “robot” secara umum berkonotasi suatu wujud *anthropomorphic* (seperti manusia), seperti “lengan” robot untuk pengelasan. Kecenderungan untuk berfikir tentang robot sebagai wujud yang seperti manusia mungkin berasal dari asal istilah “robot”. Kata “robot” hadir ke dalam kesadaran populer pada 25 Januari 1921 di Praha, dengan pertunjukan pertama Karel Capek, R.U.R. (Rossum’s Universal Robots). Dalam R.U.R., seorang penemu gaib, Rossum, telah membuat ras pekerja yang bagian biologisnya dibuat dari tong, yang cukup cerdas untuk menggantikan manusia dalam berbagai pekerjaan (karena itu disebut “*universal*”). Capek



mendeskripsikan para pekerja sebagai “robot”, sebuah istilah yang berasal dari bahasa Ceko “robota”, yang dengan bebas diartikan sebagai buruh kasar. Pekerja robot menyiratkan bahwa ciptaan buatan tersebut dimaksudkan untuk menjadi pelayan untuk membebaskan manusia “sungguhan” dari berbagai macam perburuan. (Murphy, 2000: 2-3)

Pergeseran dari ciptaan mekanik berbentuk seperti manusia menjadi bentuk apapun yang membuat pekerjaan selesai sesuai dengan kenyataan yang ada. Saat robot adalah benda mekanik, mereka tidak harus berbentuk manusia atau bahkan berbentuk hewan. Seperti pada *vacuum cleaner*, yang berbentuk seperti *vacuum cleaner*, bukan petugas kebersihan. (Murphy, 2000: 3)

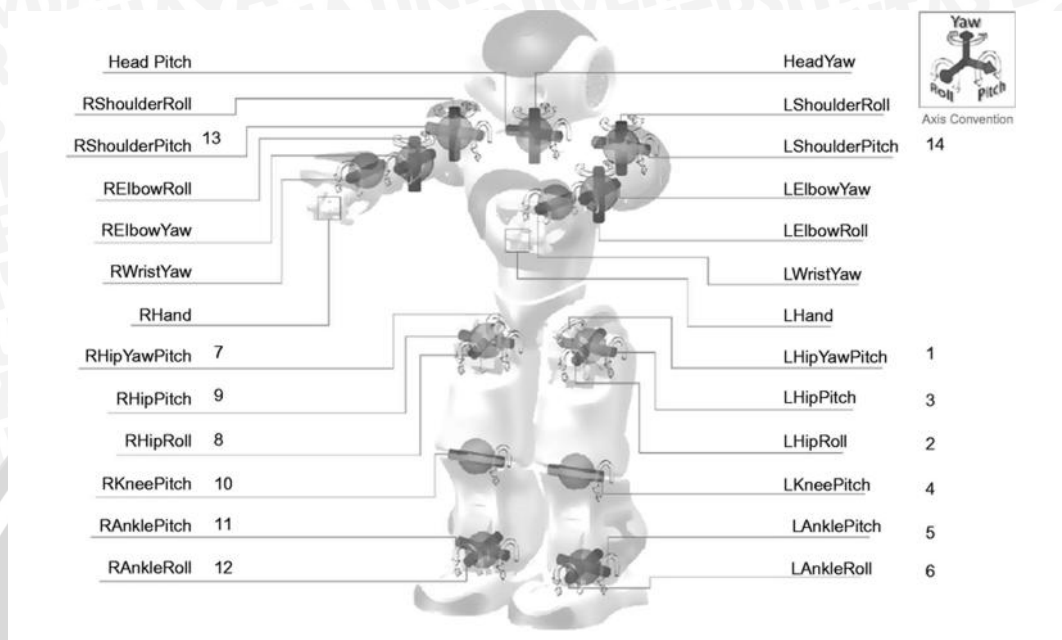
Robot dapat dikelompokkan menjadi beberapa jenis sesuai dengan spesifikasinya, misalkan fungsi dan alat gerak. Jenis robot sesuai dengan alat geraknya adalah robot beroda dan robot berkaki. Robot berkaki dapat dikelompokkan sesuai jumlah kakinya, seperti robot *bipedal*, *hexapod*, *octopod*. Robot *bipedal* berkaki dua, salah satunya robot *humanoid*.

Robot *humanoid* adalah robot yang bentuknya berdasarkan struktur tubuh manusia, mampu melakukan interaksi dengan peralatan maupun lingkungan yang dibuat untuk manusia. Secara umum robot *humanoid* memiliki tubuh dengan kepala, sepasang lengan dan sepasang kaki, meskipun ada pula beberapa bentuk robot *humanoid* yang hanya berupa sebagian dari tubuh manusia, misalnya dari pinggang ke atas. Beberapa robot *humanoid* juga memiliki wajah, lengkap dengan mata dan mulut.

Robot *humanoid* digunakan sebagai sarana penelitian dalam berbagai bidang ilmu. Untuk membuat robot *humanoid*, peneliti perlu mempelajari struktur dan perilaku tubuh manusia (biomekanik). Simulasi pergerakan tubuh manusia mengarahkan pada pemahaman yang lebih baik mengenai hal tersebut.

Secara keseluruhan, sebuah robot terdiri dari sistem elektrik, mekanik, serta program. Ditinjau dari segi kontrol, sebuah robot *humanoid* biasanya merupakan sistem *loop* tertutup, yang terdiri dari sensor, pengontrol, dan aktuator. Sensor berfungsi seperti sistem penginderaan, kontroller sebagai otak, serta aktuator sebagai sistem otot dan persendian. Robot *humanoid* yang digunakan memiliki beberapa sendi yang disebut sebagai *Degree of Freedom* (DOF) yang

berupa aktuator, seperti motor, *pneumatic*, *hydraulic*, dll. Gambar 2.2 menunjukkan contoh pemodelan persendian atau *Degree of Freedom* (DOF) pada robot Aldebaran Nao



Gambar 2.2 Derajat Kebebasan Aldebaran Nao  
Sumber : Shahbazi (2012).

### 2.3 Exoskeleton Motion Capture System

*Motion capture* adalah proses perekaman gerakan dari manusia atau objek lainnya. *Motion capture device* bertugas mentransformasi pergerakan secara aktual dari subjek menjadi data posisi, orientasi, atau sudut persendian yang bisa digunakan. Aplikasi medis termasuk kawasan dari studi pergerakan, analisa gaya berjalan, dan estimasi gaya persendian. Industry *entertainment* telah menunjukkan minat yang tinggi dalam seluruh *body motion capture*, terutama karena menyediakan cara cepat dan mudah untuk mendapatkan gerakan manusia yang realistis untuk animasi komputer. (Kolosz, 1998: 1).

Animasi komputer tradisional dan pemrograman robot membutuhkan trajektori tersebut didesain untuk setiap derajat kebebasan individual. *Motion capture* memberikan pemrograman dari setiap derajat kebebasan secara simultan, sering kali secara *real time*. Pengontrolan *real time* dari komputer atau figur robot sangatlah sulit tanpa *motion capture*. Data *motion capture* bersifat natural dan terlihat manusiawi, sejak data diambil langsung dari tubuh dari pemain. *Motion*



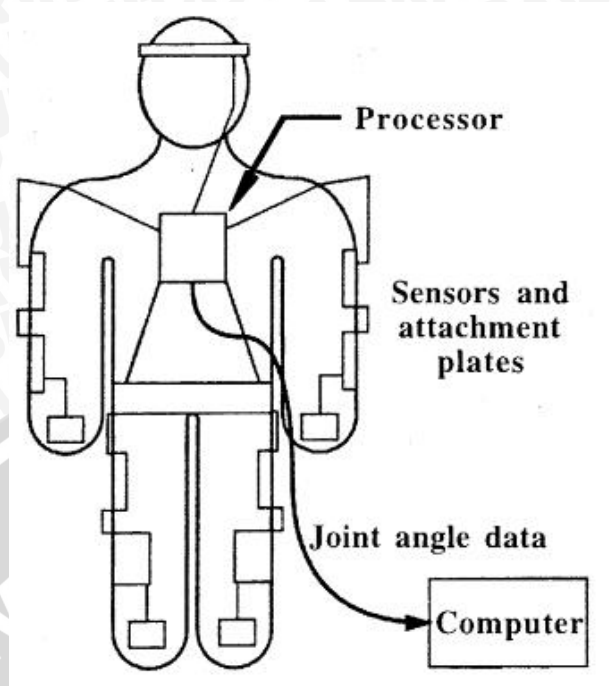
*capture* telah digunakan untuk memprogram dan animasi computer untuk kartun, iklan, dan film. (Kolosz, 1998: 1).

Sensor-sensor yang digunakan dalam *motion capture* bermacam macam, yaitu optikal, magnetik, inersial, serta elektromekanikal. Masing-masing jenis sensor menentukan tipe sistem yang digunakan dalam proses *motion capture*.

Sistem elektromekanikal sering disebut sebagai sistem *exoskeletal* atau *Exoskeleton Motion Capture System*. Biasanya sistem ini menggunakan sensor posisi putaran berupa potensiometer, *hall effect*, maupun *rotary encoder*. Alih-alih mengukur posisi sensor pada ruangan, sistem *exoskeletal* mengukur sudut persendian langsung dengan menempatkan sensor di sekitar sendi tubuh. *Exoskeleton* bisa berbentuk pakaian dimana setiap sensor dijahit, atau rangka *semi rigid* dengan pelat, sambungan, dan sensor-sensor dipasang mengikuti gerakan tubuh. Data yang ditangkap pada persendian disalurkan oleh perkabelan ke pemroses, dimana bisa dikenakan pada tubuh atau tidak. Sambungan kabel maupun nirkabel menghubungkan *exoskeleton* pada komputer. Sistem *exoskeletal* cenderung lebih simpel dari yang lainnya, semenjak sensor-sensor langsung mengukur nilai yang diinginkan, yaitu sudut persendian. Elektronika dan algoritma yang dipermudah ini dibutuhkan untuk *motion capture*. (Kolosz, 1998: 17).







Gambar 2.3 Exoskeleton motion capture system  
Sumber : Kolosz (1998:17).

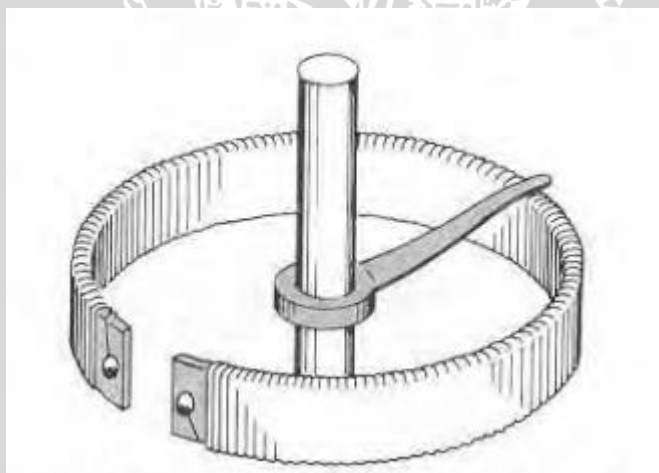
Gambar 2.3 menunjukkan rangkaian sistem *exoskeletal*. Jarak dari pemain hanya terbatas oleh panjang kabel atau sambungan nirkabel. Oleh karena itu, secara mendasar ruang gerak mereka terbatas. Permasalahan lingkungan seperti pencahayaan, bayangan, medan magnet, objek logam, dan lainnya tidak memberi dampak pada *exoskeleton*, karena sensor-sensor langsung mengukur persendian. Sistem ini juga sangat *portable* dan mudah didirikan, karena mereka hanya membutuhkan *exoskeleton* dan komputer. (Kolosz, 1998: 17-18).

Sensor untuk *exoskeleton* harus diletakkan pada persendian. Ini lebih membebani ketimbang sistem lain dan mungkin membatasi pergerakan pemain pada beberapa derajat. Dan pada akhirnya, *exoskeleton* relatif sulit untuk didesain, karena perbedaan yang cukup besar antara tinggi dan proporsi tubuh antar individu. Agar bisa digunakan secara umum, *exoskeleton* tidak bisa dibuat khusus untuk satu individu, melainkan harus pas untuk tipe dan ukuran tubuh yang berbeda. (Kolosz, 1998: 18).

## 2.4 Potensiometer

Potensiometer adalah sensor untuk mengukur perpindahan angular maupun linier, dimana disebut sebagai *angular potentiometer* atau *linear potentiometer*. Potensiometer telah berhasil diaplikasikan di banyak area keteknikan, seperti pengontrolan motor ultrasonik, instrumentasi medis dan *bioengineering*, control cerdas pada lengan robot, sistem *multi-fan hovering*, diagnose produk elektronika, dll. Pada saat ini, permintaan dari banyak kebutuhan keteknikan, potensiometer dikembangkan beserta integrasi, presisi tinggi, biaya rendah, dan pengendalian digital. (Zhang, 2005: 20)

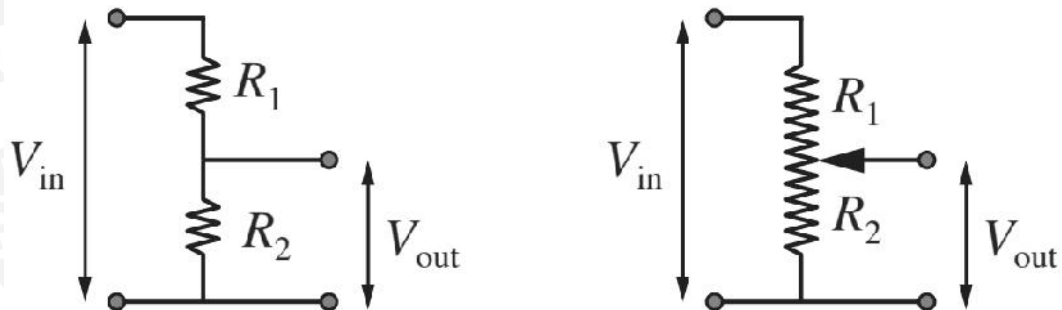
Potensiometer memiliki banyak jenis yang beredar luas di pasaran, seperti *rheostat*, *multitrim*, dll. Setiap penjual maupun pembeli kadang memiliki sebutan yang berbeda-beda untuk jenis potensiometer yang mereka maksud. Berdasarkan mekanismenya, potensiometer bisa dikelompokkan menjadi dua kelompok besar, yaitu potensiometer geser (*linear potentiometer*) dan potensiometer putar (*angular potentiometer*). Desain potensiometer putar ditunjukkan dalam Gambar 2.4.



Gambar 2.4 Potensiometer putar  
Sumber: Todd (1975: 11).

Potensiometer putar dapat mengubah posisi angular (melingkar) menjadi resistansi tertentu. Potensiometer putar mengubah besaran sudut menjadi resistansi atau sebaliknya. Biasanya potensiometer putar bila dipadukan bisa mengukur sudut lebih dari satu sumbu. Potensiometer putar biasanya digunakan

sebagai umpan balik posisi pada motor *servo*, *joystick*, tuas *analog* pada *gamepad*, maupun perangkat *motion capture*.

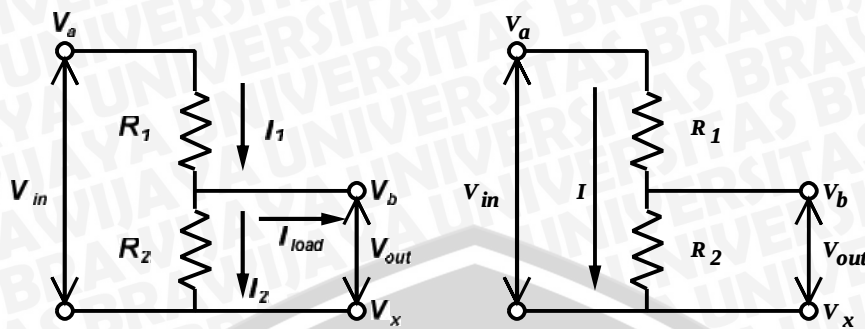


Gambar 2.5 Potensiometer sebagai pembagi tegangan  
Sumber: Recktenwald (2013:1).

Berdasarkan Gambar 2.5, potensiometer dapat digambarkan sebagai pembagi tegangan. Sebuah pembagi tegangan terdiri dari dua resistor seri terhadap sumber, dengan titik keluaran diantara kedua resistor. Di sebelah kiri gambar, tegangan masukan,  $V_{in}$ , adalah jatuh tegangan pada  $R_2$ .  $V_{out}$  kurang dari  $V_{in}$  karena tegangan total di sekitar  $R_1$  dan  $R_2$  dijumlahkan menjadi  $V_{in}$ . (Recktenwald, 2013: 1)

Sebuah potensiometer adalah pembagi tegangan yang memperbolehkan pengaturan  $V_{out}$ . Potensiometer biasanya memiliki penggeser atau *knob* yang berputar, yang menggerakkan kontak yang biasa disebut *wiper* sepanjang permukaan resistor. Seperti pada gambar, *wiper* membagi sebuah resistor tetap kedalam  $R_1$  dan  $R_2$ . Dengan menggeser *wiper* sepanjang resistor tetap, harga  $R_2$  berubah, yang memperbolehkan tegangan keluaran diatur dari 0 hingga  $V_{in}$ . (Recktenwald, 2013: 1)





Gambar 2.6 Analisis rangkaian potensiometer

Dari Gambar 2.6, dengan menggunakan hukum Kirchoff, diperoleh persamaan:

$$I_1 = I_2 + I_{load} \quad (2-1)$$

Ada aplikasi praktis dimana  $I_{load}$  sangat kecil, sehingga berdampak pada operasi pembagi tegangan yang diabaikan. Pada bagian kanan Gambar 2.8, menunjukkan arus yang mengalir dalam pembagi tegangan dengan resistansi tak hingga. (Recktenwald, 2013: 1-2)

Jika kita berasumsi bahwa arus beban diabaikan, persamaan (2-1) menunjukkan bahwa  $I_1 = I_2$ . Saat  $I_{load} = 0$ , semua arus yang dipasok oleh  $V_{in}$  mengalir ke *ground* sepanjang kombinasi seri  $R_1$  dan  $R_2$ . Sebut saja arus ini disebut  $I$ , dimana  $I = I_1 = I_2$ , seperti dalam Gambar 2.8 sebelah kanan. Tegangan harus memiliki dua titik, yaitu titik tegangan positif atau negatif, dan referensi. Beda tegangan untuk  $V_{out}$  adalah  $V_b$  dan referensi  $V_x$ , sementara untuk  $V_{in}$  adalah  $V_a$  dan referensi  $V_x$ .

Dengan menerapkan hukum Ohm, arus  $I$  adalah arus yang melewati  $R_{total}$  yang merupakan kombinasi seri  $R_1$  dan  $R_2$ , maka tegangan masukan  $V_{in}$  adalah:

$$V_{in} = I.R_{total}$$

$$V_a - V_x = I(R_1 + R_2) \quad (2-2)$$

Jika arus  $I$  yang digunakan, maka:

$$I = (V_a - V_x) / (R_1 + R_2) \quad (2-3)$$

Dengan menggunakan hukum Ohm pada  $R_2$ :

$$V_{out} = I.R_2$$

$$V_b - V_x = I.R_2 \quad (2-4)$$

Gabung persamaan (2-3) dan persamaan (2-4), untuk menghilangkan  $I$ :

$$V_{out} = V_{in} [R_2 / (R_1 + R_2)]$$

$$V_b - V_x = V_a - V_x [R_2 / (R_1 + R_2)]$$

$$V_b = V_a - V_x [R_2 / (R_1 + R_2)] + V_x \quad (2-5)$$

Pada kebanyakan aplikasi, tegangan referensi adalah 0 V, jadi persamaan (2-5) bisa disederhanakan menjadi:

$$V_b = V_a - 0 [R_2 / (R_1 + R_2)] + 0$$

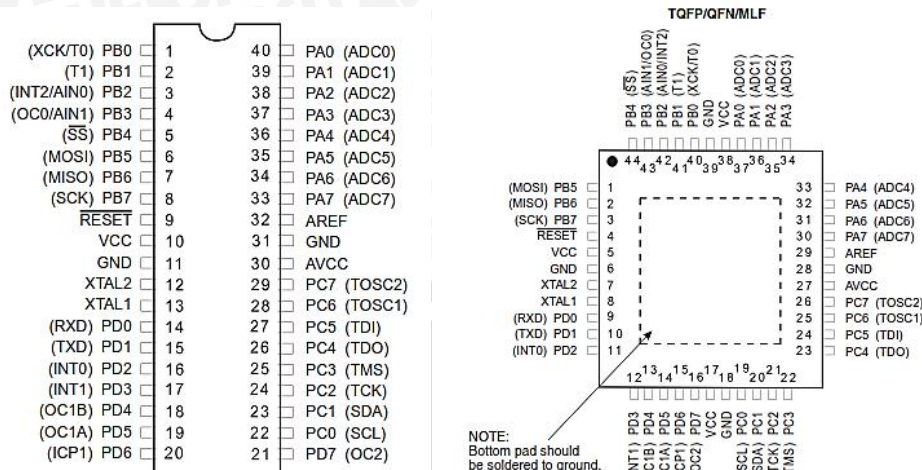
$$V_b = V_a \cdot [R_2 / (R_1 + R_2)] \quad (2-6)$$

## 2.5 **Microcontroller ATmega16**

*Microcontroller* dapat dianalogikan sebagai sebuah system computer yang dikemas dalam sebuah chip. Artinya bahwa di dalam sebuah IC *microcontroller* sebetulnya sudah terdapat kebutuhan minimal agar mikroprosesor dapat bekerja, yaitu meliputi mikroprosesor, ROM, RAM, I/O, dan clock seperti yang dimiliki oleh sebuah komputer PC. (Bejo, 2008: 1)

*Microcontroller* biasanya digunakan untuk sebuah sistem yang tidak terlalu rumit namun membutuhkan keandalan yang tinggi. Untuk sebuah sistem yang sesuai dengan kemampuannya, penggunaan *microcontroller* jauh lebih efisien ketimbang PC, diantaranya karena ukuran, konsumsi daya, *resource* dan kompleksitas yang lebih kecil. *Microcontroller* dapat diprogram dan dioperasikan secara langsung dengan memanfaatkan berbagai *peripheral* yang dimilikinya. Salah satu contoh *microcontroller*, yaitu ATmega16 yang merupakan keluarga AVR, ditunjukkan dalam Gambar 2.7.





