SISTEM KENDALI SERVO POSISI DAN KECEPATAN DENGAN *PROGRAMMABLE LOGIC CONTROLLER (PLC)*

TUGAS AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan program Strata Satu Jurusan Elektro Fakultas Teknik Universtas Hasanuddin Makassar

Disusun oleh:

IQBAL ZAKARIAH
D411 06 086

ASIFA NURUL HUSNAH
D411 08 255

J U R U S A N E L E K T R O F A K U L T A S T E K N I K
U N I V E R S I T A S H A S A N U D D I N
M A K A S S A R

2 0 1 1

SISTEM KENDALI SERVO POSISI DAN KECEPATAN DENGAN *PROGRAMMABLE LOGIC CONTROLLER (PLC)*

TUGAS AKHIR

Diterima dan disahkan sebagai kolokium
Untuk memenuhi persyaratan guna mencapai
Gelar Sarjana Teknik Sub Program Studi
Teknik Komputer, Sistem Kendali dan Elektronika
Jurusan Elektro Fakultas Teknik Universitas Hasanuddin

Oleh :

IQBAL ZAKARIAH	**ASIFA NURUL HUSNAH**
D411 06 086	**D411 08 255**

Disetujui **:**

Tanggal : ……

Pembimbing Tugas Akhir

Pembimbing I	**Pembimbing II**
Dr. Ir. H. Andani Ahmad, MT	**Ir. Christoforus Y, MT**
NIP : 19601211 198703 1 022	**NIP : 19600716 198702 1 002**

Ketua Jurusan Teknik Elektro
Fakultas Teknik
Universitas Hasanuddin

Dr. Ir. H. Andani Ahmad, MT
NIP : 19601211 198703 1 022

ABSTRAK

Penggunaan *Programmable Logic Control (PLC)* sebagai sistem kendali pada dunia industri dewasa ini memegang peranan penting karena kemudahan penggunaannya dan efektifitas biaya yang dikeluarkan. PLC juga merupakan sistem pengendali yang tergolong *safety* karena didalamnya tersedia sistem pengontrolan dan monitoring sehingga seorang operator tidak perlu lagi ke lapangan untuk melakukan hal tersebut.

Kendalian Servo Posisi dan Kecepatan menggunakan Motor dari *DC Servo Trainer* tipe ED-4400B buatan *ED Co., Ltd.* menggunakan parameter tegangan untuk pengendaliannya. Kendalian ini dipakai dalam tugas akhir sebagai bentuk simulasi pengontrolan dari dunia industri sehingga dapat terus dikembangkan. Maka, PLC berguna untuk mengontrol modul praktikum tersebut sehingga dapat dikendalikan sekaligus dimonitoring.

Pengujian dilakukan dengan membandingkan tegangan masukan dan keluaran secara teori dan praktek dari kendalian dan membuat presentase kesalahan dari data tersebut. Dilakukan juga pengujian melihat parameter dari alat ukur praktikum dengan tampilan pada antarmuka sehingga dapat pula dibuat presentase kesalahannya sehingga simulasi dapat mendekati kenyataan . Pada Pengendali Posisi presentase kesalahan berkisar antara 2.24% sampai dengan 22.22%. Sedangkan untuk pengontrolan kecepatan presentase kesalahan berkisar antara 0.67% sampai dengan 2.92%.

Kata kunci : *Programmable Logic Control*, Antarmuka, Servo Posisi, Kontrol Kecepatan, *PLC-5*, *RSLogix-5*, *RSView32*.

KATA PENGANTAR

Syukur alhamdulillah saya panjatkan kehadirat ALLAH SWT, karna atas kehendak-Nya lah tugas akhir ini dapat saya selesaikan dengan baik. Tugas akhir ini merupakan salah satu syarat untuk menyelesaikan studi pada Jurusan Elektro Fakultas Teknik Universitas Hasanuddin.

Meskipun banyak hambatan maupun tantangan yang saya alami selama penyusunan tugas akhir ini yang berjudul "**Sistem Kendali Servo Posisi Dan Kecepatan Dengan *Programmable Logic Controller (PLC)"*** ini**,** namun berkat bantuan dan kerjasama berbagai pihak, akhirnya saya dapat mengatasi hambatan dan tantangan tersebut. Untuk semua itu, pada kesempatan ini saya dengan tulus mengucapkan terima kasih sebesar-besarnya kepada :

1. **Bapak Dr. Ir. H. Andani Ahmad, MT**, selaku pembimbing pertama dan **Bapak Ir. Christoforus Y, MT** selaku pembimbing kedua yang telah berkenan memberikan bimbingan, perhatian, saran serta pengarahan sejak awal penyusunan hingga akhir penulisan tugas akhir ini.
2. **Bapak Dr. Ir. H. Andani Ahmad, MT,** selaku ketua Jurusan Elektro Fakultas Teknik Universitas Hasanuddin, bapak dan ibu dosen, serta seluruh staf dan karyawan Jurusan Elektro yang telah banyak membantu selama masa perkuliahan di universitas ini.
3. **Ayah dan Ibu** tercinta atas doanya yang selalu memberikan nasehat dan menjadi penggugah semangat.

4. Teman – teman **D'Eleven** yang selalu membantu, menghibur dan memberi semangat hingga saat ini. Terkhusus kepada **Fina, Ade,** dan **Yunita** yang selalu menemani kapan pun dan di mana pun.
5. Rekan-rekan seperjuangan **Spyware 08** yang telah bersama-sama berjuang, seluruh kenangan akan teringat dalam "nyanyian sahabat"..
6. Saudara-saudaraku **se-Elektro** dan **se-Teknik**, kanda-kanda senior, adik-adik di Jurusan Elektro Fakultas Teknik Universitas Hasanuddin yang selalu membantu dan memberi semangat dan kepada semua pihak yang telah membantu dan tidak sempat disebutkan satu-persatu.

Penulis telah berusaha maksimal mungkin agar tugas akhir ini dapat terselesaikan sesuai dengan harapan, namun keterbatasan kemampuan sehingga tugas akhir ini tampil dengan segala kekurangannya. Oleh karena itu, saya senantiasa membuka diri terhadap saran dan kritik yang bertujuan untuk penyempurnaan tugas akhir ini.

Makassar, 10 Februari 2012

Penulis

DAFTAR ISI

Halaman

HALAMAN SAMPUL i

LEMBAR PENGESAHAN ii

ABSTRAK iii

KATA PENGANTAR iv

DAFTAR ISI vi

DAFTAR GAMBAR ix

DAFTAR TABEL xii

BAB I PENDAHULUAN 1

I.1 Latar Belakang 1

I.2 Rumusan Masalah 2

I.3 Maksud dan Tujuan Penelitian 2

I.4 Bataasan Masalah 3

I.5 Metode Penelitian 3

I.6 Sistematika Penulisan 4

BAB II TEORI DASAR 6

II.1. *Programmable Logic Controller (PLC)* *6*

II.1.1. *Hardware PLC-5 Allen Bradley* 11

II.1.1.1. Modul Analog *Input 1771-IFE Series C* 12

II.1.1.2. Modul Analog *Output 1771-OFE Series B* 14

II.1.2. *Software* 16

II.1.2.1. *RS Logix-5* 17

II.1.2.1.1. Instruksi *Bit* 18
II.1.2.1.2. Instruksi *Timer/Counter* 19
II.1.2.1.3. Instruksi Matematika 20
II.1.2.1.4. Instruksi *Compare* 22
II.1.2.1.5. Instruksi *Mov/ Logical* 24
II.1.2.1.6. Instruksi *Input/ Output* 24
II.1.2.1.7 Instruksi *Program Control* 26
II.1.2.2. *RS View32* 26
II.1.2.2.1 *Node window* 27
II.1.2.2.2 *Tag Database* 28
II.1.2.2.3 *Display Window* 29
II.2. Motor DC Servo Trainer 30
II.2.1. Pengendali Sudut Dengan Loop Tertutup 30
II.2.2. Kecepatan Motor dan Karakteristik Input 32
II.3. Rangkaian Pembagi Tegangan 33
BAB III PERANCANGAN SISTEM KENDALI 37
III.1 Perangkat Keras Kendalian 38
III.1.1 Rangkaian Pengendali Servo Posisi 39
III.1.2 Rangkaian Pengendali Kecepatan Motor 41
III. 2. Perancangan dan desain sistem antarmuka 43
III. 2. 1. Perancangan *Ladder diagram* 43
III.2.1.1 Program Pembacaan masukan dan keluaran analog 43
III.2.1.2 Pengontrolan Servo Posisi 45

III.2.1.3 Pengontrolan Kecepatan Motor 48
III. 2 . 2 Perancangan Antarmuka 55
III. 2. 2. 1. Pembuatan *Tag Database* 55
III. 2. 2. 2. Perancangan jendela *Main Menu* 57
III. 2. 2. 3. Perancangan jendela Servo Position 58
III. 2. 2. 4. Perancangan jendela *Motor Speed* 59
IV. PENGUJIAN DAN ANALISIS 63
IV.1 Pengujian tegangan masukan dan keluaran 63
IV.1.1 Pegaturan Servo Posisi 63
IV.1.2 Pengontrolan Kecepatan Motor 66
IV.2. Perbandingan Parameter Antarmuka dengan Alat Praktikum 68
IV.2.1 Pengaturan Sudut Posisi 69
IV.2.2 Pengontrolan Kecepatan Motor 70
V. PENUTUP 73
V.1 Kesimpulan 73
V.2 Saran 74
DAFTAR PUSTAKA
LAMPIRAN A : Ladder Diagram Program
LAMPIRAN B : *DC Servo Trainer* tipe ED-4400B
LAMPIRAN C : Dokumentasi program
LAMPIRAN D : Makalah seminar hasil

DAFTAR GAMBAR

Gambar II.1. Sebuah *Programmable Logic Control* 7

Gambar II.2 Sistem PLC` 8

Gambar II.3. Sinyal: (a) diskrit, (b) digital, (c) analog 10

Gambar II.4. Pilihan plug pada modul analog masukan 13

Gambar II.5. Komunikasi antara prosesor dan modul masukan analog 14

Gambar II.6. Komunikasi antara prosesor dan modul 16

Gambar II.7. Tampilan awal *RSLogix5* 17

Gambar II.8. Simbol XIO 18

Gambar II.9. Simbol XIC 18

Gambar II.10. Simbol OTE 19

Gambar II.11 Simbol ONS 19

Gambar II.12 Tampilan instruksi TON 20

Gambar II.13 Tampilan instruksi ADD 20

Gambar II.14 Tampilan isntruksi SUB 21

Gambar II.15 Tampilan Instruksi MUL 21

Gambar II.16. Tampilan Instruksi DIV 22

Gambar II.17. Tampilan instruksi EQU 22

Gambar II.18. Tampilan instruksi GEQ 23

Gambar II.19. Tampilan instruksi LEQ 23

Gambar II.20. Tampilan instruksi MOV 24

Gambar II.21. Tampilan instruksi BTR 25

Gambar II.22. Tampilan instruksi BTW 25

Gambar II.23. Tampilan instruksi MCR 26
Gambar II.24. Tampilan jendela *Node* 27
Gambar II.25 Tampilan jendela *Tag Database* 28
Gambar II.26. Tampilan jendela *Display* 30
Gambar II.27. Sebuah loop tertutup pengontrol posisi servo 31
Gambar II.28. Hubungan antara kecepatan motor dan tegangan masuk 32
Gambar II. 29 Rangkaian ekivalen pengontrolan motor 33
Gambar II. 30. Rangkaian pembagi tegangan 34
Gambar II. 31. Resistor sederhana pembagi tegangan 35
Gambar III.1 Diagram bagan kotak pengendali motor 37
Gambar III.2 Flowchart Rangkaian pengendali sudut posisi 39
Gambar III.3 Rangkaian Pengendali Servo Posisi 40
Gambar III.4 Flowchart Rangkaian pengendali kecepatan motor 41
Gambar III.5. Rangkaian pengendali kecepatan motor 42
Gambar III.6 Program pembacaan analog 43
Gambar III.7 Flowchart untuk servo posisi 45
Gambar III.8 Program untuk servo posisi 46
Gambar III.9 Flowchart untuk servo posisi 48
Gambar III.10 Program pengontrolan kecepatan motor bagian 1 50
Gambar III.11 Program pengontrolan kecepatan motor bagian 2 51
Gambar III.12 Program pengontrolan kecepatan motor bagian 3 52
Gambar III.13 Program pengontrolan kecepatan motor bagian 4 54
Gambar III.14 Database label yang dipakai 56

Gambar III.15 Jendela *Main Menu* 57

Gambar III.16 Jendela *Servo Position* 58

Gambar III.17 Jendela *Motor Speed* 60

Gambar IV.1. Grafik perbandingan tegangan masukan servo posisi 66

Gambar IV.2. Grafik perbandingan tegangan keluaran servo posisi 66

Gambar IV.3. Perbandingan Tegangan Masukan Kecepatan Motor 68

Gambar IV.4 Grafik perbandingan sudut pengontrolan sudut posisi 69

Gambar IV.5 Grafik perbandingan RPM pengontrolan kecepatan motor ... 70

DAFTAR TABEL

Tabel II.1 Pilihan Jarak Masukan 13

Tabel II.2 Pilihan Jarak Keluaran 15

Tabel III.1 Berkas label *Position* 56

Tabel III.2 Berkas label *Speed* 57

Tabel IV.1 Tegangan masukan servo posisi 64

Tabel IV.2 Tegangan Keluaran Servo Posisi 65

Tabel IV.3 Tegangan masukan Kecepatan Motor 67

Tabel IV.4 Derajat sudut pengontrolan posisi sudut 69

Tabel IV.5 Perbandingan RPM pada pengontrolan kecepatan motor 71

BAB I
PENDAHULUAN

I.1. Latar Belakang Masalah

Kemajuan teknologi telah membuat segala sesuatu menjadi lebih praktis, demikian juga dalam kegiatan sehari-hari, kebutuhan industri menginginkan hal yang demikian. Berbagai inovasi telah dibuat untuk mempermudah pekerjaan manusia.