Gambar 2.7 Microcontroller AVR ATmega16  
Sumber: Atmel (2010: 2).

AVR merupakan *microcontroller* arsitektur Harvard, cirinya kode program dan data disimpan dalam memori secara terpisah. Umumnya arsitektur Harvard ini menyimpan kode program dalam memori permanen atau semi permanen (*non-volatile*) sedangkan data disimpan dalam memori tidak permanen (*volatile*). (Bejo, 2008: 1)

ATmega16 adalah *microcontroller* CMOS 8-bit berdaya rendah berbasis AVR. Dengan mengeksekusi instruksi dalam satu siklus *clock*, ATmega16 mampu mencapai 1 MIPS per MHz, memungkinkan perancang sistem untuk mengoptimalkan konsumsi daya terhadap kecepatan pemrosesannya. (Atmel, 2010: 3)

### 2.5.1 ADC Microcontroller

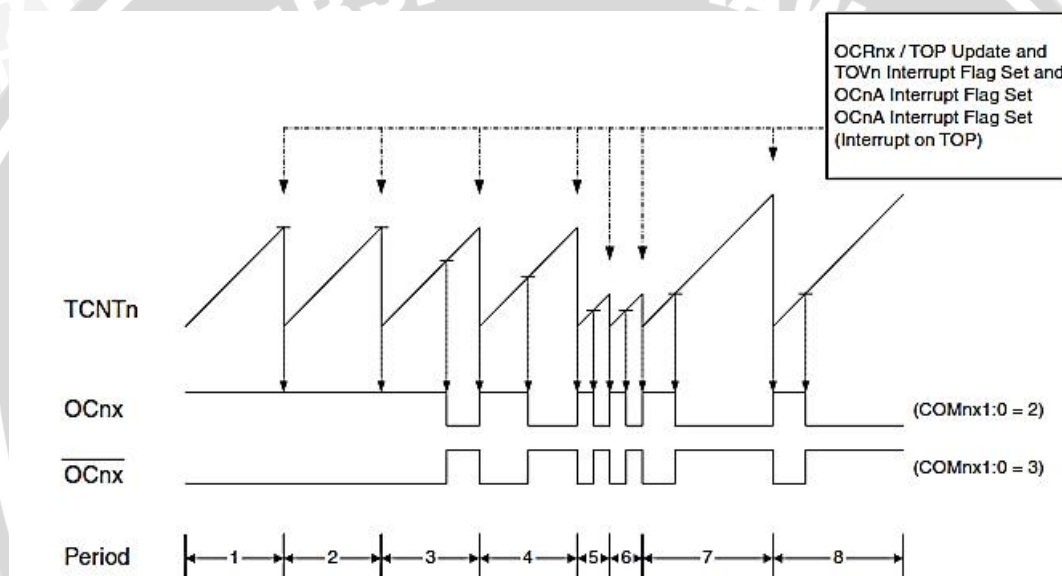
Mikrokontroler ATmega16 memiliki fasilitas *Analog to Digital Converter* yang sudah tertanam dalam *chip*. Sinyal dari ADC akan dipilih oleh *multiplexer* dikarenakan *converter* ADC dalam *chip* hanya satu buah sedangkan salurannya ada delapan maka dibutuhkan *multiplexer* untuk memilih ADC secara bergantian. Operasi ADC membutuhkan tegangan referensi VREF. ADC mengonversi tegangan analog menjadi bilangan digital selebar 10-bit atau 8-bit. GND (0 volt) adalah nilai minimum yang mewakili ADC dan nilai maksimum ADC diwakili oleh tegangan pada AREF minus 1 LSB. Sinyal ADC tidak boleh melebihi tegangan referensi. Data hasil konversi ADC untuk resolusi 10-bit (1024) dirumuskan dalam Persamaan (2-7). (Ardi Winoto, 2010:158).



$$ADC = \frac{V_{in}}{V_{ref}} \times 1024 \quad (2-7)$$

### 2.5.2 Pulse Width Modulation (PWM)

Sinyal PWM dapat dibangkitkan dengan menggunakan *peripheral Timer/Counter* yang terdapat pada *Microcontroller*. Pada *microcontroller* ATmega16 terdapat dua buah *Timer/Counter* 8-bit dan satu buah *Timer/Counter* 16-bit. *Timer/Counter* 1 merupakan *Timer/Counter* 16-bit. Salah satu mode yang bisa digunakan untuk membangkitkan sinyal PWM adalah mode *fast PWM*, seperti yang terdapat dalam Gambar 2.8.



Gambar 2.8 Fast PWM mode  
Sumber: Atmel (2010:108).

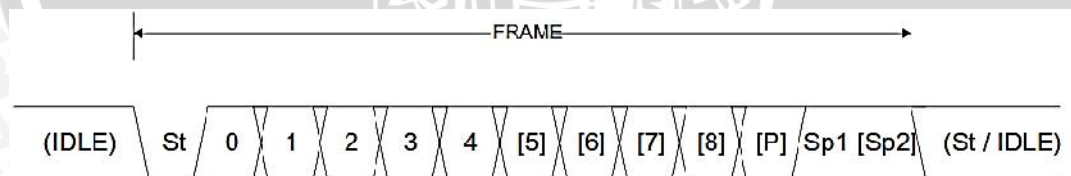
Mode *fast Pulse Width Modulation* atau *fast PWM* menyediakan pilihan pembangkitan gelombang PWM frekuensi tinggi. *Fast PWM* berbeda dengan opsi PWM lainnya oleh operasi *single-slope*. *Counter* mencacah dari BOTTOM hingga TOP, kemudian dimulai lagi dari BOTTOM. Dalam mode *non-inverting Compare Output*, *Output Compare* (OC1x) dinolkan pada *compare match* antara TCNT1 dan OCR1x, dan masuk kondisi *set* pada BOTTOM. Dalam mode *inverting Compare Output*, keluaran berada pada kondisi *set* pada *compare match*, dan dinolkan pada BOTTOM. Berdasarkan operasi *single-slope*, frekuensi yang beroperasi pada mode *fast PWM* bisa dua kali lebih tinggi dari mode *phase correct*

PWM yang menggunakan operasi *dual-slope*. Setiap pengiriman data pada UART menggunakan bit tanda *start* bit dan *stop* bit. `

### 2.5.3 Komunikasi Serial

Pada prinsipnya, komunikasi serial ialah komunikasi dengan pengiriman data yang dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi paralel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali waktu. Devais pada komunikasi serial port dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Pengiriman data secara serial dapat berupa sinkron, yaitu pengiriman clock dilakukan bersamaan dengan data, atau berupa asinkron, yaitu pengiriman clock tidak bersamaan dengan data namun secara dua tahap, saat data dikirim dan data diterima. Untuk istilah yang sering digunakan untuk mengirim dan menerima data secara asinkron biasa disebut *Universal Asynchronous Receiver Transmitter* (UART). Komunikasi data serial menggunakan UART sangat umum dan mudah penggunaannya, misalnya pada port serial PC. Pada UART jalur pengiriman dan penerimaan data serial dipisahkan.

Setiap pengiriman data pada UART menggunakan bit tanda *start* bit dan *stop* bit. Jalur data yang digunakan hanya satu untuk setiap pengiriman data. Data-data serial dikirim melewati jalur data satu persatu setiap satuan waktu. Format pengiriman data serial secara asinkron ditunjukkan dalam Gambar 2.9.



Gambar 2.9 Format frame data serial UART  
Sumber: Atmel (2010:148).

Dengan :

- St = *Bit start* selalu berlogika rendah
- (n) = Banyaknya data yang dikirim (0-8)
- P = *Bit paritas* (ganjil atau genap)
- Sp = *Bit stop* selalu berlogika tinggi (*bit stop* bisa berjumlah 1 atau 2

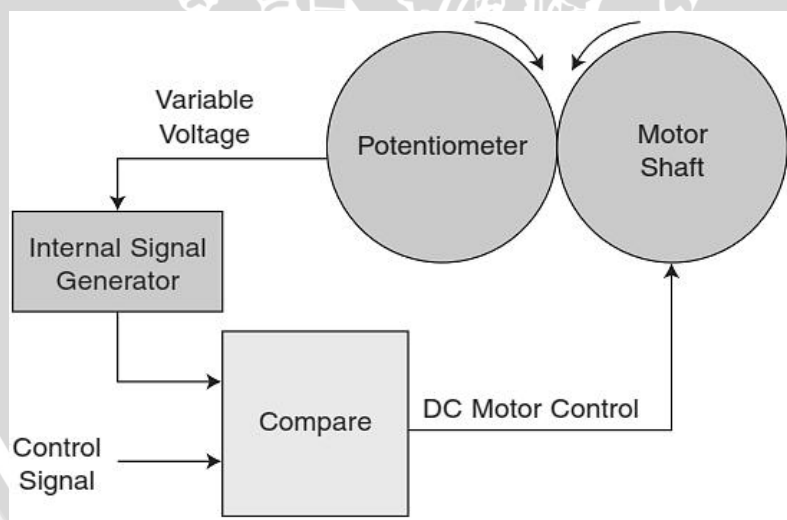


IDLE = Tidak ada data yang ditransfer pada RX dan TX, IDLE selalu berlogika tinggi

## 2.6 Motor DC Servo

*Servo* digunakan untuk mengatur posisi angular motor secara presisi, seringkali terbatas putaran maksimumnya pada 180°. *Servo* umum digunakan pada mainan mobil radio kontrol untuk mengatur stir, pesawat radio control untuk mengatur *rudder*, bahkan system *cruise control* pada mobil. *Servo* ideal untuk aplikasi yang membutuhkan penempatan posisi absolut untuk poros motor. (Pinckney, 2006: 27)

*Servo* dapat berputar searah (CW) maupun berlawanan arah jarum jam (CCW). Selain *standard servo* yang hanya dapat berputar maksimal 180°, *servo* juga ada yang berjenis *continuous rotation*, yang dapat berputar hingga lebih dari 360°. Motor DC servo akan bergerak sampai posisi porosnya sesuai dengan posisi yang diinginkan. Setelah posisi yang diinginkan tercapai, motor akan berhenti berputar dan mempertahankan posisi terakhirnya, atau biasa disebut keadaan *stall*.



Gambar 2.10 Diagram motor DC Servo  
Sumber: Pinckney (2006: 27).

Berdasarkan Gambar 2.10, motor DC servo disusun oleh motor DC yang dihubungkan dengan poros, potensiometer, serta pengendali berupa generator sinyal dan komparator. Motor DC dikopel dengan *gear* internal untuk memperbesar torsi keluaran motor. Setiap *gear* dikopel dengan perbandingan tertentu, untuk menghasilkan kecepatan angular yang lebih kecil namun dengan



torsi yang lebih besar. Biasanya torsi yang besar dibutuhkan untuk menggerakkan poros/lengan dengan massa dan jari-jari yang besar. Selain itu *servo* digunakan untuk rotasi dengan sudut yang relatif kecil, sehingga tidak membutuhkan kecepatan angular yang terlalu tinggi. Bagian internal motor DC *servo* ditunjukkan dalam Gambar 2.11

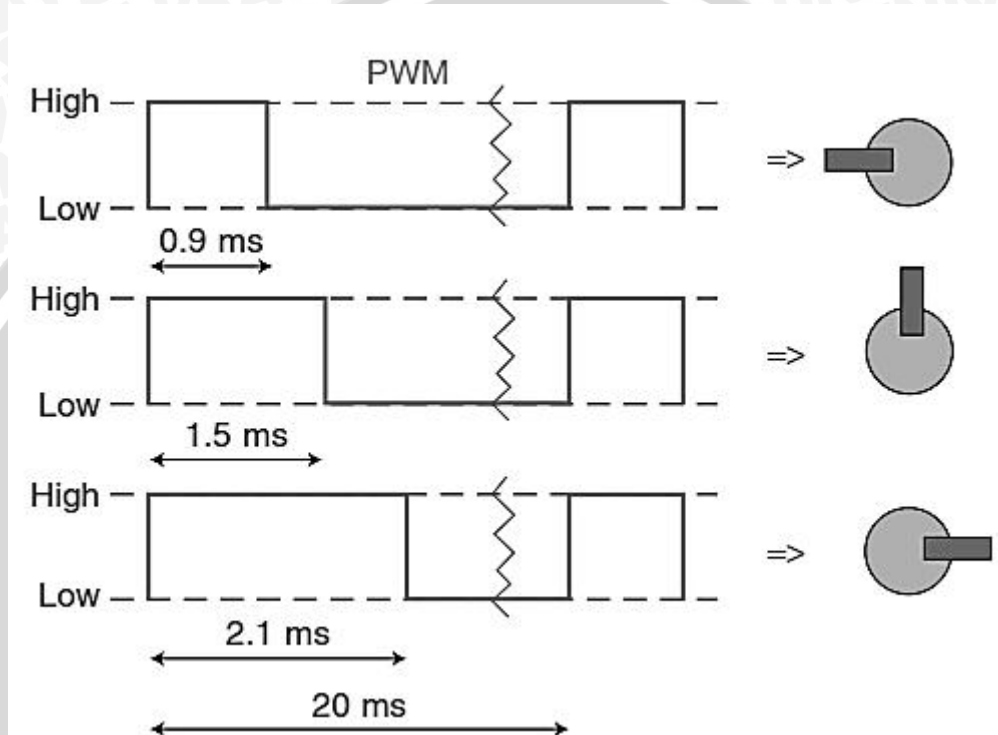


Gambar 2.11 Bagian internal motor DC *servo*  
Sumber: Sawicz (2002:9)

Potensiometer atau sensor rotasi lain digunakan untuk umpan balik posisi sudut motor. Potensiometer dipasang langsung dengan poros keluaran *servo* agar posisi *servo* yang sebenarnya yang terbaca. Selain potensiometer, beberapa motor DC *servo* menggunakan sensor rotasi lain seperti *optical* maupun *magnetic rotary encoder*.

Rangkaian kontroler internal digunakan untuk memproses masukan sinyal dari kontroler eksternal menjadi sudut *servo* yang diinginkan. Rangkaian kontroler ini terbagi menjadi dua jenis, yaitu kontroler analog dan digital. Jenis kontrolernya menjadi patokan dua jenis *servo* yang beredar di pasaran, yaitu *servo* analog dan *servo* digital. *Servo* digital biasanya memiliki tingkat kepresisian yang lebih tinggi daripada *servo* analog. Saat ini, *servo* digital biasanya menggunakan *microcontroller* sebagai kontroler internalnya.

Motor DC servo dihubungkan dengan sepasang jalur *supply* tegangan DC dan jalur data. Jalur data digunakan untuk menyalurkan sinyal kontrol dari pengontrol eksternal. Sinyal kontrol yang digunakan bisa menggunakan PWM maupun komunikasi serial. Untuk *servo* dengan sinyal control PWM, jalur data yang digunakan hanya satu. Sinyal PWM yang dibutuhkan untuk mengendalikan motor DC *servo* ditunjukkan dalam Gambar 2.12.

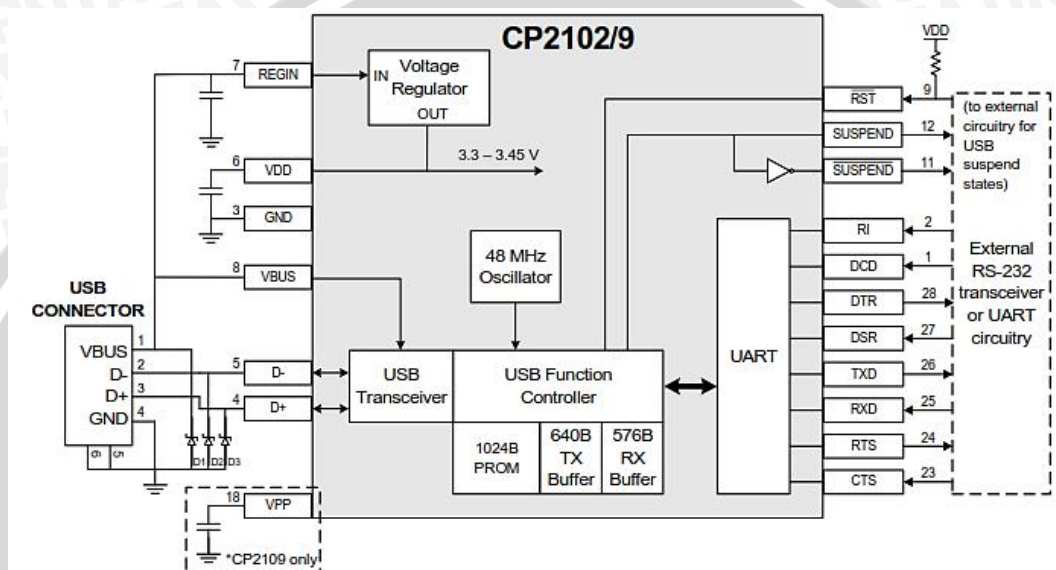


Gambar 2.12 Sinyal kontrol motor DC *servo*  
Sumber: Pinckney (2006:29).

Menurut gambar di atas, motor DC *servo* membutuhkan sinyal kotak dengan periode sebesar 20ms. Setiap satu siklus sinyal diawali logika tinggi dengan *duty cycle* atau periode siklus positif ( $T_{ON}$ ) dengan lebar tertentu. Lebar  $T_{ON}$  inilah yang biasanya mewakili posisi sudut tertentu. Biasanya motor DC *servo* di pasaran memiliki spesifikasi  $T_{ON}$  maksimal dan minimal yang berbeda-beda sesuai tipenya. Biasanya lebar  $T_{ON}$  untuk posisi  $90^\circ$  diwakili oleh periode 1,5ms. Beberapa motor DC *servo* digital yang beredar di pasaran dapat diprogram batas sudutnya terhadap sinyal PWM yang diberikan. Sedangkan untuk *servo* yang lain harus dilakukan karakterisasi untuk mengetahui perilaku *servo* tersebut.

## 2.7 USB to TTL

Untuk melakukan komunikasi data dari PC dengan sistem *microcontroller*, tidak bisa dilakukan sembarangan. PC yang pada umumnya menggunakan antarmuka USB (dengan level tegangan sinyal differensial, D+ dan D-), harus dihubungkan dengan *microcontroller* (dengan level tegangan TTL UART) melalui sebuah *converter*. Di pasaran beredar modul konverter USB ke TTL dengan rangkaian yang bervariasi. Salah satunya rangkaian berbasis CP2102.



Gambar 2.13 Rangkaian USB to TTL CP2102  
Sumber: Silicon Labs (2013: 1).

Gambar 6.16. menunjukkan rangkaian CP2102. Data dari PC dihubungkan melalui port USB ke pin D+ dan D-, sedangkan data dari mikrokontroler dihubungkan ke pin TXD serta RXD. Tegangan kerja yang digunakan dari rangkaian ini ke mikrokontroler bisa menggunakan tegangan 3V maupun 5V, dengan satu keluaran tegangan referensi (GND). Untuk membaca data dari mikrokontroler ke PC, bisa menggunakan fasilitas *Serial Terminal*.



## BAB III

### METODE PENELITIAN

#### 3.1 Studi Literatur

Studi literatur dilakukan untuk mempelajari teori penunjang sistem yang dibutuhkan dalam perencanaan dan pembuatan alat. Teori yang diperlukan antara lain berkaitan dengan robot *humanoid*, Kontes Robot Seni Indonesia, *exoskeleton motion capture system*, potensiometer, *microcontroller* ATmega 16, motor DC *servo*, serta USB to TTL converter.