Sistem kontrol saat ini kebanyakan masih menggunakan kontrol lokal, dimana seorang operator harus kelapangan untuk mengoprasikan mesin agar dapat bekerja. Begitu pula jika operator tersebut ingin memonitor status dari mesin tersebut, seorang operator pun harus ke lapangan untuk melakukan hal tersebut. Sekarang ini hal tersebut kurang efisien karena memerlukan waktu yang lama dan tidak bisa terus menerus. Dan bahkan juga hal tersebut tidak diperbolehkan jika operator tersebut meninjau ke lapangan dengan pertimbangan *safety* dari seorang operator tersebut dimana kondisi lingkungan yang panas dan bising membuat kenyamanan untuk bekerja menjadi kurang bahkan keselamatan diri menjadi sangat berbahaya.

Dengan melihat pertimbangan tersebut, dalam dunia industri dapat memanfaatkan teknologi pengendali yang ada untuk meningkatkan keselamatan kerja dan juga efisiensi satu pekerjaan. Banyak sistem manual tersebut dibuat menjadi otomatis untuk membuat pekerjaan tersebut lebih praktis. Seiring dengan perkembangan teknologi sistem kendali didunia industri, sistem pengontrolan dan monitoring mulai diambil alih oleh alat kendali untuk menggantikan pekerjaan

manual yang penuh resiko tersebut. Salah satunya adalah sistem pengendali dengan menggunakan *Programmable Logic Control* (PLC).

PLC ini dengan segala fasilitas didalamnya mampu menggantikan peran manusia untuk mengoprasikan kendalian dari jarak jauh dengan sistem otomatis. Dalam [3] dijelaskan bahwa PLC dapat memantau masukan-masukan maupun keluaran-keluaran sesuai dengan instruksi didalam program dan melaksakan aturan kontrol yang telah diprogram. Hal ini membuat pekerjaan lebih efisien dan efektif dari beberapa hal seperti ekonomi, *safety* dan tenaga kerja.

Berangkat dari hal tersebut maka kami membuat tugas akhir dengan judul **"Sistem Kendali Servo Posisi dan Kecepatan dengan *Programmable Logic Controller* (PLC)".**

I.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam tugas akhir ini yaitu, bagaimana merancang suatu sistem kendali berupa motor yang dapat di kontrol melalui suatu antarmuka dengan menggunakan PLC sehingga dapat di kontrol dan di monitoring setiap waktu.

I.3 Maksud dan Tujuan Penulisan

Berangkat dari latar belakang permasalaahn yang telah diuraikan di atas, kami bermaksud untuk mengembangkan sistem kendali posisi dan kecepatan motor dengan menggunakan PLC, perancangan *hardware* dan *software*, yang dapat digunakan untuk mengendalikan putaran motor tersebut.

Adapun tujuan dari penulisan tugas akhir ini adalah :

1. Memahami prinsip kerja PLC dan aplikasinya untuk mengendalikan sebuah motor dalam perangkat sistem kendali terkhusus untuk analog masukan dan analog keluaran.
2. Memahami cara pengelolaan parameter analog berupa sinyal listrik yang masuk ke PLC dan memanfaatkannya untuk membaca parameter yang ada.
3. Mengolah parameter yang masuk ke PLC untuk membuat sebuah antarmuka yang berfungsi sebagai pusat pengendali dan monitoring motor.
4. Untuk menyelesaikan studi pada Jurusan Elektro Fakultas teknik Universitas Hasanuddin dan mendapatkan gelar Sarjana Teknik.

I.4. Batasan Masalah

Untuk kemudahan dan lebih terperincinya pembahasan penulisan, permasalahan yang dibahas dalam tugas akhir ini dibatasi pada :

1. Untuk kendalian yang digunakan adalah sebuah Motor dari *DC Servo Trainer* tipe ED-4400B buatan *ED Co., Ltd.*
2. Sistem pengontrol yang digunakan adalah *Programmable Logic Control* jenis Allen Bradley tipe *PLC-5*.
3. *Software* yang digunakan adalah *RSLogix5* untuk pemograman dan *RSView32* untuk membuat suatu antarmuka.

I.5. Metode Penelitian

Dalam penyusunan tugas akhir ini ada beberapa metode yang akan kami gunakan yaitu:

1. Studi Literatur (*Library Research*). Yakni membaca dan mempelajari bahan kuliah, literatur-literatur, data sheet, dan tulisan-tulisan yang berkaitan dengan tugas akhir ini.
2. Merancang dan Membuat sistem secara *hardware.*
3. Merancang diagram ladder dengan menggunakan *RSLogix5* kemudian pembuatan antarmuka menggunakan *RSView 32*.
4. Menguji dan mengambil data dari perancangan.
5. Menganalisis hasil dan membuat kesimpulan.

I.6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini terbagi dalam lima bab dengan harapan maksud dan tujuan dari penulisan ini dapat terangkum seluruhnya. Pembagian bab tersebut adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang penguraian secara singkat latar belakang, tujuan, perumusan masalah, batasan masalah, batasan masalah, metodologi penulisan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini akan dijelaskan tentang teori penunjang yang digunakan dalam pembuatan proyek akhir ini. Teori tersebut antara lain mengenai PLC, sensor, dan motor.

BAB III PERANCANGAN SISTEM KENDALI KECEPATAN DAN POSISI MOTOR

Pembahasan mengenai perancangan dan prinsip kerja sistem yang akan dibuat, meliputi hardware dan software.

BAB IV PENGUJIAN DAN ANALISIS

Pembahasan mengenai implementasi sistem dan analisa hasil yang diperoleh.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari pembahasan permasalahan dan saran-saran untuk perbaikan dan penyempurnaan proyek akhir ini.

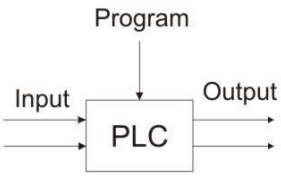
BAB II
TEORI DASAR

Bab ini akan diawali dengan pembahasan tentang teori dasar sistem kendali berbasis *Programmable Logic controller (*PLC*)*. Baik perangkat keras yang digunakan maupun perangkat lunak dari *PLC-5* dan teori dasar *Motor DC Servo Trainer* sebagai modul yang digunakan dalam proyek ini, serta teori dasar rangkaian pembagi tegangan sebagai perangkat yang modul dengan *PLC-5*.

II.1. *Programmable Logic Controller (PLC)*

Programmable Logic Control (singkatnya PLC) merupakan suatu bentuk khusus pengontrol berbasis mikroprosesor yang memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi-instruksi dan untuk maengimplementasikan fungsi-fungsi semisal logika, *sequencing*, pemwaktuan (*timing*), pencacah (*counting*) dan aritmatika guna mengontrol mesin-mesin dan proses-proses (Gambar II.1) dan dirancang untuk dioprasikan oleh para insinyur yang hanya memiliki sedikit pengetahuan mengenai komputer dan pemograman. Piranti ini dirancang sedemikian rupa agar tidak hanya programer komputer saja yang dapat membuat atau mengubah program-programnya. Oleh karena itu, para perancang PLC telah menempatkan sebuah program awal didalam piranti ini (*pre-program*) yang memungkinkan program-program kontrol dimasukkan dengan menggunakan suatu bentuk bahasa pemograman yang sederhana dan intuitif. Istlah logika (*logic*) dipergunakan karena pemograman yang harus dilakukan sebagian besar berkaitan dengan pengimplementasikan operasi-operasi logika dan penyambungan (*switching*), misalnya jika A atau B terjadi maka sambungkan

(atau hidupkan) C, jika A dan B terjadi maka sambungkan D. Perangkat-perangkat masukan, yaitu , sensor-sensor semisal saklar, dan perangkat-perangkat keluaran didalam sistenm yang dikontrol, misalnya, motor, katub, dsb., disambungkan ke PLC. Sang operator kemudian memasukkan serangkai instruksi, yaitu, sebuah program, kedalam memori PLC. Perangkat pengontrol tersebut kemudian memantau masukan-masukan dan keluaran-keluaran sesuai dengan instruksi-instruksi didalam program dan melaksanakan aturan-aturan kontrol yang telah diprogram.



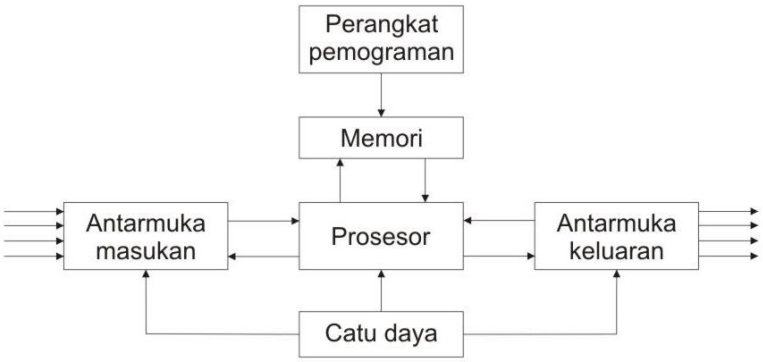
Gambar II.1. Sebuah *Programmable Logic Control*

PLC memiliki keunggulan yang signifikan, karena sebuah perangkat pengontrol yang sama dapat digunakan didalam beraneka ragam sistem kontrol. Untuk memodifikasi sebuah sistem kontrol dan aturan-aturan pengontrolan yang dijalankan, yang harus dilakukan oleh sorang operator hanyalah memasukkan seperangkat instruksi yang berbeda dari yang digunakan sebelumnya. Penggantian rangkaian kontrol tidak perlu dilakukan. Hasilnya adalah sebuah perangkat yang fleksibel dan hemat-biaya yang dapat dipergunakan didalam sistem-sistem kontrol yang sifat dan kompleksitasnya sangat beragam.

PLC serupa dengan komputer, bedanya : komputer dioptimalkan untuk tugas-tugas perhitungan dan penyajian data, sedangkan PLC dioptimalkan untuk

tugas-tugas pengontrolan dan pengoprasian didalam lingkungan industri. Dengan demikian PLC memiliki karakteristik:

1. Kokoh dan dirancang untuk tahan terhadap getaran, suhu, kelembaban dan kebisingan.
2. Antarmuka untuk masukan dan keluaran telah tersedia secara *built-in* didalamnya.
3. Mudah diprogram dan menggunakan sebuah bahasa pemograman yang mudah dipaham, yang sebagian besar berkaitan dengan operasi-operasi logika dan penyambungan.



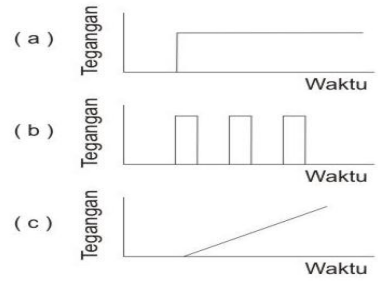
Gambar II.2 Sistem PLC

Perangkat PLC pertama dikembangkan pada tahun 1969. Dewasa ini PLC secara luas digunakan dan telah dikembangkan dari unit-unti kecil yang berdiri sendiri (*self-contained*) yang hanya mampu menangani sekitar 20 masukan/keluaran menjadi sistem-sistem modular yang dapat menangani masukan/keluaran dalam jumlah besar, menangani masukan/keluaran analog maupun digital, dan melaksanakan mode-mode kontrol proposional-integral-derivatif.

Umumnya, sebuah sistem PLC memiliki lima komponen dasar. Komponen-komponen ini adalah unit prosesor, memori, unit catu daya, bagian antarmuka masukan/keluaran, dan perangkat pemograman. Gambar II.2 menampilkan konfigurasi dasarnya.

1. *Unit prosesor* atau *central processing unit* (unit pengolahan pusat)(*CPU*) adalah unit yang berisi mikroprosesor yang menginterpretasikan sinyal-sinyal masukan dan melaksanakan tindakan-tindakan pengontrolan, sesuai dengan program yang tersimpan dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal-sinyal kontrol ke antarmuka keluaran.
2. *Unit catu daya* diperlukan untuk mengkonversikan tegangan AC sumber menjadi tegangan rendah DC (5 Volt) yang dibutuhkan oleh prosesor dan rangkaian-rangkaian didalam modul-modul antarmuka masukan dan keluaran.
3. *Peangkat pemograman* dipergunakan untuk memasukkan program yang dibutuhkan didalam memori. Program tersebut dibuat dengan menggunakan perangkat ini dan kemudian dipindahkan kedalam unit memori PLC.
4. *Unit memori* adalah tempat dimana program yang digunakan untuk melaksanakan tindakan-tindakan pengontrolan oleh mikroprosesor disimpan.
5. *Bagian masukan dan keluaran* adalah antarmuka dimana prosesor menerima informasi dari dan mengkomunikasikan informasi kontrol ke

perangkat-perangkat eksternal. Sinyal-sinyal masukan, oleh karenanya, dapat berasal dari saklar-saklar pada kasus mesin bor otomatis, atau sensor-sensor lain, seperti misalnya sel-sel fotoelektris pada mekanisme perhitungan, sensor suhu atau sensor aliran cairan, dsb. Sinyal-sinyal keluaran mungkin diberikan pada kumparan-kumparan *starter* motor, katup-katup selenoida, dll. Perangkat-perangkat masukan dan keluaran dapat digolongkan menjadi perangkat-perangkat yang menghasilkan sinyal diskrit atau digital, dan yang menghasilkan sinyal-sinyal analog (Gambar II.3). Perangkat-perangkat yang menghasilkan sinyal-sinyal digital adalah perangkat-perangkat yang hanya mengindikasikan kondisi ‘mati’ (*off*) atau ‘hidup’(*on*). Sehingga, saklar adalah sebuah perangkat yang menghasilkan sebuah sinyal diskrit, yaitu, ada tegangan atau tidak ada tegangan. Perangkat-perangkat digital pada dasarnya dapat dipandang sebagai perangkat-perangkat diskrit yang menghasilkan serangkai sinyal ‘mati;-‘hidup’. Perangkat-perangkat analog menghasilkan sinyal-sinyal yang amplitudonya sebanding dengan nilai variable yang dipantau. Sebagai contoh, sensor suhu akan menghasilkan tegangan yang nilainya sebanding dengan suhu.



Gambar II.3. Sinyal: (a) diskrit, (b) digital, (c) analog

PLC yang digunakan dalam tugas akhir ini adalah *PLC-5 Allen Bradley* yang penjelasannya dapat di bagi menjadi dua bagian yaitu perangkat keras *(hardware)* dan perangkat lunak *(software)*.