#### 3.2 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

- 1) *Exoskeleton* berbahan dasar Aluminium tipe *Hollow*, dengan sambungan/sendi dari aluminium lembaran .
- 2) Sensor pada setiap sendi menggunakan potensiometer mono B50K $\Omega$  *single turn linear taper*.
- 3) *Exoskeleton* hanya dibuat untuk bagian *upper body*.
- 4) Robot memiliki rangka berbahan dasar *Acrylic* dan Aluminium.
- 5) *Exoskeleton* dan robot memiliki sendi sebanyak 8 DOF pada bagian *upper body*.
- 6) Sumber tegangan berupa baterai *Lithium Polymer* 2S 7,4V 2200mAH.
- 7) Motor *servo* yang digunakan bertipe Hitec HS-5685 dan Corona DS-238-HV.
- 8) Rangkaian catu daya 5V berupa rangkaian regulator *switching* LM2576.
- 9) Rangkaian *microcontroller* menggunakan ATmega16 SMD.
- 10) Modul USB to TTL *converter* yang digunakan bertipe CP2102.
- 11) Tombol *capture* yang digunakan merupakan saklar *push-on*.
- 12) Fasilitas *Serial Terminal* menggunakan program CVAVR.

### 3.3 Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat dalam penelitian ini dibagi menjadi perancangan *hardware* dan *software*.

#### 3.3.1 Perancangan dan Pembuatan Mekanik

Perancangan dan pembuatan perangkat keras terdiri atas dua bagian yakni perancangan mekanik dan perancangan elektronik. Perancangan mekanik diperlukan untuk membuat desain *exoskeleton* dan robot *humanoid*. Bentuk mekanik secara umum dirancang dengan menggunakan perangkat lunak 3Ds MAX 2011.

#### 3.3.2 Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

Perancangan rangkaian elektronik ditekankan pada perancangan rangkaian antarmuka potensiometer, *mainboard microcontroller*, serta ATmega16 SMD. Papan rangkaian tercetak (PCB) dirancang dengan menggunakan perangkat lunak EAGLE (*Easily Applicable Graphical Layout Editor*) versi 6.4.0. Sementara untuk rangkaian USB to TTL, menggunakan modul yang sudah jadi.

#### 3.3.3 Perancangan dan Pembuatan Perangkat Lunak (*Software*)

Setelah melalui proses perancangan dan pembuatan perangkat keras, selanjutnya akan dilakukan perancangan dan pembuatan perangkat lunak. Perangkat lunak berfungsi untuk memberikan instruksi kerja kepada perangkat keras tersebut.

Perangkat lunak dirancang melalui pembuatan diagram alir (*flowchart*) sub sistem hingga sistem secara keseluruhan. Kemudian dilakukan penulisan program menggunakan *compiler* bahasa pemrograman C CVAVR (*Code Vision AVR C Compiler*) versi 2.05.0.

### 3.4 Pengujian Alat

Untuk menganalisis kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian sistem. Pengujian dilakukan pada masing-masing blok pada perancangan *hardware* serta pengujian keseluruhan untuk mengetahui *software* dapat berjalan atau tidak.



### 3.4.1 Pengujian Sub Sistem

Pada bagian ini pengujian dilakukan pada masing-masing blok. Pengujian ini bertujuan untuk mengetahui apakah masing-masing blok dapat bekerja sesuai dengan fungsinya seperti yang telah direncanakan. Pengujian tersebut meliputi :

#### 1. Pengujian Potensiometer

Pengujian potensiometer dilakukan untuk menguji tegangan keluaran terhadap posisi sudut potensiometer.

#### 2. Pengujian Serial Terminal

Pengujian dilakukan untuk melihat kinerja komunikasi *microcontroller* ke PC.

#### 3. Pengujian ADC Microcontroller

Pengujian ini dilakukan untuk menguji nilai digital dari hasil konversi tegangan analog.

#### 4. Pengujian Sinyal PWM

Pengujian ini dilakukan untuk menguji keluaran lebar sinyal *high PWM* dari masukan data pada *microcontroller*.

#### 5. Pengujian Motor DC Servo

Pengujian dilakukan untuk mengetahui keluaran sudut motor DC servo dari sinyal PWM yang diberikan.

### 3.4.2 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini dengan menyambungkan semua *hardware* yang dibuat berdasarkan blok diagram dan memasukkan program berupa *software* yang bekerja untuk mengendalikan *hardware* yang telah dibuat. Sistem bekerja dengan baik jika dapat berjalan sesuai *flowchart* yang telah direncanakan.

## 3.5 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian secara keseluruhan.

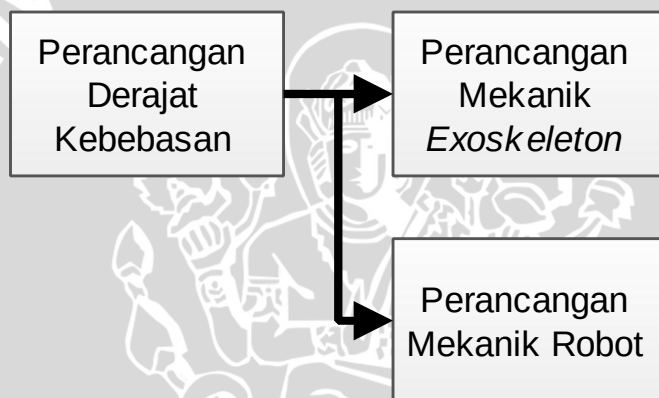


## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

#### 4.1 Perancangan Mekanik

Mekanik alat yang dirancang meliputi *exoskeleton* dan robot *humanoid*. Mekanik yang dirancang tidak hanya sebagai sistem yang menopang rangkaian elektronik, tetapi juga sebagai penghubung antara pengguna *exoskeleton* dan sistem itu sendiri. Sistem mekanik kedua alat juga merupakan sistem rangka yang mengacu pada rangka manusia. Sistem mekanik yang cocok diperlukan untuk membuat *exoskeleton* dan robot *humanoid*, agar robot dapat bergerak mengikuti gerakan pengguna *exoskeleton*.



Gambar 4.1 Alur perancangan mekanik sistem

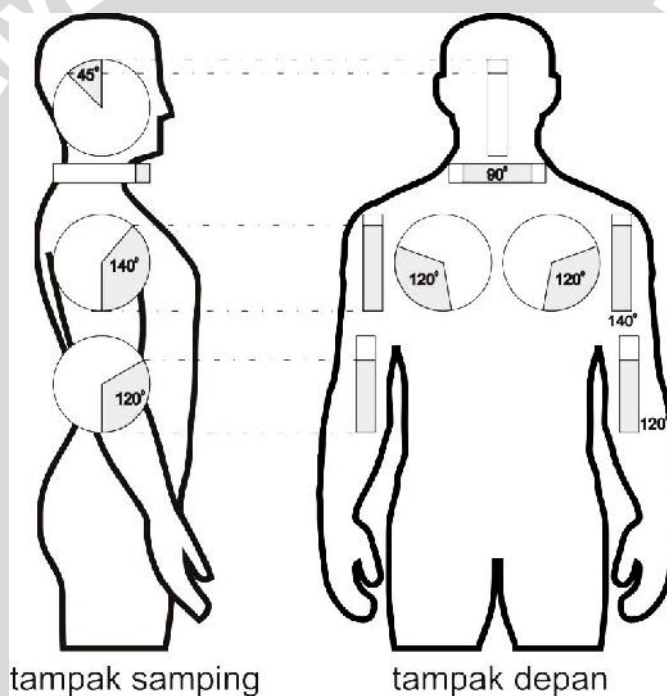
Berdasarkan Gambar 4.1, perancangan mekanik dimulai dengan mendesain derajat kebebasan. Setiap bagian yang bergerak pada sistem memiliki pergerakan rotasi, jadi inti dari mekanik kedua alat adalah sendi, atau derajat kebebasannya (DOF). Setiap DOF harus diatur jumlah, posisi, dan keluwesannya. *Exoskeleton* dan robot *humanoid* dirancang dengan struktur yang identik, menyesuaikan sistem derajat kebebasannya. Setiap DOF pada robot merepresentasikan derajat kebebasan pada *exoskeleton*.

##### 4.1.1 Perancangan Derajat Kebebasan Sistem

Derajat kebebasan yang dirancang mengikuti anatomi pergerakan tubuh manusia. Setiap sendi yang terdapat pada bagian *upper body* diidentifikasi, kemudian dikelompokkan menjadi sebuah DOF yang diinginkan. Sendi yang dirancang harus bersumbu tunggal, mengingat sensor dan aktuator yang tersedia

hanya memiliki satu sumbu. Jadi, sendi di tubuh yang memiliki lebih dari satu sumbu (seperti sendi peluru pada ketiak) harus dipecah menjadi masing-masing hanya satu sumbu. Sendi peluru memiliki tiga sumbu, membuat tipe sendi ini dapat berputar ke segala arah. Jadi, derajat kebebasan alat untuk sendi peluru dipecah menjadi tiga sendi engsel yang hanya memiliki satu sumbu. Satu sendi engsel terdiri dari bagian statis yang terhubung langsung pada ruas rangka sebelumnya, serta bagian dinamis yang menggerakkan ruas rangka setelahnya.

Untuk mempermudah pembuatan mekanik alat dan keteresuaian dengan perangkat keras, tidak semua artikulasi sendi manusia diterapkan pada sistem. Derajat kebebasan yang digunakan hanya yang mendukung peletakan sensor pada bagian yang seporos dengan satu derajat kebebasan.



Gambar 4.2 Desain derajat kebebasan sistem

Rancangan derajat kebebasan ditunjukkan dalam Gambar 4.2. Total derajat kebebasan masing-masing alat adalah 8 DOF. Setiap sendi terdiri dari 3 DOF lengan kanan, 3 DOF lengan kiri, dan 2 DOF kepala. Setiap derajat kebebasan sendi dirancang dengan mengukur kemampuan rotasi sendi tersebut.

3 DOF sendi lengan terdiri dari sendi siku (lengan bawah ke depan), sendi ketiak ke depan (lengan atas ke depan), dan sendi ketiak menyamping (lengan atas ke samping). Satu bagian lengan kanan merupakan pencerminan dari lengan kiri.

Sendi siku sejatinya merupakan sendi engsel yang menjadi dari bagian lengan paling ujung yang bergerak. Sendi siku memiliki bagian statis pada lengan atas dan menggerakkan bagian dinamis pada lengan bawah. Sudut maksimal sendi ini adalah  $120^\circ$ .

Bagian sendi pada ketiak pada dasarnya merupakan jenis sendi peluru dengan tiga sumbu. Pada alat ini, sumbu yang di ambil hanya dua, maka bagian ketiak terbagi menjadi sendi ketiak ke depan dan sendi ketiak menyamping.

Bagian sendi ketiak menyamping menghubungkan bagian statis pada bahu dan menggerakkan bagian dinamis pada ruas penghubung ke sendi ketiak ke depan. Sudut maksimal sendi ini adalah  $120^\circ$ .

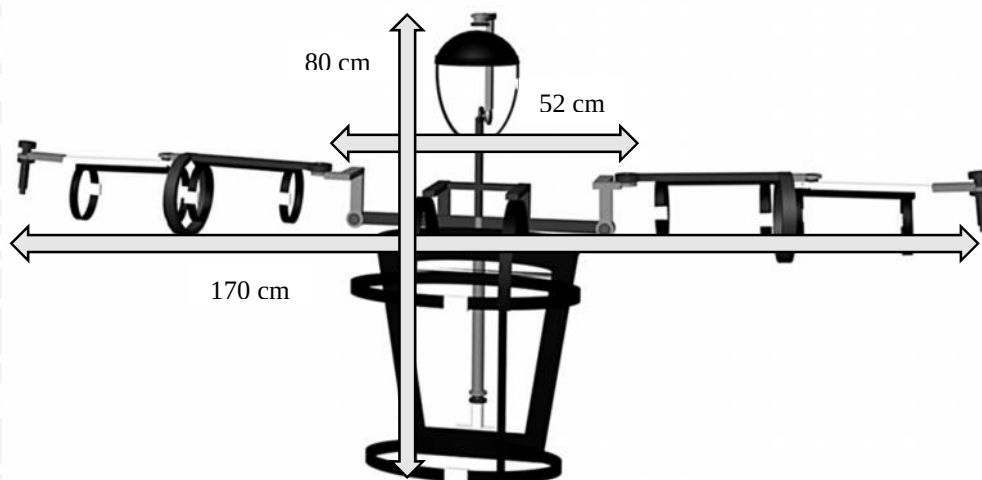
Bagian sendi ketiak ke depan menghubungkan bagian statis pada ruas penghubung dari sendi ketiak menyamping dan menggerakkan ruas lengan atas. Sudut maksimal sendi ini adalah  $140^\circ$ .

2 DOF kepala terdiri dari  $90^\circ$  sendi geleng ke samping dan  $45^\circ$  sendi angguk ke belakang. Khusus pada bagian kepala, kedua sendi tidak identik untuk sistem *exoskeleton* dan robot *humanoid*. Bagian statis dan dinamis keduanya juga mengikuti mekanik masing-masing sub-sistem.

#### **4.1.2 Perancangan Mekanik *Exoskeleton***

Perancangan mekanik *exoskeleton* dibuat berdasarkan struktur tubuh penggunanya. Pada perancangan ini, struktur tubuh yang jadi patokan adalah struktur tubuh penulis. Jadi, ukuran-ukuran setiap ruas *exoskeleton* dibuat berdasarkan tubuh penulis. *Exoskeleton* akan dipasang potensiometer dan tombol *capture*. Desain 3D Mekanik *exoskeleton* ditunjukkan dalam Gambar 4.3.





Gambar 4.3 Desain mekanik *exoskeleton*

*Exoskeleton* dibuat dari rangka berbahan dasar Aluminium Hollow 1,2 x 2,5 cm. Penyambungan rangka statis menggunakan baut 3 mm, mur 3 mm, dan rivet 3,5 mm. Dimensi keseluruhan *exoskeleton* adalah tinggi sekitar 80 cm dan lebar bahu 52 cm, dan rentang tangan sekitar 170 cm. Dimensi *exoskeleton* tidak harus memenuhi patokan ini, tetapi disesuaikan dengan ukuran tubuh pengguna.

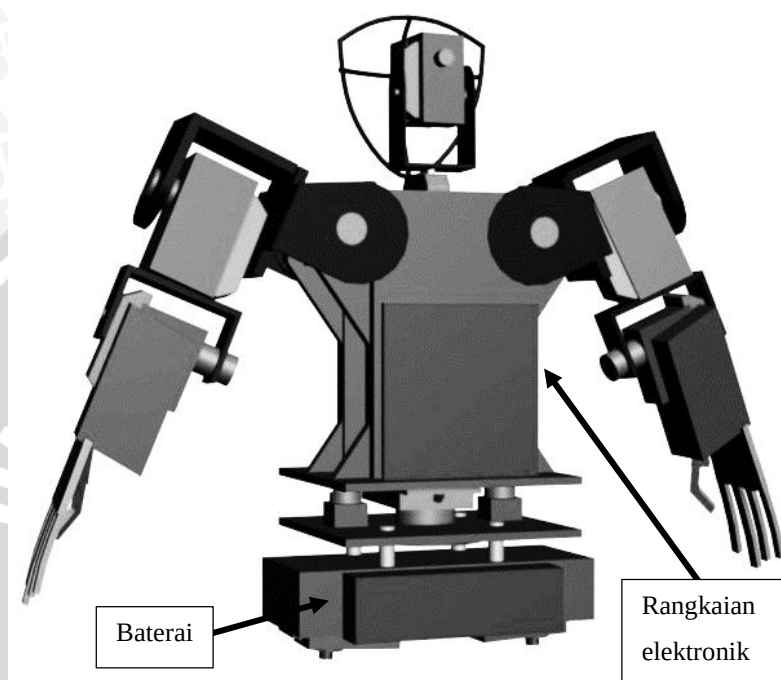
Sama halnya seperti pakaian, *exoskeleton* bisa digunakan untuk orang dengan tubuh yang lebih kecil dari standar ukuran yang ditetapkan, asalkan memiliki struktur rangka yang sesuai. Akan tetapi, apabila struktur rangka dari pengguna relatif kecil atau relatif besar terhadap struktur *exoskeleton*, pengguna mungkin akan kesulitan menyesuaikan pergerakan tubuh dengan *eksoskeleton*.

Setiap potensiometer diletakkan seporos dengan sendi tubuh, menghubungkan bagian statis dan bagian dinamis dari *exoskeleton*. Untuk meletakkan potensiometer agar *wiper* tidak patah, maka bagian tersebut diberi selongsong besi silinder pejal, kemudian dijepit oleh dua buah pelat sejajar untuk menyangganya. Untuk menyesuaikan sudut acuan setiap sendi, potensiometer bisa diputar dan dikencangkan kembali.

Khusus untuk *exoskeleton*, sendi geleng ke samping tidak diletakkan di leher, melainkan di atas kepala, untuk mempermudah pengukuran. Di bagian ujung lengan terdapat satu buah tombol *capture* yang berupa saklar *push-on*. Potensiometer dan tombol *capture* dihubungkan ke robot *humanoid* melalui kabel sepanjang 2 meter.

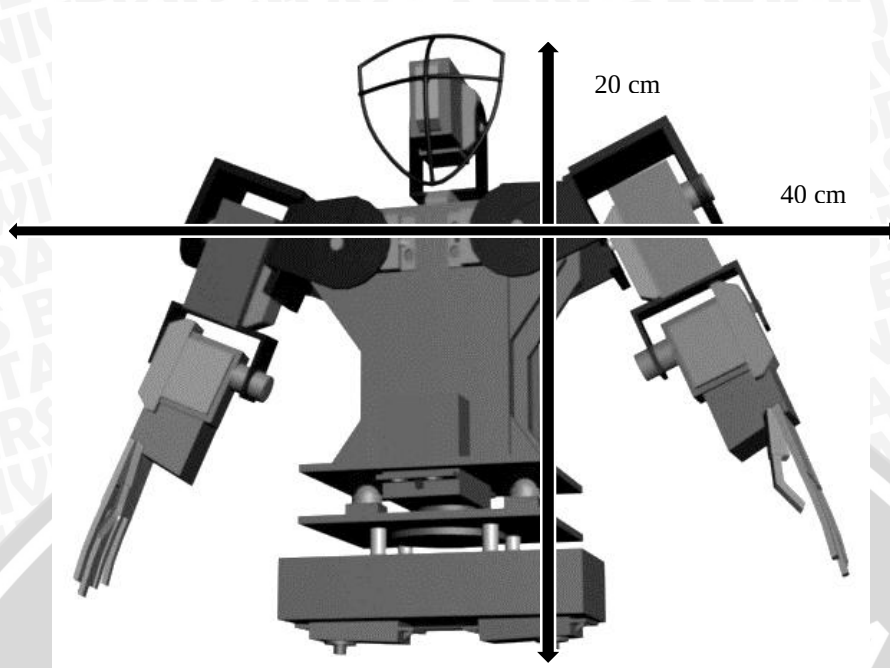
#### 4.1.3 Perancangan Mekanik Robot *Humanoid*

Bagian tubuh yang digerakkan hanya bagian *upper body*. Bagian *lower body* hanya difungsikan sebagai penyangga tubuh robot. Desain bagian belakang mekanik robot *humanoid* ditunjukkan dalam gambar 4.4.



Gambar 4.4 Desain bagian belakang robot *humanoid*

Robot *humanoid* akan dipasang motor DC *servo* serta rangkaian elektronik. Robot dirancang menyesuaikan *exoskeleton*, hanya saja sendi pada bagian geleng ke samping diletakkan pada leher. Rangkaian elektronik sistem diletakkan di bagian belakang robot. Robot *humanoid* menggunakan baterai *Lithium Polymer 2S 7,4V 2200mAh* untuk mencatu keseluruhan system. Baterai diletakkan di bagian pinggang robot.