II.1.1. *Hardware PLC-5 Allen Bradley*

Bagian –bagian perangkat keras dari *PLC-5 Allen Bradley* :

- *Chasis I/O*, adalah tempat atau *chasing* dimana modul *input* (masukan) dan modul *output* (keluaran) berada.
- Prosesor *PLC-5/80E*, komponen ini merupakan segmen pengolah dari keseluruhan bagian dan unsur-unsur yang terlibat dalam sistem pengontrolan.
- Modul *I/O*, yaitu piranti masukan dan keluaran (modul masukan dan keluaran) yang b€erfungsi untuk menghubungkan prosesor dengan masukan-masukan yang diperoleh dari peralatan kontrol seperti saklar otomatis, saklar manual, maupun masukan analog, dan lain-lain. Kemudian oleh modul keluaran hasil olahan prosesor diterima dan dihubungkan dengan peralatan keluaran agar pengontrolan yang diinginkan dapat dilakukan sesuai dengan isi pemrograman.
- *Power Supply*, yaitu sumber yang digunakan untuk mencatu daya keseluruhan rangkaian PLC mulai dari prosesor sampai ke modul masukan dan keluaran.
- *Programming terminal*, yaitu terminal program atau tempat penyimpanan sementara program yang akan dikirim ke prosesor.
- *Communication Card*.

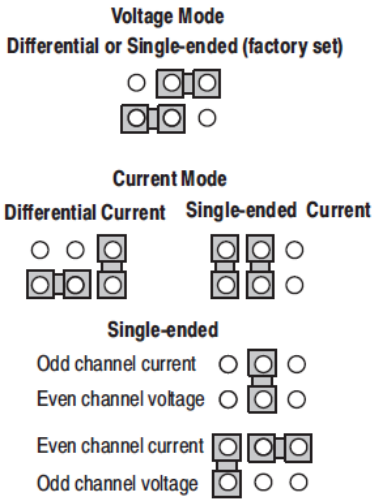
- Kabel *Programming terminal*.
- Kabel *Ethernet*.

Prosesor *PLC-5* adalah modul slot tunggal yang ditempatkan pada bagian paling kiri slot 1771 *I/O chasis.* Prosesor *PLC-5* memiliki ruang yang besar untuk memori *I/O* dan kapabilitas yang tinggi dalam komunikasi.

II.1.1.1. Modul Analog *Input 1771-IFE Series C*

Modul analog *input* (masukan) adalah modul yang menghubungkan antara masukan sinyal analog dengan sinyal prosesor *PLC-5 Allen Bradley*. Modul analog *input* merupakan *single-slot* modul dan tidak memerlukan *power supply* tambahan. Modul menerima *power supply* pada *blackplane* dari 1771 *I/O* sebesar 500mA. Arus ini menyuplai seluruh modul yang terdapat pada *I/O chasis*. Setelah sinyal analog masuk pada modul, data *input* dikonversikan ke bentuk format digital untuk dikirim ke prosesor.

Modul analog masukan terdiri dari 16 *channel single ended* atau 8 *channel differential* (pada simulasi ini digunakan 8 *channel differential*) dan mengubah sinyal masukan ke bentuk nilai integer. Setiap masukan dikonfigurasikan sebagai arus atau tegangan masukan, adapun pilihan jarak masukannya dapat dilihat pada Tabel II.1. Modul ini (1771-IFE/C) mengkonfigurasikan tegangan dan arus ke dalam jumper pada modul tegangan sinyal masukannya. Bisa dikonfigurasi menggunakan *single ended* atau *diffenetial*.



Gambar II.4. Pilihan plug pada modul analog masukan

Tabel II.1 Pilihan Jarak Masukan

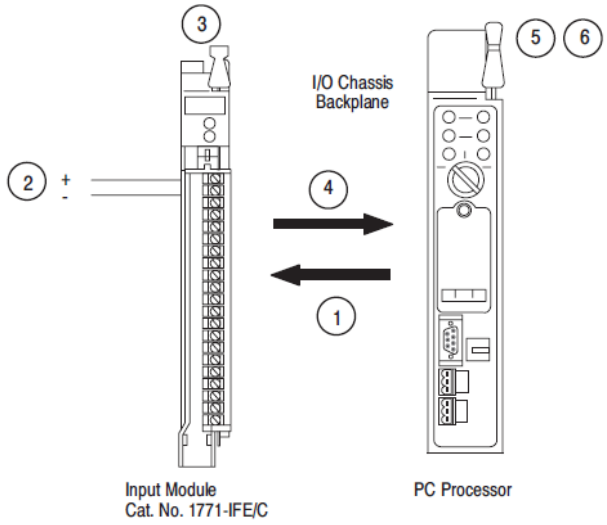
Tegangan (Vdc)	Arus (mA)
1 - 5 0 – 5 -5 – (+5) -10 – (+10)	4 – 20 0 – 20 -20 - (+20)

Namun pada proyek ini yang digunakan adalah jarak keluaran berupa tegangan antara 0-5 Volt.

Pemindahan data dari modul ke prosesor menggunakan instruksi BTW dan BTR yang terdapat pada *ladder diagram*. Cara kerja instruksi ini adalah :

1. Menggunakan peralatan yang menghasilkan sinyal analog untuk dihubungkan pada *channel* modul.
2. Modul mengubah sinyal analog ke bilangan integer dan menyimpan nilai-nilai tersebut hingga prosesor memintanya.
3. Ketika diinstruksikan oleh ladder, prosesor membentuk BTR untuk membaca nilai-nilai masukan yang terdapat pada modul.

4. Instruksi BTW digunakan untuk menulis nilai-nilai masukan dari modul ke memori prosesor.
5. Pada proses pemindahan ini dipastikan sudah tidak ada lagi *error*.



Gambar II.5. Komunikasi antara prosesor dan modul masukan analog

II.1.1.2. Modul Analog *Output 1771-OFE Series B*

Modul analog output (keluaran) adalah sebuah modul *intelligent block transfer* yang mengubah besaran biner atau nilai empat digit BCD (yang berasal dari prosesor) ke sinyal analog pada ke empat keluaran modul. Modul akan mencapai dengan transfer data dengan pemograman *block transfer*.

Pemograman *Block Transfer Write* (BTW) memindahkan 13 *words* data dari prosesor ke modul untuk konversi digital ke analog (D/A) pada sekali *program scan*. Informasi ini dikonversi ke sinyal analog dan akan dikirim ke saluran keluaran yang tepat. Sebuah *Block Transfer Read* (BTR) memindahkan lima *words* data dari modul ke data tabel prosesor, jika diperlukan, untuk tujuan pemeriksaan .

Modul ini mempunya fitur *scaling* yang mengubah data yang dikirim ke modul dalam unit modul ke sinyal analog yang tepat. Ini bisa dihubungkan dengan empat alat keluaran analog, seperti *valve positioners*, kontrol kecepatan motor, konversi sinyal atau perekam, ke empat *channel* modul keluaran analog. Semua alat *input* dari keluaran analog harus dipastikan apakah yang digunakan adalah arus atau tegangan pada setiap *channel* modul keluaran.