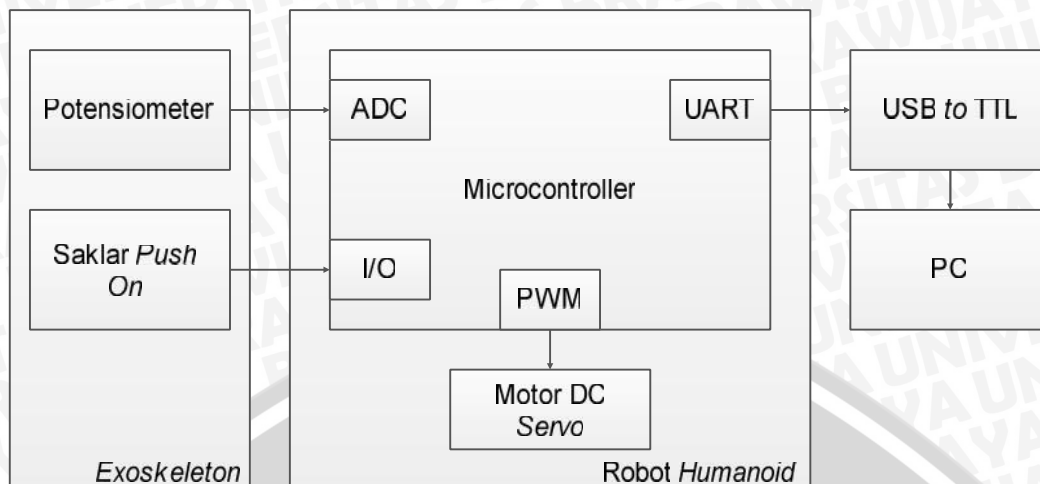
Gambar 4.5 Desain bagian depan robot *humanoid*.

Gambar 4.5 menunjukkan desain 3D robot dari depan beserta dimensinya. Robot memiliki dimensi tinggi 20 cm dan rentang tangan maksimal 40 cm. Dimensi ini dibuat berdasarkan robot *humanoid* yang digunakan pada perlombaan KRSI 2014. Aktuator yang digunakan pada robot disesuaikan dengan jumlah derajat kebebasan, yaitu 8 DOF motor DC *servo*. Untuk 4 DOF bagian ketiak, *servo* yang digunakan adalah Hitec HS-5685. Sementara untuk 4 DOF lainnya menggunakan servo Corona DS-238-HV. Kedua jenis *servo* ini adalah *servo* digital yang bisa langsung dicatu oleh baterai *Lithium Polymer* 2S 7,4V.

#### 4.2 Perancangan *Hardware*

Secara garis besar, diagram blok perancangan perangkat keras (*hardware*) sistem secara keseluruhan ditunjukkan dalam Gambar 4.6.





Gambar 4.6 Diagram blok *hardware* sistem secara keseluruhan

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

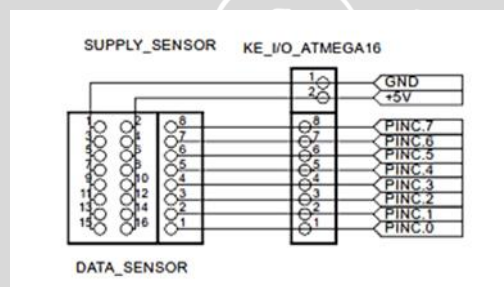
- 1) *Potensiometer* berfungsi sebagai masukan yang mengubah sudut pada setiap sendi tubuh menjadi tegangan analog.
- 2) *Microcontroller* ATmega16 berfungsi sebagai pengolah utama, dengan *peripheral* berupa *Analog to Digital Converter* (ADC) untuk mengubah tegangan analog menjadi data digital, *Timer Pulse Width Modulation* (PWM) sebagai pembangkit sinyal kontrol motor DC servo, dan *Universal Asynchronous Receiver/Transmitter* (UART) sebagai jalur komunikasi ke USB to TTL.
- 3) Motor DC servo sebagai aktuator yang menggerakkan setiap sendi robot humanoid.
- 4) Tombol *capture* yang berupa saklar *push-on* sebagai pemicu transfer data dari *microcontroller* ke PC.
- 5) *Personal Computer* untuk melihat data gerakan dalam bentuk sudut.

Untuk cara kerja alat sendiri, pertama sensor potensiometer yang terpasang pada setiap sendi pada *exoskeleton* akan membaca sudut pergerakan manusia menjadi tegangan analog. Data analog dari potensiometer akan dimasukkan ke *microcontroller* melalui fasilitas ADC, untuk mengubah data analog menjadi digital. Data digital akan diproses oleh *microcontroller* menjadi data sudut, yang akan dikonversi menjadi nilai tertentu untuk mengatur lebar sinyal *high* PWM. Sinyal PWM yang dibangkitkan akan digunakan untuk

menggerakkan motor DC servo yang merupakan aktuator robot *humanoid*. Robot akan bergerak mengikuti gerakan manusia yang mengenakan *exoskeleton*. Apabila gerakan robot sudah untuk satu *sequence* sudah bagus, sudut dari gerakan tersebut dapat ditampilkan ke PC dengan menekan tombol *capture*. Data dari *microcontroller* akan disalurkan ke PC setelah dikonversi dari antarmuka UART ke USB oleh modul USB to TTL. Serangkaian data sudut akan ditampilkan di PC oleh program *serial terminal*.

#### 4.2.1 Perancangan Rangkaian Antarmuka Potensiometer

Sensor potensiometer memiliki tiga kaki, bisa dikategorikan sebagai pin referensi (GND), tegangan suplai positif, serta data. Rangkaian ini dirancang sebagai modul yang bisa langsung ditancapkan ke rangkaian antarmuka *microcontroller*. Gambar 4.7 menunjukkan rangkaian antarmuka sensor.



Gambar 4.7 Skematik Rangkaian Antarmuka Potensiometer

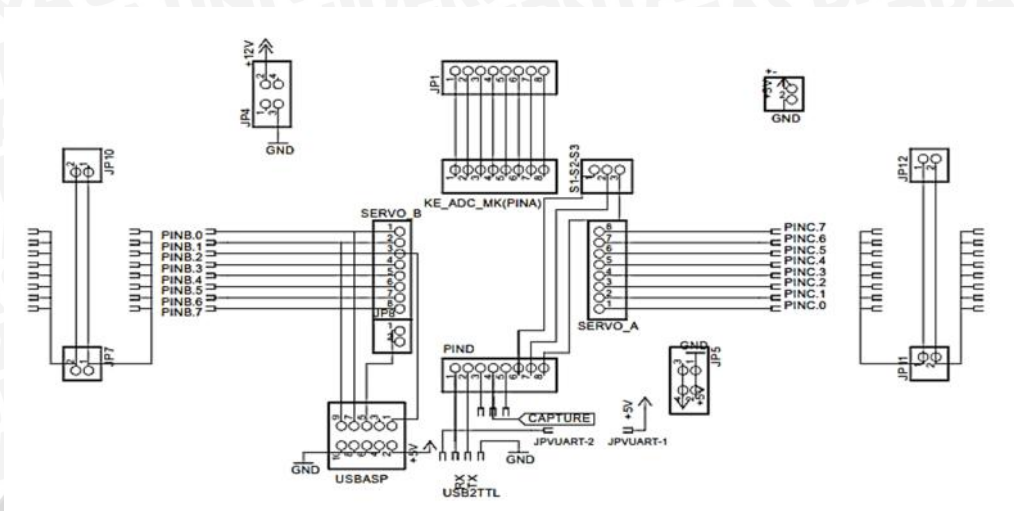
Rangkaian ini hanya terdiri dari header *male* dan *female* yang disolder pada PCB saja. Ada delapan sensor yang digunakan, jadi jalur yang disediakan adalah masing-masing satu pin dari keluaran *microcontroller*, yang pada perancangan ini menggunakan PORTA.0 hingga PORTA.7. PORTA dipilih karena memiliki fasilitas ADC yang dibutuhkan untuk mengubah sinyal analog dari potensiometer menjadi data digital yang dapat diproses oleh *microcontroller*.

#### 4.2.2 Perancangan Rangkaian Mainboard Microcontroller

Rangkaian ini merupakan basis dari sistem elektronik yang digunakan. Rangkaian ini mengakomodasi *minimum system* ATmega16 SMD dengan berbagai perangkat luar. Fitur yang tersedia untuk rangkaian antarmuka ini adalah soket ATmega16 SMD, delapan pin ADC, enam belas antarmuka servo, satu paket antarmuka UART, antarmuka 10-pin USB ASP *programmer*, enam pin



general I/O, serta konektor catu daya. Gambar 4.8 menunjukkan rangkaian *microcontroller mainboard*.



Gambar 4.8 Skematik Rangkaian *Mainboard Microcontroller*

Pada antarmuka *servo*, terdapat konektor catu daya, yaitu tegangan langsung baterai *Lithium Polymer 2 cell* (7,3 - 8,6 V), sesuai spesifikasi yang digunakan. Apabila *servo* yang digunakan membutuhkan tegangan yang lebih kecil, konektor catu daya bisa dihubungkan ke rangkaian regulator eksternal. Terdapat maksimal 2 x 8 *servo*, yang dihubungkan pada PORTC dan PORTB.

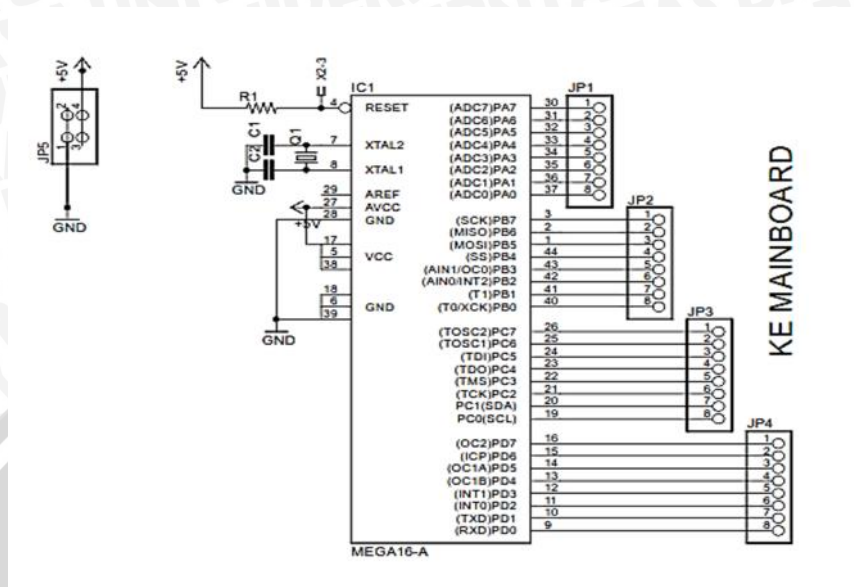
PORTA digunakan sebagai masukan data analog sensor. PORTA dihubungkan dengan barisan header yang bisa langsung dihubungkan dengan rangkaian antarmuka sensor potensiometer.

Untuk data dari tombol *capture* dimasukkan ke dalam PORTD.3. Sementara itu PORTD.0 dan PORTD.1 digunakan untuk komunikasi UART yang dihubungkan ke modul USB to TTL. Bagian PORTD lainnya dapat difungsikan sebagai pin I/O biasa.

#### 4.2.3 Perancangan Rangkaian ATmega16 SMD

Rangkaian ini merupakan rangkaian *minimum system* sekaligus soket bagi ATmega16 SMD. Rangkaian ini menghubungkan pin ATmega16 dengan keluaran pada *mainboard*. Sumber *clock* menggunakan *External XTAL* 16 MHz. Kapasitor kembar yang menghubungkan dua kaki XTAL bernilai 22 pF. Resistor *Pull-Up* yang menghubungkan pin RESET dan VCC adalah 4,7 kΩ. Untuk ADC, tegangan referensi yang digunakan melalui pin AVCC, yang diparalel dengan VCC. Rangkaian seperti dalam Gambar 4.9.



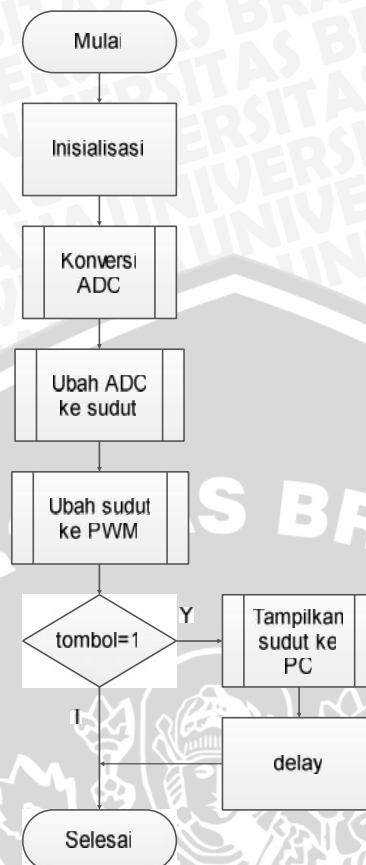


Gambar 4.9 Skematik Rangkaian ATmega16 SMD

### 4.3 Perancangan Perangkat Lunak

#### 4.3.1 Perancangan Perangkat Lunak Utama

Perangkat lunak utama merupakan perangkat lunak yang menangani sistem secara keseluruhan. Perangkat lunak ini terdiri dari suatu rutin program sesuai flowchart, yang didalamnya terdapat sub fungsi untuk menangani langkah-langkah program yang membutuhkan penanganan khusus. Algoritma program utama ditunjukkan dalam Gambar 4.10.

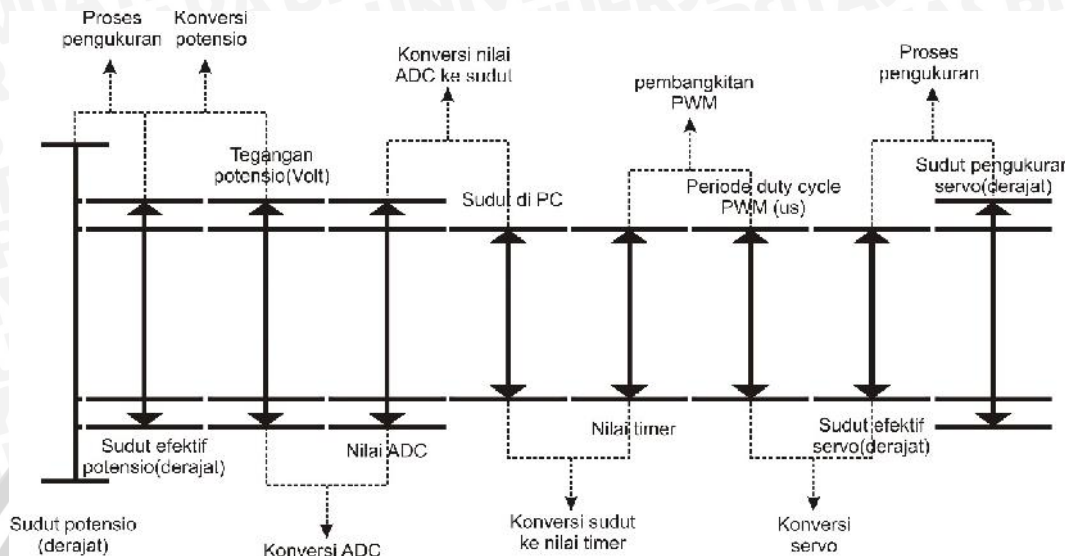


Gambar 4.10 Diagram Alir Program Utama

Pertama, proses yang berjalan adalah inisialisasi ADC, *interrupt*, *timer*, serta UART. ADC dibutuhkan untuk pembacaan data sensor. *Timer dan interrupt digunakan* untuk membangkitkan PWM untuk menggerakkan banyak Motor DC Servo. UART digunakan untuk *serial terminal*.

Setelah inisialisasi, selanjutnya adalah pembacaan ADC. Nilai ADC yang telah dibaca kemudian dimasukkan kedalam variabel “pot”. Data kemudian diproses ke fungsi yang mengubah data ADC 8 bit menjadi data sudut, yang disimpan dalam variabel “sudut”. Variabel “sudut” nantinya akan digunakan untuk ditampilkan ke PC. Setelah itu, data sudut diubah menjadi nilai digital untuk membangkitkan sinyal PWM melalui sebuah fungsi konversi. Pada fungsi ini, nilai digital tersebut akan dikonversi menjadi nilai  $T_{ON}$  (lebar periode logika tinggi) sinyal PWM untuk menggerakkan Motor DC Servo. Pembangkitan sinyal PWM dilakukan oleh *Timer/Counter* yang sudah diinisialisasi pada awal program. Setelah robot bergerak, robot masih bisa bergerak lagi sesuai data sudut yang dibaca oleh sensor, apabila tombol *Capture* tidak ditekan. Apabila tombol

*Capture* ditekan, maka fungsi untuk menampilkan variabel “sudut” ke PC melalui *Serial Terminal* akan dipanggil. Kemudian program akan berulang lagi dari awal setelah diberi *delay*.



Gambar 4.11 Diagram konversi program

Gambar 4.11 menunjukkan proses diagram konversi yang dibutuhkan program. Diagram tersebut dapat menjadi patokan untuk mencari batas rentang data masukan dan keluaran yang akan diolah. Untuk mengolah data masukan sudut dari potensiometer, menjadi sudut untuk ditampilkan pada PC dan untuk menggerakkan motor DC servo, dibutuhkan sebuah proses konversi yang runtut. Proses konversi yang berjalan dalam perangkat lunak antara lain konversi ADC, konversi nilai ADC ke sudut, konversi sudut ke nilai timer, serta konversi nilai timer ke nilai  $T_{ON}$  sinyal PWM.

Potensiometer dan servo yang digunakan tidak memiliki *datasheet* dari pabrikannya, jadi untuk mendapatkan parameter yang dibutuhkan untuk proses konversi, diperlukan proses karakterisasi. Dengan asumsi bahwa proses konversi yang dijalankan adalah linear, maka program konversi bisa menggunakan hukum persamaan garis linear, yaitu:

$$y = m.x + c \quad (4-1)$$

Variabel  $y$  adalah keluaran proses konversi,  $x$  adalah masukan proses konversi, sementara  $m$  adalah gradien dan  $c$  adalah konstanta yang nilainya dapat dicari dengan mengetahui dua buah persamaan garis di dua titik. Dengan mencari batas atas dan batas bawah yang diperlukan untuk setiap masukan dan keluaran,



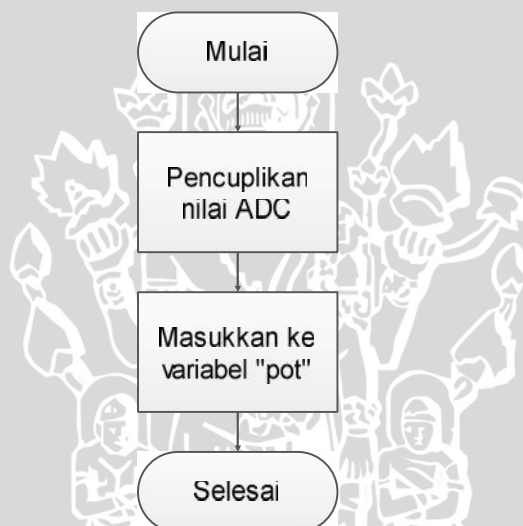
dapat dihasilkan dua persamaan dengan  $m$  dan  $c$  yang sama serta batas keluaran dan masukan yang berbeda.

$$y_0 = m.x_0 + c \quad (4-2)$$

$$y_1 = m.x_1 + c \quad (4-3)$$

#### 4.3.2 Perancangan Perangkat Lunak Konversi ADC

Sesuai fungsinya, *Analog to Digital Converter* (ADC) mengubah sinyal masukan analog menjadi digital agar dapat diolah oleh rangkaian digital, dalam hal ini *microcontroller*. ADC yang digunakan adalah ADC 8-bit. Untuk tegangan referensi ADC, diambil dari pin AVCC pada *microcontroller*. Sementara frekuensi *clock* ADC yang digunakan adalah 1MHz.



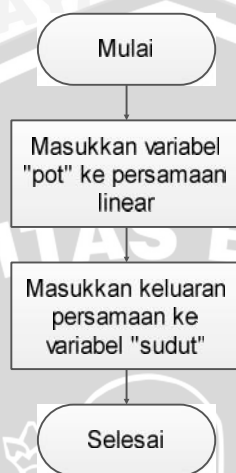
Gambar 4.12 Diagram alir program konversi ADC

Gambar 4.12 menunjukkan diagram alir fungsi konversi ADC. Proses ADC berjalan saat fungsi konversi ADC dipanggil. Kemudian hasil pencuplikan nilai ADC tersebut dimasukkan ke dalam sebuah *array* variabel “pot” yang berjumlah delapan. Kemudian proses selesai sampai fungsi dipanggil kembali.

#### 4.3.3 Perancangan Perangkat Lunak Konversi Nilai ADC ke Sudut

Konversi ini diperlukan untuk mengetahui nilai sudut yang mewakili hasil konversi ADC. Konversi menggunakan persamaan linear, seperti pada persamaan (4-1) untuk menemukan keluaran sudut yang sesuai. Parameter yang digunakan adalah batas atas dan batas bawah nilai ADC sebagai masukan serta nilai sudut sebagai keluaran. Nilai ADC didapatkan dari tabel karakterisasi

tegangan keluaran potensiometer terhadap sudutnya dan persamaan konversi tegangan analog ke nilai digitalnya. Sudut yang digunakan adalah sudut efektif potensiometer dengan rentang yang sesuai dengan sudut efektif motor DC *servo*. Persamaan didapatkan dengan mencari nilai gradien dan konstanta melalui metode substitusi dua persamaan linear dua variabel.



Gambar 4.13 Diagram alir program konversi nilai ADC ke sudut

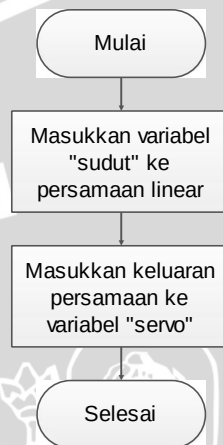
Gambar 4.13 menunjukkan diagram alir fungsi konversi nilai ADC ke sudut. Saat fungsi ini dipanggil, variabel “pot” dimasukkan ke dalam persamaan linear yang telah didapatkan. Keluaran persamaan linear kemudian akan dimasukkan ke dalam variabel “sudut” yang berjumlah delapan. Kemudian proses selesai sampai fungsi dipanggil kembali.

#### 4.3.4 Perancangan Perangkat Lunak Konversi Sudut ke PWM

Konversi sudut ke sinyal PWM dibagi menjadi dua tahap. Pertama, data sudut dikonversi menjadi nilai TOP yang menjadi parameter akhir siklus positif sinyal PWM. Kemudian nilai TOP tersebut dimasukkan ke dalam algoritma pembangkitan sinyal PWM yang dapat dilakukan dengan *Timer/Counter*.

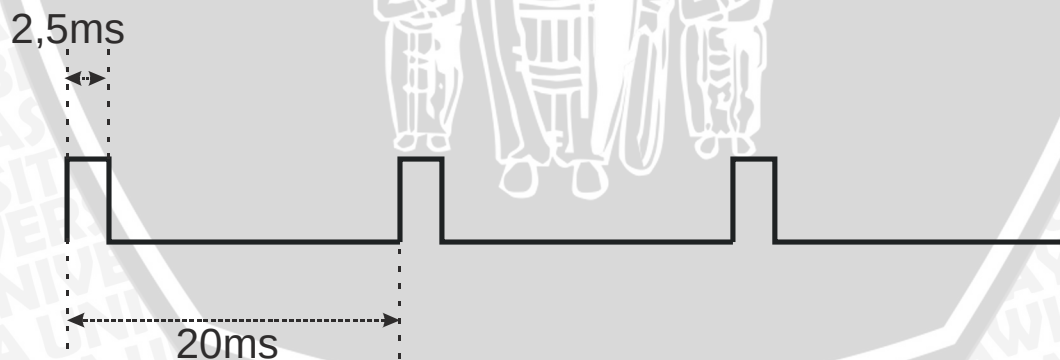
Konversi sudut ke nilai TOP menggunakan persamaan linear, seperti pada persamaan (4-1) untuk menemukan keluaran nilai TOP yang sesuai. Parameter yang digunakan adalah batas atas dan batas bawah nilai sudut sebagai masukan serta nilai TOP sebagai keluaran. Sudut yang digunakan adalah sudut efektif potensiometer dengan rentang yang sesuai dengan sudut efektif motor DC *servo*. Sudut yang digunakan adalah sudut efektif motor DC *servo*. Rentang nilai TOP didapatkan dari karakterisasi sudut motor DC *servo* terhadap nilai TOP yang

mempengaruhinya. Nilai TOP tertentu akan menghasilkan sinyal PWM dengan lebar  $T_{ON}$  tertentu. Sementara itu motor DC servo memiliki batas lebar  $T_{ON}$  sinyal PWM tertentu yang bisa digunakan untuk menggerakkan dan menahan posisi *stall*. Persamaan didapatkan dengan mencari nilai gradien dan konstanta melalui metode substitusi dua persamaan linear dua variabel.



Gambar 4.14 Diagram alir program konversi sudut ke nilai TOP.

Gambar 4.14 menunjukkan diagram alir program konversi sudut ke nilai TOP. Saat fungsi ini dipanggil, variabel “sudut” dimasukkan ke dalam persamaan linear yang telah didapatkan. Keluaran persamaan linear kemudian akan dimasukkan ke dalam delapan variabel “servo”. Kemudian proses selesai sampai fungsi dipanggil kembali.



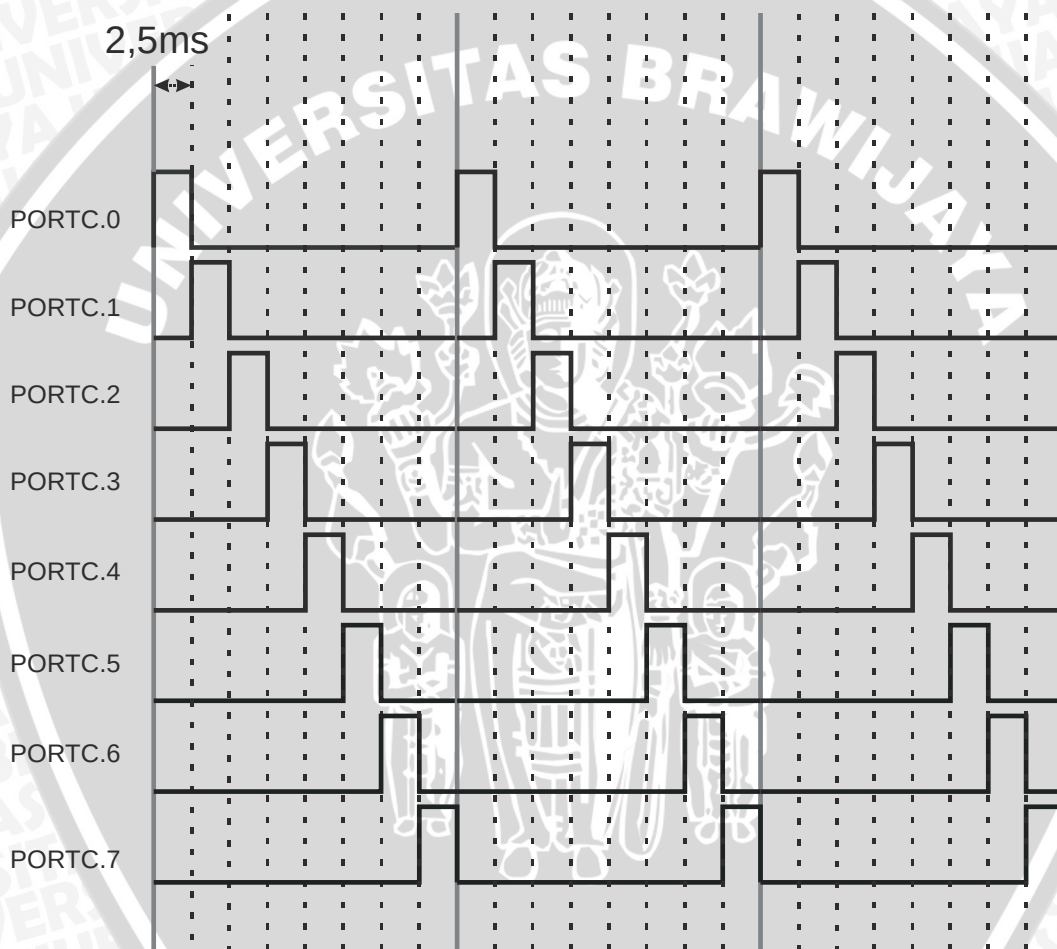
Gambar 4.15 Sinyal kontrol motor DC servo

Selanjutnya adalah membangkitkan sinyal PWM. Gambar 4.13 menunjukkan sinyal PWM yang dibutuhkan untuk dapat mengontrol motor DC servo. Untuk dapat mengontrol motor DC servo, diperlukan sinyal PWM dengan periode 20 ms. Sementara itu, siklus positifnya hanya membutuhkan lebar  $T_{ON}$



maksimal sekitar 2,5 ms pada awal satu siklus periodik, selebihnya adalah siklus negatif. Sinyal ini harus berulang terus secara periodik, jika tidak maka servo kehilangan kemampuan untuk mempertahankan posisi.

Motor DC servo yang akan dikontrol tidak harus dikeluarkan melalui pin *Timer/Counter*, tetapi bisa juga melalui pin I/O. Dengan menggunakan pin I/O, servo yang bisa dikontrol jumlahnya lebih banyak. Untuk mengeluarkan sinyal PWM pada pin biasa, bisa dilakukan dengan menggilir keluaran *timer* secara bergantian ke setiap pin I/O, seperti ditunjukkan dalam Gambar 4.16



Gambar 4.16 Sinyal kontrol motor DC servo jamak pada pin I/O

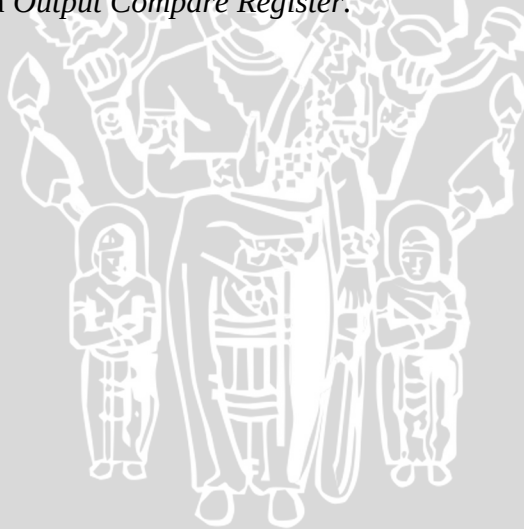
Pin I/O yang ingin digunakan, misalnya PORTC dapat diatur untuk mengeluarkan sinyal PWM dari satu buah keluaran *Timer/Counter*. Sinyal PWM pada satu *timer* diinterupsi setiap 2,5 ms, ditunjukkan oleh garis putus-putus dalam Gambar 4.16. Setelah interupsi selesai dieksekusi, pembangkitan sinyal PWM dimulai lagi, dan sinyal dikeluarkan ke pin selanjutnya. Proses berulang terus hingga maksimal delapan kali. Lebar  $T_{ON}$  satu sinyal PWM pada satu pin

yang berbeda dapat diatur melalui *Output Compare Register* (OCR) yang diisi oleh nilai TOP dari hasil konversi sudut. Dengan menggunakan satu buah *Timer/Counter1* (16-bit), dapat dihasilkan sinyal berdasarkan nilai TOP dari 0 hingga 65535. Dua buah OCR, yaitu OCR1A dan OCR1B juga dapat digunakan untuk menggerakkan hingga 16 buah *servo* untuk masing-masing delapan *servo*. Perhitungan nilai pada register OCRA dan OCRB diperoleh dari datasheet ATmega16 sebagai berikut:

$$f_{OCPWM} = \frac{f_{clk\_I/O}}{N \cdot 1 + TOP} \quad (4-4)$$

$$TOP = \frac{f_{clk\_I/O}}{N \cdot f_{OCPWM}} - 1 \quad (4-5)$$

$f_{OCnPWM}$  adalah frekuensi PWM pada mode *output compare*. Dengan mengetahui frekuensinya, maka periode output compare adalah satu dibagi frekuensinya.  $f_{clk\_I/O}$  adalah frekuensi clock yang digunakan, dalam perancangan ini menggunakan *clock* 16 MHz. Nilai N adalah *prescaler* atau pembagi. Dan TOP adalah nilai pada *Output Compare Register*.

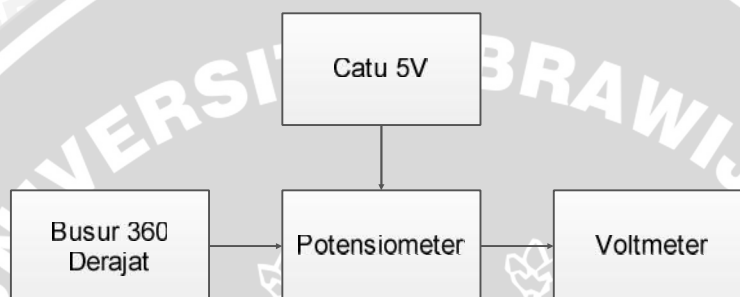


## BAB V

### PENGUJIAN DAN ANALISIS

#### 5.1 Pengujian Potensiometer

Pengujian potensiometer dilakukan untuk menguji tegangan keluaran terhadap posisi sudut potensiometer. Pengujian dilakukan dengan menggunakan busur 360°, plat penunjuk, dan voltmeter. Diagram blok pengujian ini ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Diagram Blok Pengujian Potensiometer

Parameter yang diamati adalah tegangan keluaran potensiometer yang diukur setiap lima derajat. Busur 360 derajat dengan diameter 20 cm dilubangi tengahnya untuk dipasang Potensiometer. Pada tengah *knob* potensiometer, dipasang aluminium berketebalan 2mm dengan panjang 27 cm, sebagai jarum penunjuk yang menunjukkan sudut yang dimaksud dan sudut yang berlawanan. Tegangan keluaran potensiometer diukur dengan multimeter SANWA CD771.

Potensiometer disusun sebagai pembagi tegangan, dan yang diukur adalah pin *ground* dan data. Arah orientasi Potensiometer adalah *clockwise* (CW). Pengukuran diambil setiap 5°, dengan memutar *knob* hingga posisi maksimal. Potensiometer dapat diputar hingga 305° dengan keluaran tegangan dari 0 - 5,08 V.

Untuk tegangan keluaran potensiometer, secara teori dapat menggunakan persamaan (2-6). Dengan mengibaratkan parameter resistansi sebagai posisi sudut potensiometer, didapat persamaan:

$$V_{out} = V_{im} [\theta / \theta_{total}] \quad (5-1)$$

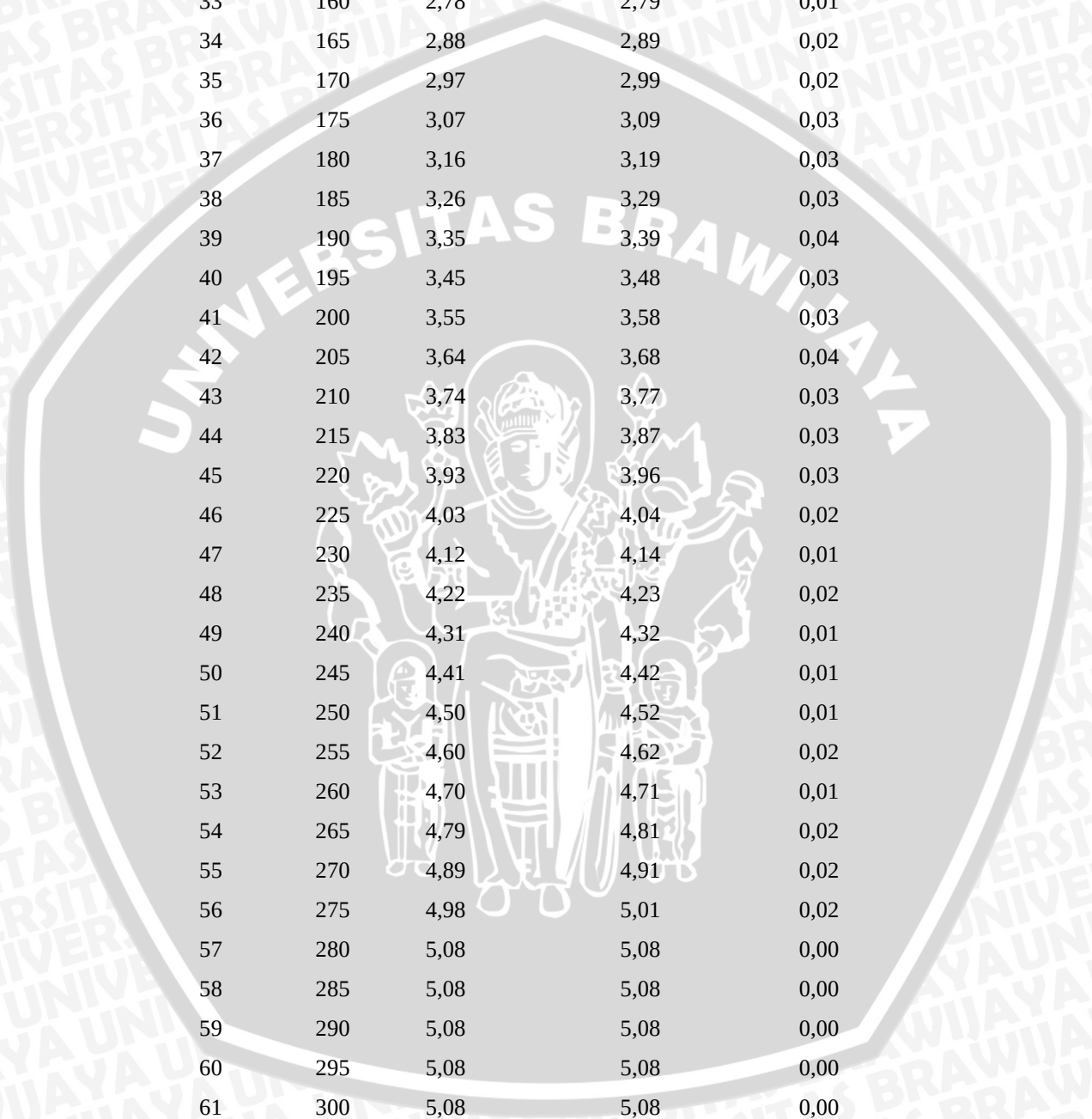
$V_{im}$  adalah 5,08 V,  $\theta$  adalah posisi sudut potensiometer, dan  $\theta_{total}$  adalah rentang sudut maksimal potensiometer. Dari hasil percobaan, ternyata dari sudut



total yang bisa dijelajah potensiometer, hanya rentang sudut tertentu yang dapat mengkonversi posisi sudut potensiometer menjadi tegangan, yaitu sudut efektif. Karena sudut ideal memiliki batas bawah  $0^\circ$ , maka untuk potensiometer yang memiliki sudut efektif, nilai  $\theta$  adalah sudut posisi potensiometer dikurangi batas bawah sudut efektifnya. Sedangkan untuk  $\theta_{total}$  adalah rentang sudut efektif, yaitu batas atas dikurangi batas bawah sudut efektif tersebut. Hasil percobaan dicatat dalam Tabel 5.1.