Ada tiga versi dari modul keluaran analog :

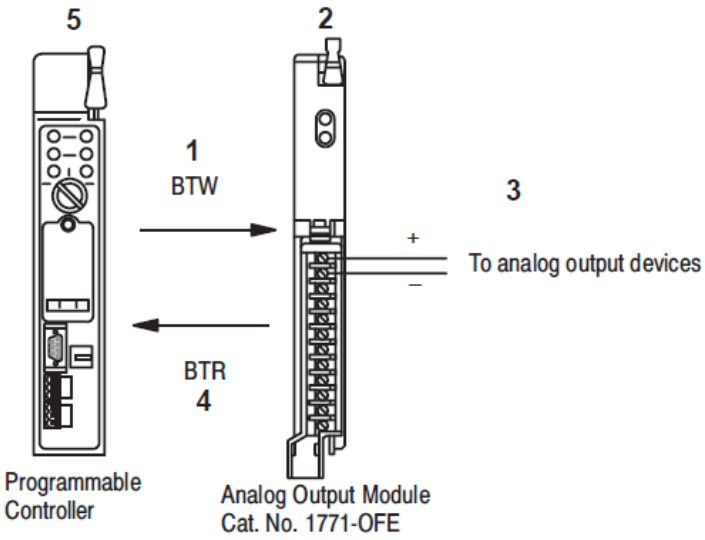
Tabel II.2 Pilihan Jarak Keluaran

Nomor Katalog	Modul keluaran	Jarak keluaran	Keterangan
1771-OFE1	Tegangan	1-5 V dc 0-10 V dc ± 10 V dc	Dipilih dengan konfigurasi jumper
1771-OFE2	Arus	4-20 mA	Pengaturan pabrik
1771-OFE3	Arus	0-50 mA	Pengaturan pabrik

Namun pada proyek ini yang digunakan adalah 1771-OFE2 dengan jarak keluaran berupa arus antara 4-20mA.

Komunikasi Modul Analog

Transfer data dari prosesor ke modul (*Block Transfer Write*) dan dari modul (*Block Transfer Read*) menggunakan instruksi BTW dan BTR pada perangkat lunak ladder diagram. Instruksi ini membuat prosesor mengirimkan nilai keluaran ke modul, membangun pilihan operasi modul (lihat ilustrsi dibawah) dan menerima status informasi dari modul.



Gambar II.6. Komunikasi antara prosesor dan modul

1. Prosesor mentransfer konfigurasi dan data output ke modul melalui instruksi *Block Transfer Write*.
2. Modul mengubah data menjadi proporsi keluaran tegangan atau arus.
3. Modul keluaran ini menuju alat eksternal analog.
4. Ketika selesai instruksi dari program ladder, prosesor akan membaca *block transfer* dari nilai keluaran dan ststus modul.
5. Prosesor dan modul akan menentukan transfer yang dibuat tanpa kesalahan.
6. Program ladder diagram bisa digunakan dan atau pindah (jika valid) sebelum ditulis dengan memindahkan data baru pada subsequent transfer.

II.1.2. *Software*

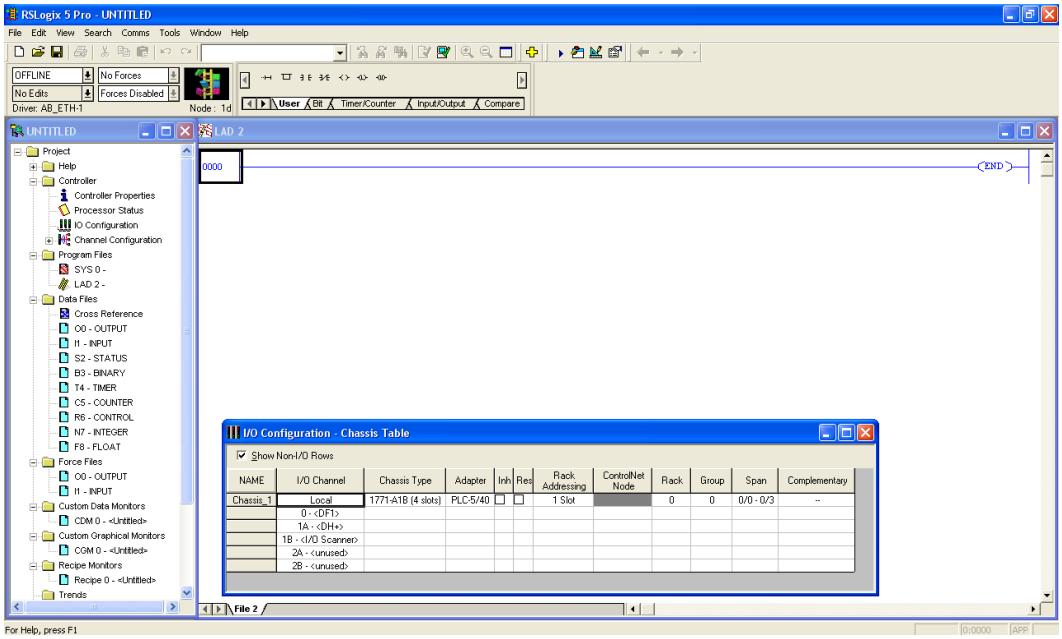
Software atau perangkat lunak menjadi bagian penting dalam merencanakan sebuah sistem kendali berbasis PLC, dimana setiap instruksi keja yang hendak dilaksanakan terlebih dahulu harus melalui perancangan secara

perangkat lunak, sebelum dimasukkan ke dalam peralatan itu sendiri. Dalam hal ini terdapat berbagai macam tipe perangkat lunak yang digunakan dalam pemrogramannya. Untuk sistem berbasis *PLC -5 Allen Breadley* ini menggunakan *RSLogix-5*.

II.1.2.1. *RS Logix-5*

RSLogix-5 adalah perangkat lunak berbasis pada grafis sehingga memudahkan setiap penggunanya dalam merencanakan program PLC yang diinginkan dengan menggunakan teknologi *Human Machine Interface*, sehingga pengguna dapat mengakses instruksi hanya dengan metode *klik* dan *drag*.

Perangkat lunak ini juga terdiri dari beberapa bagian seperti *menu bar, wordsheet, sub menu,* serta *toolbar*.



Gambar II.7. Tampilan awal *RSLogic5*

II.1.2.1.1. Instruksi *Bit*

Instruksi mengoperasikan satuan *bit* per data. Selama pengoperasian, prosesor men-*set* atau me-*reset bit* berdasarkan logika kontinuitas dari *rung ladder*. Adapun instruksi *bit* ini terdiri dari instruksi-instruksi antara lain :

- ***Examine if Open*** **(XIO)**

  Operasi dari instruksi XIO memiliki alamat *file* data masukan. Ketika ada masukan dari divais eksternal pada suatu rangkaian, keadaan *"off"* menunjukkan bahwa terminal masukan terhubung pada divais.

Gambar II.8. Simbol XIO

- ***Examine if Closed*** **(XIC)**

  Operasi dari instruksi XIC memiliki lamat *file* data masukan. Ketika ada masukan dari divais eksternal pada suatu rangkaian, keadaan *"on"* menunjukkan bahwa terminal masukan terhubung pada divais.