Tabel 5.1 Data Hasil Pengujian Potensiometer

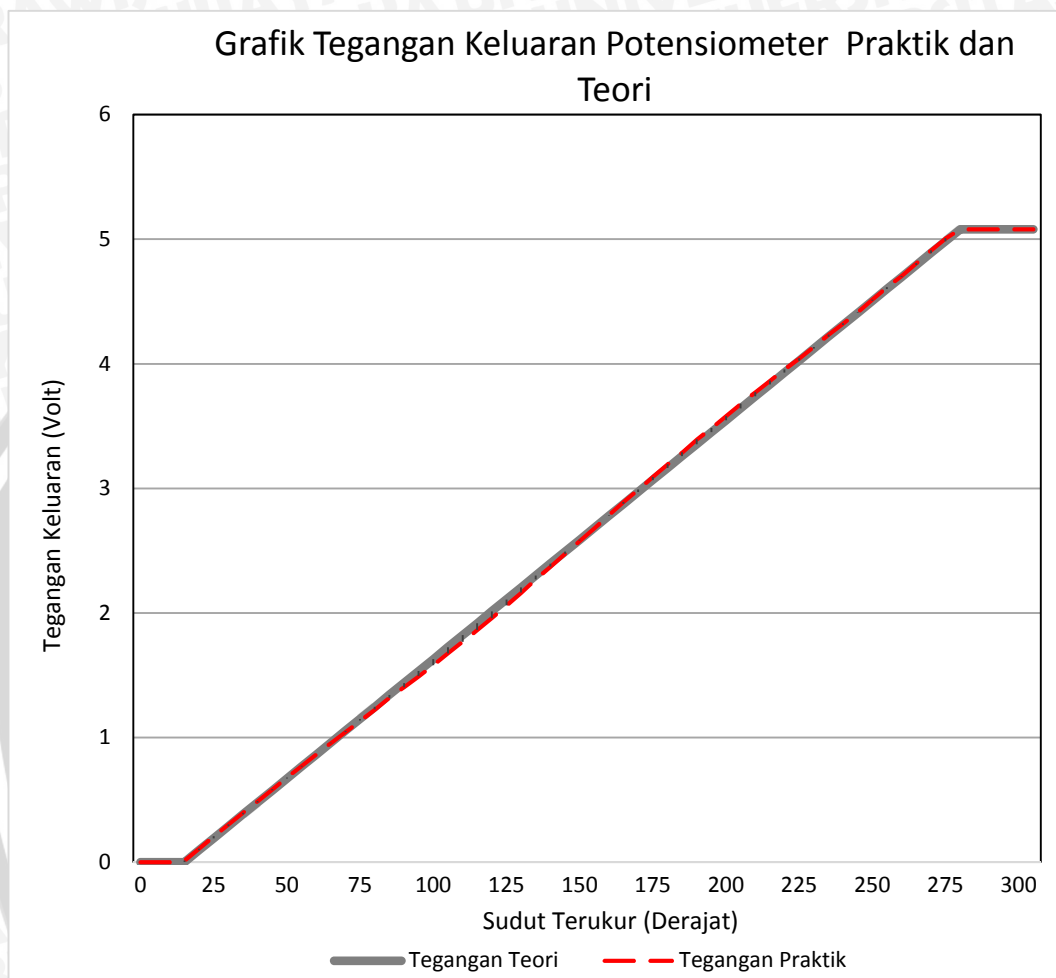
| No. | Sudut ( $^\circ$ ) | Tegangan Keluaran Potensiometer (V) |                |           |
|-----|--------------------|-------------------------------------|----------------|-----------|
|     |                    | Teori                               | Rerata Praktik | Kesalahan |
| 1   | 0                  | 0,00                                | 0,00           | 0,00      |
| 2   | 5                  | 0,00                                | 0,00           | 0,00      |
| 3   | 10                 | 0,00                                | 0,00           | 0,00      |
| 4   | 15                 | 0,00                                | 0,01           | 0,01      |
| 5   | 20                 | 0,10                                | 0,11           | 0,01      |
| 6   | 25                 | 0,19                                | 0,21           | 0,02      |
| 7   | 30                 | 0,29                                | 0,30           | 0,02      |
| 8   | 35                 | 0,38                                | 0,40           | 0,01      |
| 9   | 40                 | 0,48                                | 0,49           | 0,01      |
| 10  | 45                 | 0,58                                | 0,59           | 0,01      |
| 11  | 50                 | 0,67                                | 0,68           | 0,01      |
| 12  | 55                 | 0,77                                | 0,77           | 0,01      |
| 13  | 60                 | 0,86                                | 0,86           | 0,00      |
| 14  | 65                 | 0,96                                | 0,95           | 0,00      |
| 15  | 70                 | 1,05                                | 1,04           | 0,01      |
| 16  | 75                 | 1,15                                | 1,14           | 0,01      |
| 17  | 80                 | 1,25                                | 1,23           | 0,02      |
| 18  | 85                 | 1,34                                | 1,32           | 0,02      |
| 19  | 90                 | 1,44                                | 1,40           | 0,03      |
| 20  | 95                 | 1,53                                | 1,49           | 0,04      |
| 21  | 100                | 1,63                                | 1,58           | 0,05      |
| 22  | 105                | 1,73                                | 1,68           | 0,05      |
| 23  | 110                | 1,82                                | 1,77           | 0,05      |
| 24  | 115                | 1,92                                | 1,86           | 0,05      |
| 25  | 120                | 2,01                                | 1,96           | 0,05      |
| 26  | 125                | 2,11                                | 2,06           | 0,05      |
| 27  | 130                | 2,20                                | 2,16           | 0,04      |



|           |     |      |      |      |
|-----------|-----|------|------|------|
| 28        | 135 | 2,30 | 2,27 | 0,03 |
| 29        | 140 | 2,40 | 2,37 | 0,02 |
| 30        | 145 | 2,49 | 2,48 | 0,02 |
| 31        | 150 | 2,59 | 2,58 | 0,01 |
| 32        | 155 | 2,68 | 2,68 | 0,00 |
| 33        | 160 | 2,78 | 2,79 | 0,01 |
| 34        | 165 | 2,88 | 2,89 | 0,02 |
| 35        | 170 | 2,97 | 2,99 | 0,02 |
| 36        | 175 | 3,07 | 3,09 | 0,03 |
| 37        | 180 | 3,16 | 3,19 | 0,03 |
| 38        | 185 | 3,26 | 3,29 | 0,03 |
| 39        | 190 | 3,35 | 3,39 | 0,04 |
| 40        | 195 | 3,45 | 3,48 | 0,03 |
| 41        | 200 | 3,55 | 3,58 | 0,03 |
| 42        | 205 | 3,64 | 3,68 | 0,04 |
| 43        | 210 | 3,74 | 3,77 | 0,03 |
| 44        | 215 | 3,83 | 3,87 | 0,03 |
| 45        | 220 | 3,93 | 3,96 | 0,03 |
| 46        | 225 | 4,03 | 4,04 | 0,02 |
| 47        | 230 | 4,12 | 4,14 | 0,01 |
| 48        | 235 | 4,22 | 4,23 | 0,02 |
| 49        | 240 | 4,31 | 4,32 | 0,01 |
| 50        | 245 | 4,41 | 4,42 | 0,01 |
| 51        | 250 | 4,50 | 4,52 | 0,01 |
| 52        | 255 | 4,60 | 4,62 | 0,02 |
| 53        | 260 | 4,70 | 4,71 | 0,01 |
| 54        | 265 | 4,79 | 4,81 | 0,02 |
| 55        | 270 | 4,89 | 4,91 | 0,02 |
| 56        | 275 | 4,98 | 5,01 | 0,02 |
| 57        | 280 | 5,08 | 5,08 | 0,00 |
| 58        | 285 | 5,08 | 5,08 | 0,00 |
| 59        | 290 | 5,08 | 5,08 | 0,00 |
| 60        | 295 | 5,08 | 5,08 | 0,00 |
| 61        | 300 | 5,08 | 5,08 | 0,00 |
| 62        | 305 | 5,08 | 5,08 | 0,00 |
| Rata-rata |     |      |      | 0,02 |

Dari tabel percobaan, sudut efektif potensiometer sebesar  $265^{\circ}$ , dengan batas bawah  $15^{\circ}$  dan batas atas  $280^{\circ}$ . Kesalahan rata-rata potensiometer adalah

0,02 V, kesalahan terendah 0 V, dan kesalahan tertinggi 0,05 V. Data pada tabel dapat diolah menjadi grafik keluaran tegangan potensiometer terhadap sudut, pada Gambar 5.2.



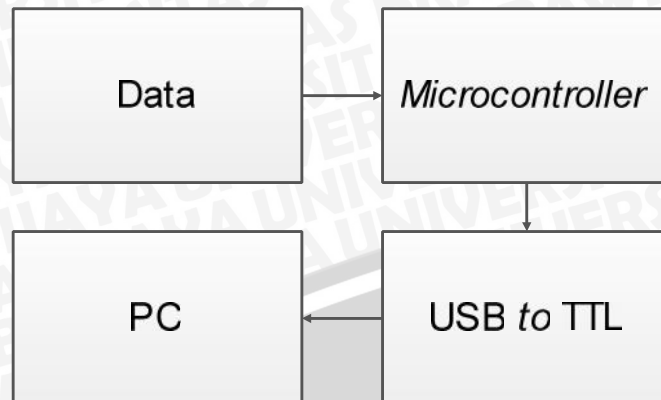
Gambar 5.2 Grafik Pengujian Potensiometer

Dari Pengujian Potensiometer, dapat disimpulkan bahwa keluarannya cukup linear, dengan *error* yang kecil, dan ada sudut efektif dimana potensiometer bisa berubah-ubah tegangan terhadap sudutnya.

## 5.2 Pengujian *Serial Terminal*

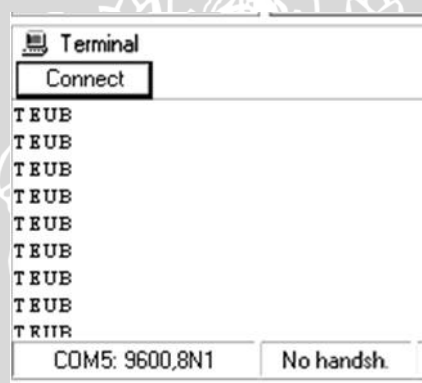
Pengujian *serial terminal* dapat dilakukan dengan mengetikkan data tertentu ke dalam sebuah program, kemudian diatur agar menuliskan perintah ke komputer. Sementara Gambar 5.3 menunjukkan pelaksanaan pengujian *Serial Terminal*.





Gambar 5.3 Skema Pengujian USB to TTL

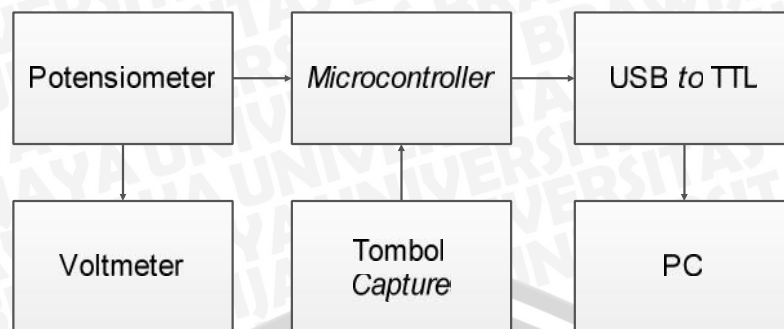
Pada program dituliskan data “TEUB” dan berhasil ditampilkan seperti pada Gambar 16. Data berhasil ditampilkan sesuai dengan yang diinginkan. Hasil pengujian dapat dilihat dalam Gambar 5.4.



Gambar 5.4 Hasil Pengujian USB to TTL

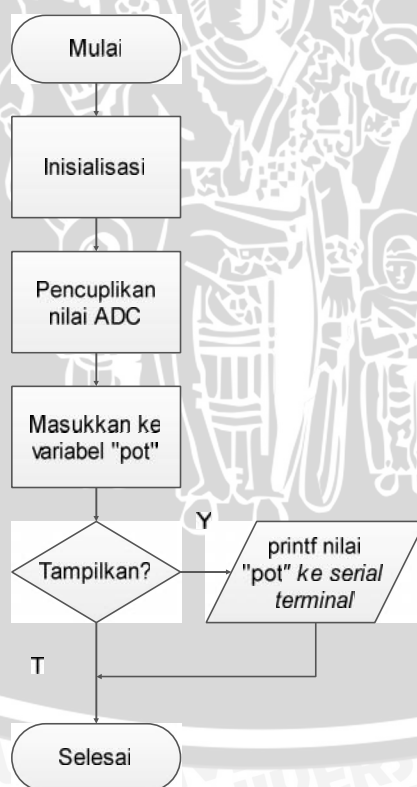
### 5.3 Pengujian ADC *microcontroller*

Pengujian ini dilakukan untuk mengamati perubahan nilai ADC terhadap tegangan masukan analog. Pengujian ini juga dapat membuktikan fungsionalitas dari tombol *capture* yang digunakan. Pengujian dilakukan dengan skema seperti Gambar 5.5.



Gambar 5.5 Skema Pengujian ADC

Tegangan keluaran Potensiometer diamati melalui Multimeter SANWA CD731a, kemudian dimasukkan ke dalam ADC *microcontroller*, untuk dibaca nilai hasil konversinya, kemudian ditampilkan ke PC melalui *Serial Terminal* setelah pencuplikan data dengan tombol *capture*. Kali ini, tidak diperlukan busur derajat dan aluminium, hanya memutar potensiometer hingga didapatkan tegangan yang ingin diamati keluarannya. Hasil konversi ADC ditampilkan ke PC dengan sebuah program pengujian ADC, dengan diagram alir dalam Gambar 5.6.



Gambar 5.6 Diagram alir program pengujian ADC

Data ADC diambil setiap 0,25 V, dari 0 - 5,08 V. Perhitungan nilai ADC didapatkan dari persamaan (2-6). Namun karena persamaan tersebut digunakan

untuk ADC 10-bit, maka persamaan disesuaikan untuk ADC 8-bit. Persamaan (2-7) menjadi:

$$ADC = \frac{V_{in}}{V_{ref}} \times 255 \quad (5-2)$$

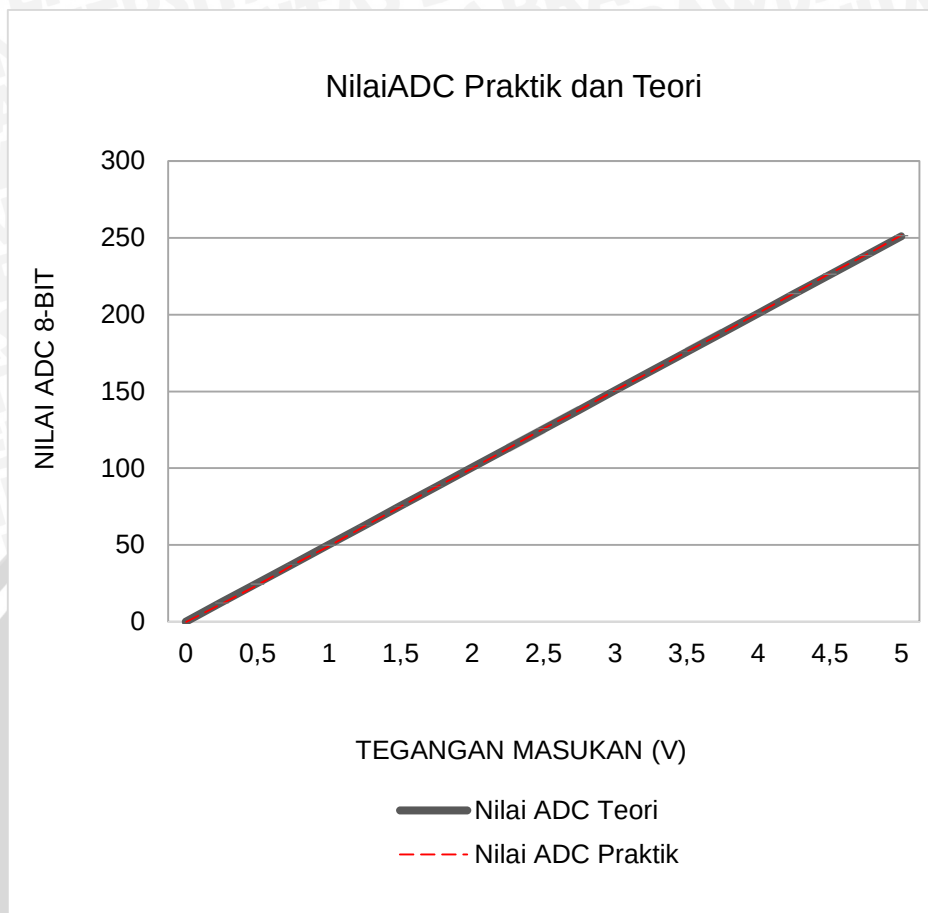
Untuk hasil pengukuran, dapat dilihat dalam tabel 5.2.

Tabel 5.2 Data Hasil Pengujian ADC

| Vref(V)   | Vin(V) | Praktik |      |      |      |      |        | Teori | Kesalahan |
|-----------|--------|---------|------|------|------|------|--------|-------|-----------|
|           |        | adc1    | adc2 | adc3 | adc4 | adc5 | adcrat | adct  |           |
| 5,08      | 0      | 0       | 0    | 0    | 0    | 0    | 0,00   | 0     | 0,00%     |
|           | 0,25   | 12      | 12   | 12   | 12   | 12   | 12,00  | 13    | 4,38%     |
|           | 0,5    | 24      | 24   | 24   | 24   | 24   | 24,00  | 25    | 4,38%     |
|           | 0,75   | 37      | 38   | 38   | 37   | 37   | 37,40  | 38    | 0,66%     |
|           | 1      | 49      | 50   | 49   | 50   | 49   | 49,40  | 50    | 1,59%     |
|           | 1,25   | 63      | 63   | 63   | 62   | 62   | 62,60  | 63    | 0,23%     |
|           | 1,5    | 75      | 75   | 75   | 75   | 75   | 75,00  | 75    | 0,39%     |
|           | 1,75   | 88      | 88   | 88   | 88   | 88   | 88,00  | 88    | 0,18%     |
|           | 2      | 100     | 100  | 100  | 100  | 100  | 100,00 | 100   | 0,39%     |
|           | 2,25   | 113     | 113  | 113  | 113  | 113  | 113,00 | 113   | 0,05%     |
|           | 2,5    | 126     | 126  | 126  | 126  | 126  | 126,00 | 125   | 0,40%     |
|           | 2,75   | 138     | 138  | 138  | 138  | 138  | 138,00 | 138   | 0,03%     |
|           | 3      | 151     | 151  | 151  | 151  | 151  | 151,00 | 151   | 0,27%     |
|           | 3,25   | 163     | 163  | 163  | 163  | 163  | 163,00 | 163   | 0,09%     |
|           | 3,5    | 176     | 176  | 176  | 176  | 176  | 176,00 | 176   | 0,18%     |
|           | 3,75   | 188     | 189  | 188  | 188  | 188  | 188,20 | 188   | 0,02%     |
|           | 4      | 201     | 201  | 202  | 201  | 201  | 201,20 | 201   | 0,21%     |
|           | 4,25   | 214     | 214  | 214  | 214  | 215  | 214,20 | 213   | 0,40%     |
|           | 4,5    | 227     | 227  | 227  | 227  | 226  | 226,80 | 226   | 0,40%     |
|           | 4,75   | 239     | 239  | 239  | 239  | 239  | 239,00 | 238   | 0,24%     |
|           | 5      | 252     | 252  | 252  | 252  | 251  | 251,80 | 251   | 0,33%     |
| Rata-rata |        |         |      |      |      |      |        |       | 0,71%     |

Karena hasil konversi ADC adalah nilai digital, maka hasil pengukuran dibulatkan tanpa angka di belakang koma. Dari tabel, dapat diketahui bahwa persen *error* rata-rata ADC adalah 0,71%, dengan persen *error* tertinggi adalah 4,38%, dan persen *error* terendah adalah 0%. Data dari tabel dapat disimpulkan menjadi grafik pada Gambar 5.6.



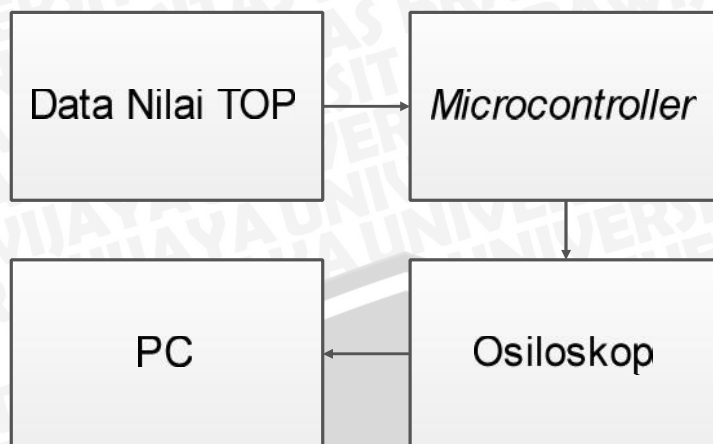


Gambar 5.7 Grafik Pengujian ADC *Microcontroller*

Dari grafik pengujian, dapat disimpulkan bahwa konversi ADC berjalan dengan baik, ditandai dengan grafik keluarannya yang linear. Nilai ADC yang dikeluarkan sudah sesuai dengan perhitungan. Nilai yang dihasilkan berada dalam rentang antara 0-255.

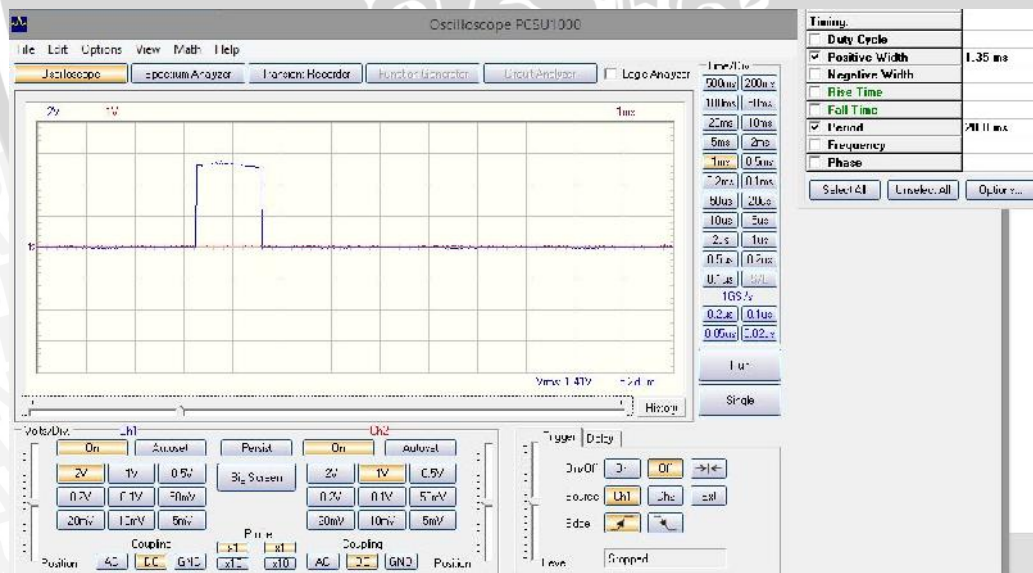
#### 5.4 Pengujian sinyal PWM

Pengujian ini memiliki tujuan untuk mengetahui pengaruh perubahan nilai TOP pada pulsa PWM yang menggerakkan motor DC servo. Skema pengujian PWM motor DC servo seperti dalam Gambar 5.7.



Gambar 5.8 Skema Pengujian Sinyal PWM

Proses pembangkitan sinyal PWM menggunakan algoritma kontrol servo jamak pada pin I/O. Nilai TOP dalam desimal akan dimasukkan ke dalam program dan dikeluarkan ke setiap pin *servo* pada *mainboard* untuk dilihat grafiknya pada osiloskop secara bergantian. Pin yang digunakan adalah PORTC dan PORTB. Sinyal PWM jamak ini dibutuhkan untuk mempercepat dan mempermudah proses pengujian sinyal PWM. Osiloskop yang digunakan merupakan osiloskop Velleman PCLAB PCSU1000. Setelah data ditahan, kemudian disimpan dalam bentuk gambar langsung ke PC. Untuk hasil pengujian seperti dalam Gambar 5.8.



Gambar 5.9 Tampilan Pengukuran Sinyal PWM

Osiloskop memiliki fitur untuk menampilkan *positive width* ( $T_{ON}$ ) dan juga periode sinyal. *Probe* ditancapkan pada *channel 1*, dengan Volt/div 2 V/div

dan time/div 1 ms/div. Sinyal PWM yang diuji berdasarkan nilai TOP dari 9599 hingga 39999. Nilai – nilai tersebut dipilih karena mewakili karakteristik  $T_{ON}$  yang biasa digunakan pada motor DC *servo*. Nilai  $T_{ON}$  teori didapat dari persamaan (4-4). Dengan mengganti nilai frekuensi PWM pada *output compare* menjadi nilai  $T_{ON}$ , diperoleh persamaan:

$$1 / T_{ON} = f_{clk_{I/O}} / N. (1+TOP) \quad (5-3)$$

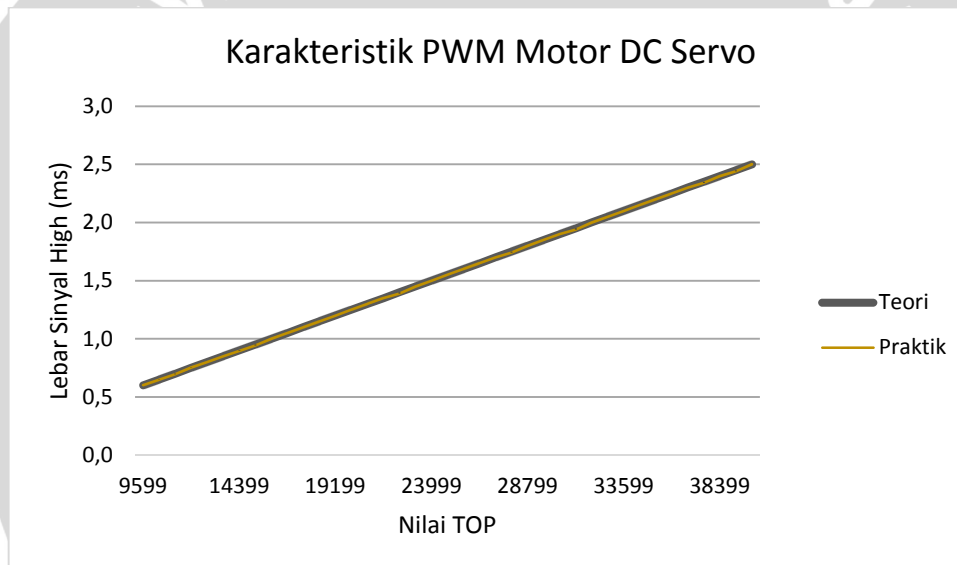
Tabel 5.3 Data Hasil Pengujian sinyal PWM

| Nilai TOP | $T_{ON}$ Teori (ms) | $T_{ON}$ Praktik (ms) | Persen <i>error</i> (%) |
|-----------|---------------------|-----------------------|-------------------------|
| 9599      | 0,60                | 0,60                  | 0,00%                   |
| 10399     | 0,65                | 0,65                  | 0,31%                   |
| 11199     | 0,70                | 0,70                  | 0,57%                   |
| 11999     | 0,75                | 0,75                  | 0,27%                   |
| 12799     | 0,80                | 0,80                  | 0,00%                   |
| 13599     | 0,85                | 0,85                  | 0,24%                   |
| 14399     | 0,90                | 0,90                  | 0,44%                   |
| 15199     | 0,95                | 0,94                  | 0,63%                   |
| 15999     | 1,00                | 1,00                  | 0,00%                   |
| 16799     | 1,05                | 1,05                  | 0,00%                   |
| 17599     | 1,10                | 1,10                  | 0,00%                   |
| 18399     | 1,15                | 1,15                  | 0,00%                   |
| 19199     | 1,20                | 1,20                  | 0,00%                   |
| 19999     | 1,25                | 1,25                  | 0,00%                   |
| 20799     | 1,30                | 1,30                  | 0,00%                   |
| 21599     | 1,35                | 1,35                  | 0,00%                   |
| 22399     | 1,40                | 1,39                  | 0,71%                   |
| 23199     | 1,45                | 1,45                  | 0,00%                   |
| 23999     | 1,50                | 1,50                  | 0,00%                   |
| 24799     | 1,55                | 1,55                  | 0,00%                   |
| 25599     | 1,60                | 1,60                  | 0,00%                   |
| 26399     | 1,65                | 1,65                  | 0,00%                   |
| 27199     | 1,70                | 1,70                  | 0,00%                   |
| 27999     | 1,75                | 1,74                  | 0,57%                   |
| 28799     | 1,80                | 1,80                  | 0,00%                   |
| 29599     | 1,85                | 1,85                  | 0,00%                   |
| 30399     | 1,90                | 1,90                  | 0,00%                   |
| 31199     | 1,95                | 1,94                  | 0,51%                   |
| 31999     | 2,00                | 2,00                  | 0,00%                   |
| 32799     | 2,05                | 2,05                  | 0,00%                   |
| 33599     | 2,10                | 2,10                  | 0,00%                   |
| 34399     | 2,15                | 2,15                  | 0,00%                   |
| 35199     | 2,20                | 2,20                  | 0,00%                   |



|           |      |      |       |
|-----------|------|------|-------|
| 35999     | 2,25 | 2,25 | 0,00% |
| 36799     | 2,30 | 2,30 | 0,00% |
| 37599     | 2,35 | 2,34 | 0,43% |
| 38399     | 2,40 | 2,40 | 0,00% |
| 39199     | 2,45 | 2,44 | 0,41% |
| 39999     | 2,50 | 2,50 | 0,00% |
| Rata-rata |      |      | 0,13% |

Data hasil pengujian ditampilkan dalam Tabel 5.3. Dari tabel di atas, dapat disimpulkan bahwa sinyal PWM yang dibangkitkan *Microcontroller* tergolong akurat. Periode sinyal *high* yang dimasukkan ke program dalam nilai TOP dapat dikeluarkan lagi ke setiap pin dengan persen *error* yang sangat kecil, yaitu 0,13%, dengan persen *error* tertinggi 0,71% dan persen *error* terendah 0%.

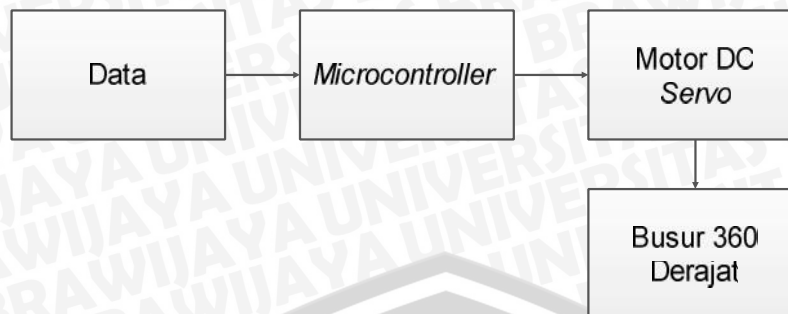


Gambar 5.10 Grafik Pengujian Sinyal PWM

Data dari Tabel 5.3. dapat diubah menjadi grafik pada Gambar 5.10. Seperti pada grafik, keluaran sinyal PWM berupa garis linear dengan data praktik dan teori yang berimpit.

## 5.5 Pengujian motor DC servo

Pengujian dilakukan untuk mengetahui keluaran sudut motor DC servo dari sinyal PWM yang diberikan. Sinyal PWM dikeluarkan oleh *microcontroller* seperti pada percobaan sebelumnya, tetapi pada percobaan ini, sinyal tidak dikeluarkan ke osiloskop, melainkan langsung ke motor DC servo.



Gambar 5.11 Skema Pengujian Motor DC Servo

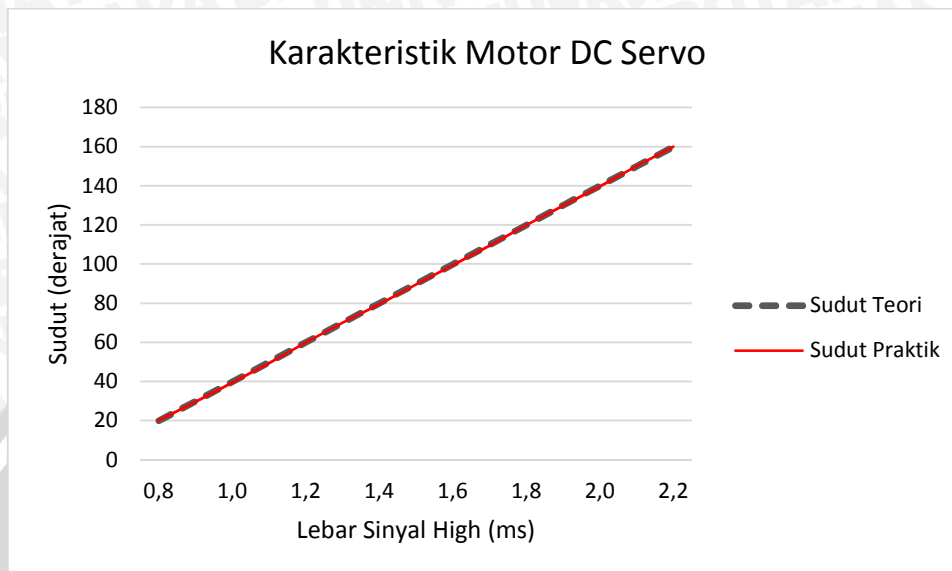
Untuk Pengujian kali ini, motor DC servo dipasang busur lingkaran penuh, dan di bagian *horn* dipasang plat tipis sebagai jarum penunjuknya. Motor DC servo yang diuji adalah Hitec HS-5685. Algoritma yang digunakan sama persis dengan pengujian sinyal PWM, hanya saja keluaran sistem yang diamati adalah motor DC servo. Pemasangan busur pada servo dilakukan dengan menyesuaikan posisi  $90^\circ$  yang diwakili oleh lebar  $T_{ON}$  1,5 us. Untuk menghitung teori sudut motor DC servo dapat dilakukan dengan menggunakan persamaan linear.

Tabel 5.4 Data Hasil Pengujian Motor DC Servo

| $T_{ON}$<br>(us) | Sudut<br>teori | Sudut1 | Sudut2 | Sudut3 | Sudut4 | Sudut5 | Rerata<br>sudut | Persen<br>error |
|------------------|----------------|--------|--------|--------|--------|--------|-----------------|-----------------|
| 800              | 20             | 20     | 19,5   | 20     | 20     | 20,5   | 20,0            | 0,00%           |
| 900              | 30             | 29,5   | 29     | 30     | 29,5   | 29,5   | 29,5            | 1,67%           |
| 1000             | 40             | 39     | 39     | 40     | 39,5   | 39,5   | 39,4            | 1,50%           |
| 1100             | 50             | 49,5   | 49     | 50     | 49,5   | 49     | 49,4            | 1,20%           |
| 1200             | 60             | 60     | 59,5   | 60     | 60     | 60     | 59,9            | 0,17%           |
| 1300             | 70             | 70     | 70     | 70     | 70     | 70     | 70,0            | 0,00%           |
| 1400             | 80             | 79,5   | 79,5   | 80     | 79     | 79     | 79,4            | 0,75%           |
| 1500             | 90             | 89,5   | 90     | 90     | 89     | 89     | 89,5            | 0,56%           |
| 1600             | 100            | 99     | 99,5   | 99     | 100    | 99,5   | 99,4            | 0,60%           |
| 1700             | 110            | 109    | 109    | 109    | 110    | 110    | 109,4           | 0,55%           |
| 1800             | 120            | 120    | 119,5  | 120    | 120    | 120    | 119,9           | 0,08%           |
| 1900             | 130            | 130    | 130    | 129    | 130    | 130    | 129,8           | 0,15%           |
| 2000             | 140            | 139    | 140    | 139    | 140    | 140    | 139,6           | 0,29%           |
| 2100             | 150            | 150    | 150    | 150    | 150    | 150    | 150,0           | 0,00%           |
| 2200             | 160            | 160    | 160    | 160    | 160    | 160    | 160,0           | 0,00%           |
| Rata-rata        |                |        |        |        |        |        | 0,50%           |                 |

Motor DC servo dapat bergerak searah dan berlawanan arah jarum jam, dengan rentang sudut antara  $0^\circ$ - $140^\circ$ , sinyal PWM yang dapat diterima oleh servo

adalah sinyal dengan rentang  $T_{ON}$  antara 0,8 ms - 2,2 ms, dengan persen *error* rata-rata 0,5%, persen *error* maksimal 1,67% dan persen *error* minimal 0%.



Gambar 5.12 Grafik Pengujian Motor DC Servo

Gambar 5.12 merupakan hasil konversi tabel percobaan motor DC servo ke dalam bentuk grafik. Dari grafik dapat diketahui bahwa motor DC servo memiliki linearitas yang baik. Hasil pengukuran dan perhitungan teori berupa dua garis berimpit, menandakan *error* yang sangat kecil.

## 5.6 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini dilakukan dengan menyambungkan semua *hardware* yang dibuat berdasarkan blok diagram dan memasukkan program berupa *software* yang bekerja untuk mengendalikan *hardware* yang telah dibuat. Sistem bekerja dengan baik jika dapat berjalan sesuai *flowchart* yang telah direncanakan.

Pada pengujian ini, sistem yang akan diuji dirangkai mengikuti diagram blok pada Gambar 4.6. Algoritma yang digunakan disusun berdasarkan *flowchart* pada Gambar 4.10. Untuk mengkonversi parameter yang digunakan pada program, dapat mengacu pada Gambar 4.11.

Sesuai dengan perancangan awal, percobaan ini memiliki masukan berupa posisi sudut yang dibaca oleh potensiometer, kemudian dikeluarkan menjadi posisi sudut yang menggerakkan motor DC servo, dan data posisi sudut



ditampilkan pada PC dengan menggunakan fasilitas *serial terminal*. Percobaan ini dititik beratkan pada akurasi proses konversi sudut masukan dan keluaran sistem. Jadi untuk data hasil percobaan, cukup dengan mencoba satu derajat kebebasan. Dengan karakteristik yang sama, percobaan satu DOF ini dapat mewakili percobaan keseluruhan sistem dengan pengujian semua DOF.

Untuk percobaan satu DOF ini, mekanik robot dan *exoskeleton* tidak digunakan. Untuk meningkatkan kepresisian pembacaan parameter sudut, potensiometer dan motor DC *servo* diuji dengan menggunakan sistem busur lingkaran penuh seperti pada pengujian masing-masing blok. Hasil pengujian sistem ditunjukkan dalam Tabel 5.5 dan Tabel 5.6.

Tabel 5.5 Data Hasil Pengujian Keseluruhan Sistem ke Robot

| Sudut Potensio (°) |     | Sudut Motor DC Servo |       |        |      |        |      |            |        | Kesalahan (°) |
|--------------------|-----|----------------------|-------|--------|------|--------|------|------------|--------|---------------|
| Peng               | eff | Praktik              |       |        |      |        |      |            |        |               |
|                    |     | P1 (°)               |       | P2 (°) |      | P3 (°) |      | Rerata (°) |        |               |
|                    |     | Peng°                | Eff   | Peng   | Eff  | Peng   | Eff  | Peng       | Eff    |               |
| 20                 | 0   | 20                   | 0     | 20     | 0    | 20     | 0    | 20,00      | 0,00   | 0,00          |
| 30                 | 10  | 29                   | 9     | 29     | 9    | 29     | 9    | 29,00      | 9,00   | 1,00          |
| 40                 | 20  | 39                   | 19    | 39     | 19   | 39     | 19   | 39,00      | 19,00  | 1,00          |
| 50                 | 30  | 49                   | 29    | 48,5   | 28,5 | 48     | 28   | 48,50      | 28,50  | 1,50          |
| 60                 | 40  | 60                   | 40    | 59     | 39   | 60     | 40   | 59,67      | 39,67  | 0,33          |
| 70                 | 50  | 70                   | 50    | 69,5   | 49,5 | 69     | 49   | 69,50      | 49,50  | 0,50          |
| 80                 | 60  | 80                   | 60    | 78,5   | 58,5 | 79     | 59   | 79,17      | 59,17  | 0,83          |
| 90                 | 70  | 89                   | 69    | 90     | 70   | 90     | 70   | 89,67      | 69,67  | 0,33          |
| 100                | 80  | 99                   | 79    | 99     | 79   | 99     | 79   | 99,00      | 79,00  | 1,00          |
| 110                | 90  | 110                  | 90    | 110    | 90   | 109    | 89   | 109,67     | 89,67  | 0,33          |
| 120                | 100 | 120                  | 100   | 119    | 99   | 120    | 99,5 | 119,50     | 99,50  | 0,50          |
| 130                | 110 | 129,5                | 109,5 | 129    | 109  | 130    | 110  | 129,50     | 109,50 | 0,50          |
| 140                | 120 | 140                  | 120   | 140    | 120  | 140    | 120  | 140,00     | 120,00 | 0,00          |
| 150                | 130 | 150                  | 130   | 150    | 130  | 150    | 130  | 150,00     | 130,00 | 0,00          |
| 160                | 140 | 161                  | 141   | 161    | 141  | 161    | 141  | 161,00     | 141,00 | 1,00          |
| Rata-rata          |     |                      |       |        |      |        |      |            |        | 0,59          |

Tabel 5.5 menunjukkan pengujian keseluruhan sistem dengan sudut posisi aktuator robot (motor DC *servo*) sebagai keluarannya. Pada potensiometer, busur dipasang menyesuaikan patokan sudut 90° motor DC *servo*. Sudut yang terukur pada *servo* kemudian dikonversi menjadi sudut efektif juga. Sudut efektif

digunakan sebagai perbandingan dengan keluaran sudut pada PC. Sudut keluaran *servo* langsung dibandingkan dengan sudut masukan dari potensiometer. Dari tiga kali pengambilan data, sudut keluaran *servo* memiliki batas bawah rata-rata 20° dan batas atas rata-rata 161°. Sementara sudut efektifnya memiliki batas bawah rata-rata 0° dan batas atas rata-rata 141°. Kesalahan rata-rata sistem adalah 0.59° dengan kesalahan terbesar 1,5° dan kesalahan terkecil 0°.

Tabel 5.6 Data Hasil Pengujian Keseluruhan Sistem ke PC

| Sudut Potensio (°) |     | Sudut PC |        |        |            |               |
|--------------------|-----|----------|--------|--------|------------|---------------|
| Peng               | Eff | Praktik  |        |        | Rerata (°) | Kesalahan (°) |
|                    |     | P1 (°)   | P2 (°) | P3 (°) |            |               |
| 20                 | 0   | 0        | 0      | 0      | 0,00       | 0,00          |
| 30                 | 10  | 10       | 9      | 10     | 9,67       | 0,33          |
| 40                 | 20  | 20       | 20     | 20     | 20,00      | 0,00          |
| 50                 | 30  | 29       | 29     | 29     | 29,00      | 1,00          |
| 60                 | 40  | 40       | 39     | 40     | 39,67      | 0,33          |
| 70                 | 50  | 49       | 49     | 49     | 49,00      | 1,00          |
| 80                 | 60  | 60       | 59     | 59     | 59,33      | 0,67          |
| 90                 | 70  | 70       | 70     | 70     | 70,00      | 0,00          |
| 100                | 80  | 80       | 80     | 80     | 80,00      | 0,00          |
| 110                | 90  | 91       | 91     | 91     | 91,00      | 1,00          |
| 120                | 100 | 102      | 101    | 101    | 101,33     | 1,33          |
| 130                | 110 | 112      | 111    | 112    | 111,67     | 1,67          |
| 140                | 120 | 123      | 122    | 123    | 122,67     | 2,67          |
| 150                | 130 | 133      | 133    | 133    | 133,00     | 3,00          |
| 160                | 140 | 144      | 144    | 144    | 144,00     | 4,00          |
| Rata-rata          |     |          |        |        |            | 1,13          |

Tabel 5.6 menunjukkan pengujian keseluruhan sistem dengan data sudut di PC sebagai keluarannya. Sesuai cara kerjanya, setelah aktuator robot *humanoid* digerakkan, pengguna dapat menampilkan posisi sudut gerakan yang sedang terjadi. Dari hasil pengukuran, *serial terminal* mampu menampilkan sudut yang telah dikonversi dalam *microcontroller* dengan batas bawah rata-rata 0° dan batas atas rata-rata 144°. Kesalahan rata-ratanya adalah 1,13° dengan kesalahan terbesar 4° dan kesalahan terkecil 0°.