Gambar II.9. Simbol XIC

- ***Output Energize*** **(OTE)**

  Operasi dari instruksi OTE memiliki alamat *file* data keluaran, keadaan dari terminal keluaran ditampilkan pada *file* data keluaran pada alamat *bit* tertentu.

  Ketika proseor menemukan jalur logika yang *true* pada *rung* yang memuat instruksi OTE, maka *bit set* (1). Perubahan ini membuat terminal

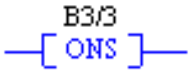
keluaran *"on"* dan mengakibatkan divais keluaran terhubung pada terminal. Kemudian ketika jalur *true* tidak lagi ditemukan, maka prosesor akan secara otomatis me-*reset* (0) atau membuat terminal *"off"* dan tidak lagi membangkitkan terminal keluaran.

Gambar II.10. Simbol OTE

- ***One-Shot*** **(ONS)**

  Instruksi ONS adalah suatu instruksi masukan yang membuat ladder bernilai *"on"* untuk satu program *scan* pada transisi dari *false-to-true* dengan kondisi sebelum instruksi ONS di *rung* yang sama.

  Biasanya instruksi ONS untuk memulai peristiwa yang dipicu oleh tombol tekan, seperti penghentian secara cepat nilai apa yang ditampilkan oleh LED.



Gambar II.11 Simbol ONS

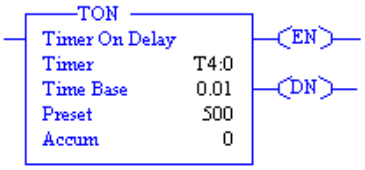
II.1.2.1.2. Instruksi *Timer/Counter*

- ***Timer on Delay*** **(TON)**

  Gunakan instruksi TON untuk membuat keluaran "*ON*" atau "*OFF*" setelah *timer* telah selesai dengan waktu intervalnya. Instruksi keluaran ini dimulai dengan waktu (antara interval satu detik atau seperseratus detik)

ketika *rung* berubah menjadi "*true*" dan berlanjut hingga salah satu dibawah ini terjadi :

- Akumulasi nilai sama dengan waktu intervalnya
- *Rung* berubah menjadi "*false*"
- Instruksi reset menyetel ulang *timer*
- Prosesor telah menyetel ulang nilai akumulasi ketika *rung* dalam kondisi *false*, walaupun *timer* telah selesai ataupun tidak

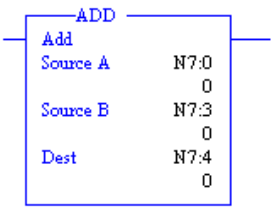


Gambar II.12 Tampilan instruksi TON

II.1.2.1.3. Instruksi Matematika

- ***Add* (ADD)**

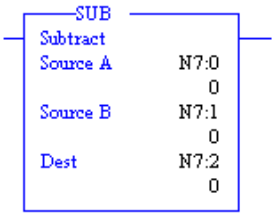
Simbol instruksi ini menjumlahkan antara suatu nilai (*source A*) dengan nilai lainnya (*source B*) dan menempatkan hasilnya pada alamat yang dituliskan pada *Dest* (*Destination*). Baik nilai *source A* maupun *source B* akan menunjuk alamat yang mempunyai data pengganti nilai *source A* dan nilai *source B*.



Gambar II.13 Tampilan instruksi ADD

- ***Subtract* (SUB)**

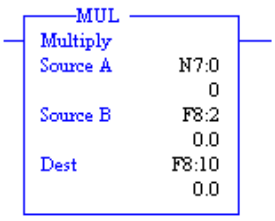
  Simbol instruksi ini mengurangkan antara suatu nilai (*source A*) dengan nilai lainnya (*source B*) dan menempatkan hasilnya pada alamat yang dituliskan pada *Dest* (*Destination*). Baik nilai *source A* maupun *source B* akan menunjuk alamat yang mempunyai data pengganti nilai *source A* dan nilai *source B*.



Gambar II.14 Tampilan isntruksi SUB

- ***Multiply* (MUL)**

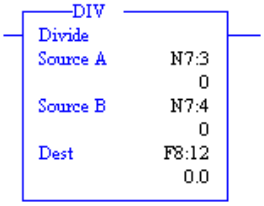
  Simbol instruksi ini mengalikan antara suatu nilai (*source A*) dengan nilai lainnya (*source B*) dan menempatkan hasilnya pada alamat yang dituliskan pada *Dest* (*Destination*). Baik nilai *source A* maupun *source B* akan menunjuk alamat yang mempunyai data pengganti nilai *source A* dan nilai *source B*.



Gambar II.15 Tampilan Instruksi MUL

- **Divide (DIV)**

  Instruksi DIV membasi (*Source A)* dengan (*Source B)* dan menyimpan hasilnya pada *Dest*. (*Source A*) dan (*Source B)* bisa berisikan alamat atupun konstanta. Instruksi ini akan mengeksekusi selama *rung* bernilai "*true*", untuk mengeksekusi sekali, kondisikan *rung* dengan instruksi ONS.

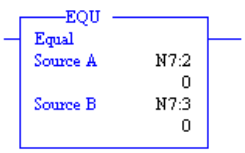


Gambar II.16. Tampilan Instruksi DIV

II.1.2.1.4. Instruksi *Compare*

Simbol ini mengoperasikan satuan *word* per data. Selama pengoperasian, prosesor men-*scan bit* berdasarkan logika *value* dari instruksi dan membaca memori sebagai refernsi tindakan. Adapun *word* ini terdiri dari instruksi-instruksi antara lain :

- ***Equal* (EQU)**



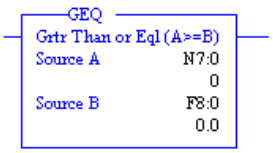
Gambar II.17. Tampilan instruksi EQU

Instruksi masukan ini membandingkan dua alamat dengan nilai yang spesifik. Jika nilainya sama, maka *rung* akan melanjutkannya. *Rung* akan

bernilai “*true*” dan keluaran akan *energized* (tergantung tidak ada instruksi lain pada *rung*)

- ***Greater Than or Equal* (GEQ)**

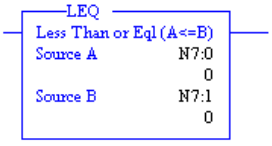
  Masukan instruksi ini akan membandingkan dua nilai yang telah ditentukan. Jika nilai yang tersimpan dalam *(Source A)* lebih besar dari atau sama dengan nilai yang disimpan dalam *(Source B)*, kondisi *rung* akan berlanjut. *Ladder* akan bernilai "*ON*" dan keluaran akan aktif (selama tidak ada instruksi lain yang mempengaruhi status *ladder*).



Gambar II.18. Tampilan instruksi GEQ

- ***Less Than or Equal* (LEQ)**

  Masukan instruksi ini akan membandingkan dua nilai yang telah ditentukan. Jika nilai yang tersimpan dalam *(Source A)*kurang dari atau sama dengan nilai yang disimpan dalam *(Source B)*, kondisi *rung* akan berlanjut. *Ladder* akan bernilai "*ON*" dan keluaran akan aktif (selama tidak ada instruksi lain yang mempengaruhi status *ladder*)



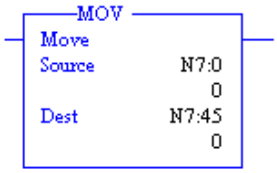
Gambar II.19. Tampilan instruksi LEQ

II.1.2.1.5. Instruksi *Mov/ Logical*

- ***Move* (MOV)**

  Instruksi MOV merupakan instruksi *output* yang akan menyalin nilai dari alamat sumber *(Source)* ke alamat tujuan *(Destination)*. Selama tidak ada instruksi lain yang mempengaruhi status *ladder*, instruksi data bergerak setiap *scan*.