Karena keperluan *exoskeleton motion capture system* adalah untuk merekam gerakan pada satu *sequence*, maka sudut yang ditampilkan hanya sesekali, pada saat artikulasi sendi tubuh sudah mencapai bentuk gerakan yang diinginkan. Pengguna cukup menggerakkan robot sesuai gerakan tubuhnya untuk mengecek apakah gerakan tersebut sudah sesuai yang diinginkan. Jika sudah, barulah pengguna menekan tombol *capture* untuk menampilkan sudut satu *sequence* gerakan tersebut. Pergerakan robot hanya dititik beratkan untuk mencapai posisi sendi tertentu dalam satu *sequence*. Pergerakan robot diperlukan untuk mewakili gerakan per *frame*, bukan gerakan yang kontinyu. Kecepatan dan kehalusan pergerakan sistem tidak terlalu berpengaruh untuk tujuan tersebut. Jadi, pada dasarnya kecepatan pemrosesan sistem tidak ditonjolkan dalam sistem ini.





## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut.

1. Sensor potensiometer yang digunakan memiliki sudut efektif sebesar  $265^{\circ}$ , dengan batas bawah  $15^{\circ}$  dan batas atas  $280^{\circ}$ , kesalahan terbesar 0,05 V dan kesalahan rata-rata 0,02 V.
2. Transfer data dari *microcontroller* ke PC melalui USB to TTL berhasil dilakukan. Data berhasil ditampilkan sesuai dengan yang diinginkan dengan tingkat keberhasilan 100%.
3. ADC internal *microcontroller* mengubah data masukan tegangan dengan rentang antara 0 - 5,08 V menjadi data digital 8-bit dengan rentang nilai antara 0 - 255, memiliki persen *error* rata-rata 0,71%, dengan persen *error* tertinggi adalah 4,38%, dan persen *error* terendah adalah 0%.
4. Sinyal PWM untuk mengontrol motor DC *servo* mampu dibangkitkan oleh *microcontroller* dengan lebar  $T_{ON}$ , sesuai kebutuhan *servo*, dengan rentang antara 0,6 - 2,5ms, dengan persen *error* terbesar 0,71% dan persen *error* rata-rata 0,13%.
5. Motor DC *servo* dapat bergerak searah dan berlawanan arah jarum jam, dengan rentang sudut antara  $0^{\circ}$ - $140^{\circ}$ , sinyal PWM yang dapat diterima oleh *servo* adalah sinyal dengan rentang  $T_{ON}$  antara 0,8 ms - 2,2 ms, dengan persen *error* rata-rata 0,5%, persen *error* maksimal 1,67% dan persen *error* minimal 0%.
6. Sistem keseluruhan berhasil dijalankan. Sendi *exoskeleton* berhasil menggerakkan sendi robot *humanoid* dengan kesalahan rata-rata  $0.59^{\circ}$ , kesalahan terbesar  $1.5^{\circ}$ , dan kesalahan terkecil  $0^{\circ}$ . Sementara sudut yang ditampilkan ke PC memiliki kesalahan rata-rata  $1.13^{\circ}$ , kesalahan terbesar  $4^{\circ}$ , dan kesalahan terkecil  $0^{\circ}$ .

## 6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah.

1. Penelitian bisa dilakukan dengan menggunakan sensor dengan tingkat kepresisian dan kestabilan yang lebih baik.
2. Bisa menggunakan sistem *motion capture* dengan sensor inersial, visual, atau bahkan neural.
3. Penelitian bisa dilanjutkan dengan menambahkan jumlah DOF pada *upper body* maupun *lower body*.
4. Bisa menggunakan jenis aktuator yang lain.
5. Bisa dikembangkan menjadi *powered exoskeleton*.



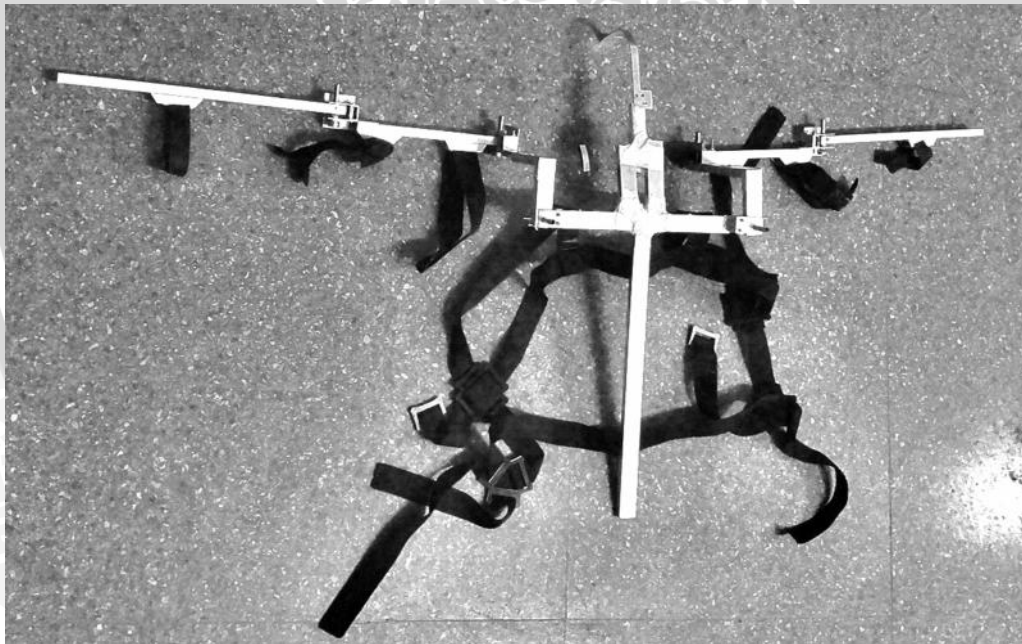
## DAFTAR PUSTAKA

- Atmel. 2007. *8-bit AVR with 16K Bytes In-System Programmable Flash ATmega16, ATmega16L*. San Jose: Atmel.
- Bejo, Agus. 2008. *C & AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATmega8535*. Yogyakarta: Graha Ilmu.
- Dikti. 2013. *Dikti, 2013*: Jakarta: Direktorat Jenderal Pendidikan Tinggi Kementrian Pendidikan dan Kebudayaan.
- Kolozs, James. 1998. *Design Of An Exoskeletal Human Motion Capture System Sensuit™*. Utah: The University of Utah.
- Labs, Silicon. 2013. *CP2102/9: SINGLE-CHIP USB TO UART BRIDGE*. Silicon Laboratories.
- Pinckney, Nathaniel. 2006. *Pulse-Width Modulation For Microcontroller Servo Control*. IEEE Potentials. 25(1):27–29.
- Recktenwald, Gerald. 2013. *Living With The Lab :: Voltage Dividers and Potentiometers*. [https://cdn.sparkfun.com/assets/resources/4/4/input\\_potentiometer.pdf](https://cdn.sparkfun.com/assets/resources/4/4/input_potentiometer.pdf)
- Sawicz, Darren. 2002. *Hobby Servo Fundamentals*. <https://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>
- Shahbazi, Hamed dkk. 2012. *Modeling Of Mesencephalic Locomotor Region For Nao Humanoid Robot*. Iran: Emerald Group Publishing Limited.
- Todd, D.C. 1975. *The Potentiometer Handbook*. USA: Bourns, Inc.
- Zhang, X. D., Kang, L. Y. & Diao, W. F. 2005. *The Principle Of The Potentiometer And Its Applications In The Vehicle Steering*. IEEE International Conference on Vehicular Electronics and Safety. Xi'an Intl. Conference Center. Xian, 14-18 Oktober 2005.



# LAMPIRAN I

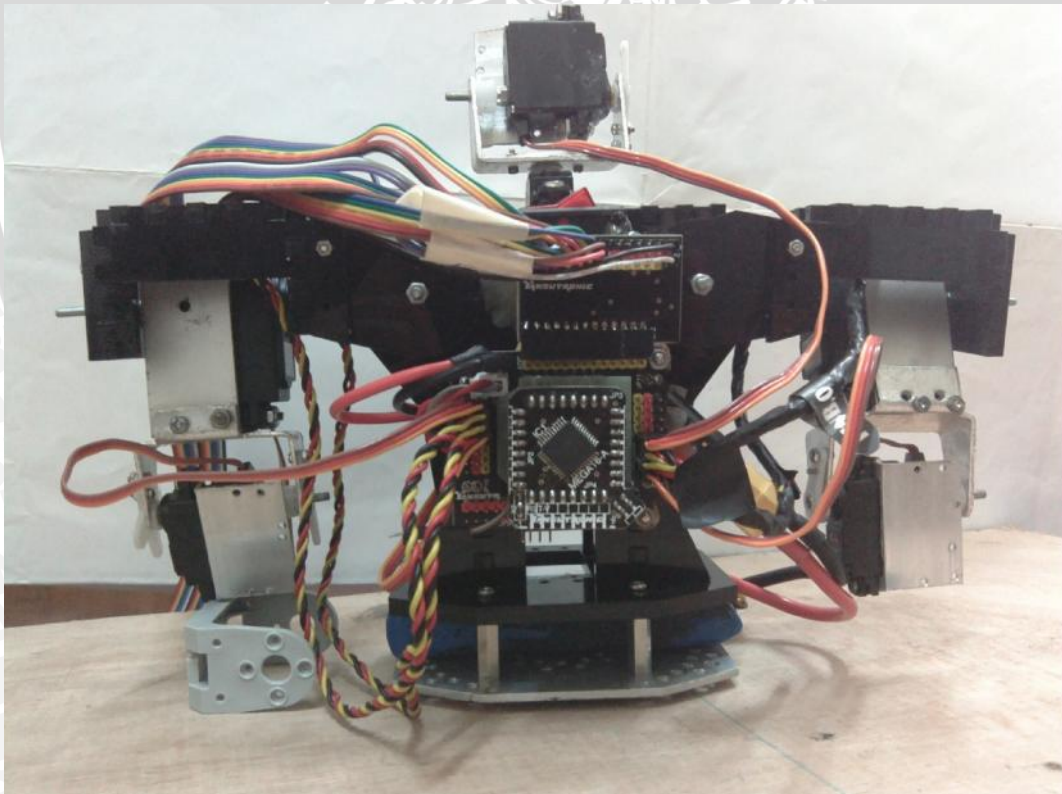
## Dokumentasi Alat



Gambar mekanik exoskeleton tampak belakang.



Gambar mekanik exoskeleton tampak depan.



Gambar Mekanik Robot Humanoid



# LAMPIRAN II

## Listing Program

```

/*****

```

```

This program was produced by the
CodeWizardAVR V2.05.0 Evaluation
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project : exoskeleton motion capture
Version : 1.01
Date : 7/6/2014
Author : Tansu
Company : TEUB
Comments: program utama

```

```

Chip type : ATmega16
Program type : Application
AVR Core Clock frequency: 16.000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256
*****/

```

```

#include <mega16.h>
#include <delay.h>
#include <stdio.h>

```

```

#define ADC_VREF_TYPE 0x60
#define tombol PIND.3

```

```

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // 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;
}

// Declare your global variables here
void baca_adc();
void potkesudut();
void sudutkeservo();
void sudutkepc();
unsigned char pot[8];
unsigned char sudut[8];
unsigned int servoA[4];
unsigned int servoB[4];
char x=0;

// Timer 1 input capture interrupt service routine
interrupt [TIM1_CAPT] void timer1_capt_isr(void)
{
    if (++x>7){x=0;}

    PORTC=0x00;
    PORTB=0x00;

    OCR1A=servoA[x];
    OCR1B=servoB[x];

    switch (x-1)
    {
        case 0 : PORTC=0x01;
                  PORTB=0x01;
                  break;

        case 1 : PORTC=0x02;
                  PORTB=0x02;
                  break;

        case 2 : PORTC=0x04;
                  PORTB=0x04;
                  break;

        case 3 : PORTC=0x08;
                  PORTB=0x08;
                  break;
    }
}

// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    switch (x-1)
```

```
{
    case 0 : PORTC.0=0;
            break;

    case 1 : PORTC.1=0;
            break;

    case 2 : PORTC.2=0;
            break;

    case 3 : PORTC.3=0;
            break;
}

}

// Timer 1 output compare B interrupt service routine
interrupt [TIM1_COMPB] void timer1_compb_isr(void)
{
    switch (x-1)
    {
        case 0 : PORTB.0=0;
                break;

        case 1 : PORTB.1=0;
                break;

        case 2 : PORTB.2=0;
                break;

        case 3 : PORTB.3=0;
                break;

    }
}

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    DDRA=0x00;
    DDRB=0xFF;
    DDRC=0xFF;
    DDRD=0xE3;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 16000.000 kHz
    // Mode: Fast PWM top=ICR1
    // OC1A output: Non-Inv.
```

```
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: On
// Compare A Match Interrupt: On
// Compare B Match Interrupt: On
TCCR1A=0xA2;
TCCR1B=0x19;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x9C;
ICR1L=0x40;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: ADC Stopped
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x38; //0x30;

// Global enable interrupts
#asm("sei")

while (1)
{
    // Place your code here
    baca_adc();
    potkesudut();
    sudutkeservo();
}
```



```

        if (tombol==1)
        {
            Delay_ms(1000);
            sudutkepc();
            delay_ms(1000);
        }
    }

    //membaca nilai adc
    void baca_adc()
    {
        pot[0]=read_adc(0);
        pot[1]=read_adc(1);
        pot[2]=read_adc(2);
        pot[3]=read_adc(3);
        pot[4]=read_adc(4);
        pot[5]=read_adc(5);
        pot[6]=read_adc(6);
        pot[7]=read_adc(7);
    }

    //mengubah data adc ke bentuk sudut
    void potkesudut()
    {
        /*
            y=mx+c
            y=keluaran=sudut[]
            x=masukan=nilai ADC=pot[]

            nilai x diambil dari karakterisasi
            hasil konversi dengan rentang sudut
            masukan potensio 140 derajat (terserah)

            y1=0
            y2=140
            x1(saat 35 derajat, Vin=0.38):
                x1=(0.38/5.08)255
                x1=19.25
            x2(saat 175 derajat, Vin=3.07);
                x2=(3.07/5.08)255
                x2=154.104

            persamaan linear:
                0=19.25m+c
                140=154.104m+c
                m=1.038
                c=-19.9846
        */
    }

```

```
sudut[0]=pot[0]*1.038-19.9846;
sudut[1]=pot[1]*1.038-19.9846;
sudut[2]=pot[2]*1.038-19.9846;
sudut[3]=pot[3]*1.038-19.9846;
sudut[4]=pot[4]*1.038-19.9846;
sudut[5]=pot[5]*1.038-19.9846;
sudut[6]=pot[6]*1.038-19.9846;
sudut[7]=pot[7]*1.038-19.9846;
}
```

```
//mengubah sudut ke nilai pwm servo
void sudutkeservo()
```

```
{
/*
```

Rumus:

$$F_{ocnXPWM} = \frac{F_{clk\_I/O}}{N \cdot (1+TOP)}$$

$$\frac{1}{T} = \frac{16000000}{1(1+TOP)}$$

$$TOP = 16000000 \cdot T - 1$$

y=mx+c

y=keluaran=TOP=servo[]

x=masukan=sudut[]

nilai TOP diambil berdasarkan karakterisasi servo, dengan rentang 140 derajat

y1=12799

y2=35199

x1=0

x2=140

persamaan linear:

$$12799=0m+c$$

$$12799=c$$

$$35199=140m+c$$

$$m=160$$

```
*/
```

```
servoA[0]=160*sudut[0]+12799;
servoA[1]=160*sudut[1]+12799;
servoA[2]=160*sudut[2]+12799;
servoA[3]=160*sudut[3]+12799;
servoB[0]=160*sudut[4]+12799;
servoB[1]=160*sudut[5]+12799;
```

```
servoB[2]=160*sudut[6]+12799;  
servoB[3]=160*sudut[7]+12799;
```

```
}
```

```
//menampilkan data sudut ke PC melalui fasilitas Serial  
Terminal
```

```
void sudutkepc()
```

```
{
```

```
printf ("sudut[0]= %d", sudut[0]); printf ("derajat\r");  
printf ("sudut[1]= %d", sudut[1]); printf ("derajat\r");  
printf ("sudut[2]= %d", sudut[2]); printf ("derajat\r");  
printf ("sudut[3]= %d", sudut[3]); printf ("derajat\r");  
printf ("sudut[4]= %d", sudut[4]); printf ("derajat\r");  
printf ("sudut[5]= %d", sudut[5]); printf ("derajat\r");  
printf ("sudut[6]= %d", sudut[6]); printf ("derajat\r");  
printf ("sudut[7]= %d", sudut[7]); printf ("derajat\r\n\n");
```

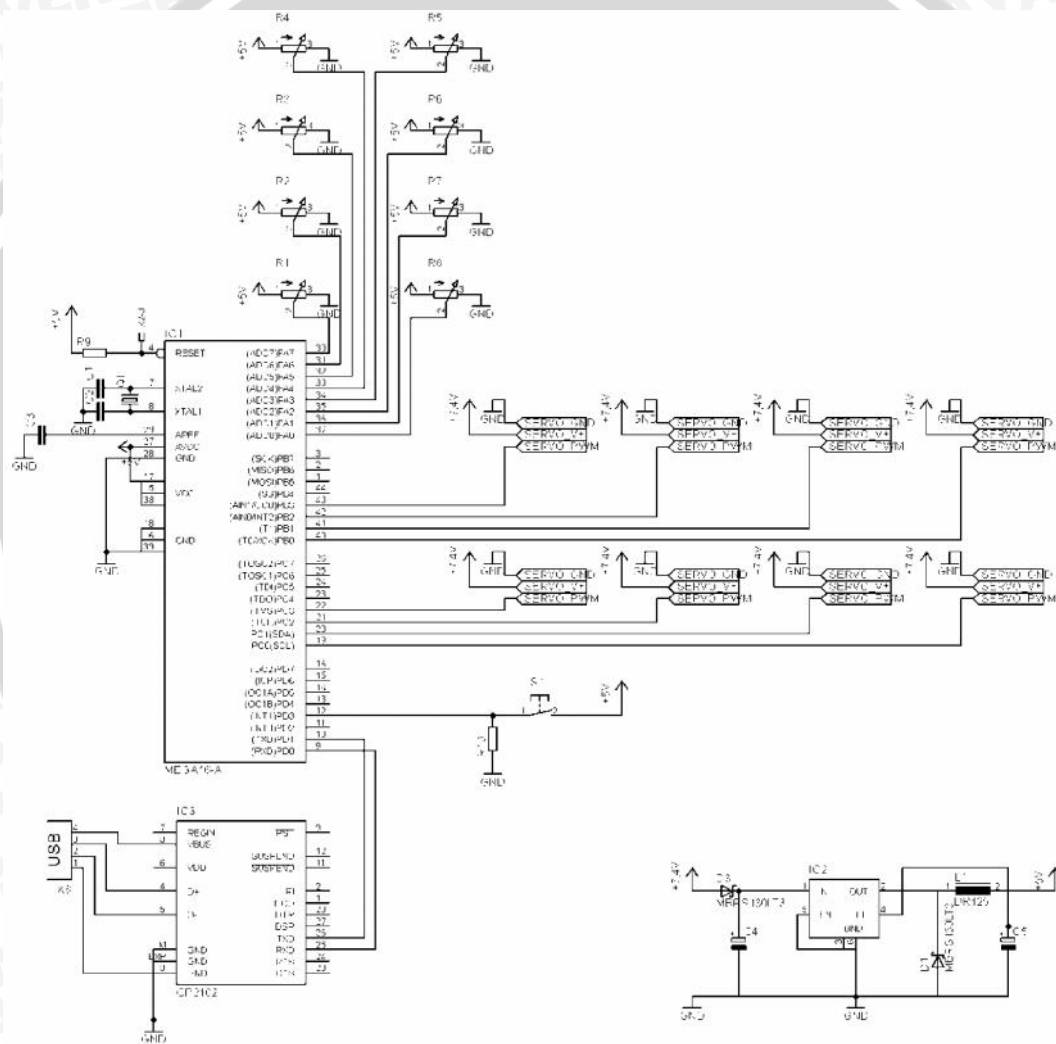
```
}
```





# LAMPIRAN III

## Skematik Keseluruhan



# LAMPIRAN IV

---

Datasheet

UNIVERSITAS BRAWIJAYA