  Instruksi ini membuat salinan asli dan menempatkan hasil salinan di lokasi baru (*Destination*). Nilai yang asli tetap utuh dan tidak berubah di lokasi sumber.



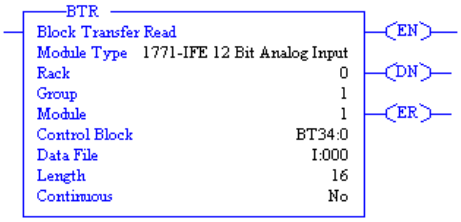
Gambar II.20. Tampilan instruksi MOV

II.1.2.1.6. Instruksi *Input/ Output*

- ***Block Transfer Read* (BTR)**

  Simbol instruksi ini bertujuan untuk mentransfer maksimal 64 *word* pada waktu yang bersamaan dari sebuah *block transfer module* pada satu *chasis I/O*, dimana pemilihan jenis modul yang digunakan ditempatkan pada *module type*, pemilihan nomor *rack* yang digunakan ditempatkan pada *rack no*, pemilihan nomor grup *I/O* yang men-spesifikasikan posisi dari modul yang digunakan di dalam *chasis* ditempatkan pada *group*, dan pemilihan nomor slot dalam *group* yang digunakan ditempatkan pada *module*.

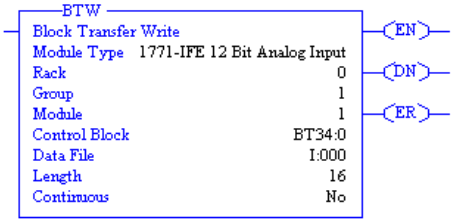
Adapun *control block* merupakan tempat untuk menuliskan tipe data *file control* yang digunakan, disarankan untuk menggunakan *file* data *block transfer* (BT), sedangkan *data file* merupakan tempat memasukkan alamat *input*, *output*, status, *integer*, *float*, *biner* atau *file* data *ASCII* darimana prosesor menulis atau ke mana prosesor membaca dan mentransfer data. Parameter lainnya yakni *length* merupakan tempat untuk memasukkan jumlah *word* yang akan ditransfer, yang mana panjangnya akan terkonfigurasi secara otomatis ketika *intelligent I/O* dikonfigurasi.



Gambar II.21. Tampilan instruksi BTR

- ***Block Transfer Write*** **(BTW)**

  Instruksi ini memiliki kesamaan yang sangat banyak dengan instruksi BTR. Perbedaannya adalah instruksi BTW digunakan untuk mentransfer data *output* dari prosesor ke modul *output*, sedangkan BTR digunakan untuk mentasnfer data *input* dari modul *input* ke prosesor



Gambar II.22. Tampilan instruksi BTW

II.1.2.1.7 Instruksi *Program Control*

- ***Master Control Reset* (MCR)**

  Instruksi keluaran ini (kadang dikenal sebagai "*Zone Control*") digunakan untuk mengatur area atau "*zones*" program *ladder* dimana semua keluaran yang ada didalamnya akan tidak aktif pada saat waktu yang sama dengan jarak waktunya. Ini dipakai berpasangan, satu MCR digunakan pada awal area *ladder* yang akan berpengaruh dan satu MCR digunakan pada akhir area tersebut.

  Instruksi masukan untuk memprogram *rung* pada MCR awal ke logika kontrol *rung* selanjutnya. Ketika *rung* berubah menjadi "*false*", semua keluaran yang termasuk dalam zona yang dikontrol akan tidak aktif. Ketika *rung* berubah menjadi "*true*", semua *rung* akan diperiksa termasuk ke kondisi normal *rung* (diluar dari zona kntrol instruksi)

Gambar II.23. Tampilan instruksi MCR

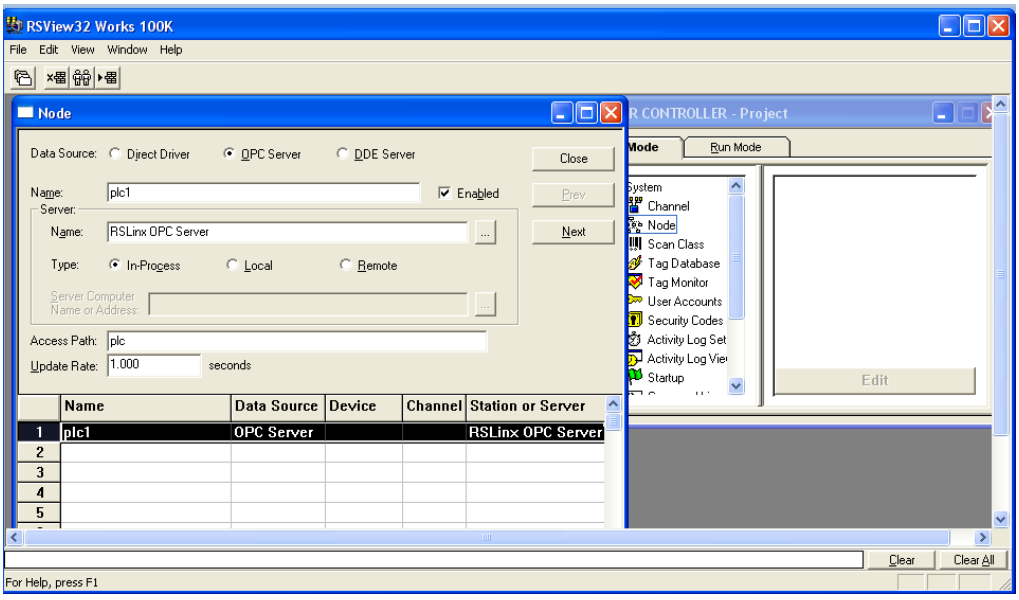
II.1.2.2. *RS View32*

RSView32 adalah sebuah perangkat lunak yang berfungsi untuk membuat sebuah antarmuka agar seluruh sistem pada PLC dapat dikendalikan dan dimonitor oleh antarmuka tersebut. Perangkat lunak ini dapat berkomunikasi dengan perangkat lunak pemogramman dari *Allen Breadley* (dalam hal ini *RSLogix 5*) sehingga setiap parameter dan kondisi yang berubah pada PLC dapat diketahui oleh antarmuka yang dibuat.

Penggunaannya pun sangat sederhana dan mudah dimengerti untuk membuat sebuah antarmuka. Karena didalamnya terdapat beberapa fitur untuk membuat sebuah *Node* yang bertujuan untuk merujuk ke sebuah PLC yang ingin dikendalikan, *Tag* yang berisikan sebuah alamat agar lebih mudah dikenali dan *Display* untuk pembuatan antarmuka sesuai yang kita inginkan.

II.1.2.2.1 *Node window*

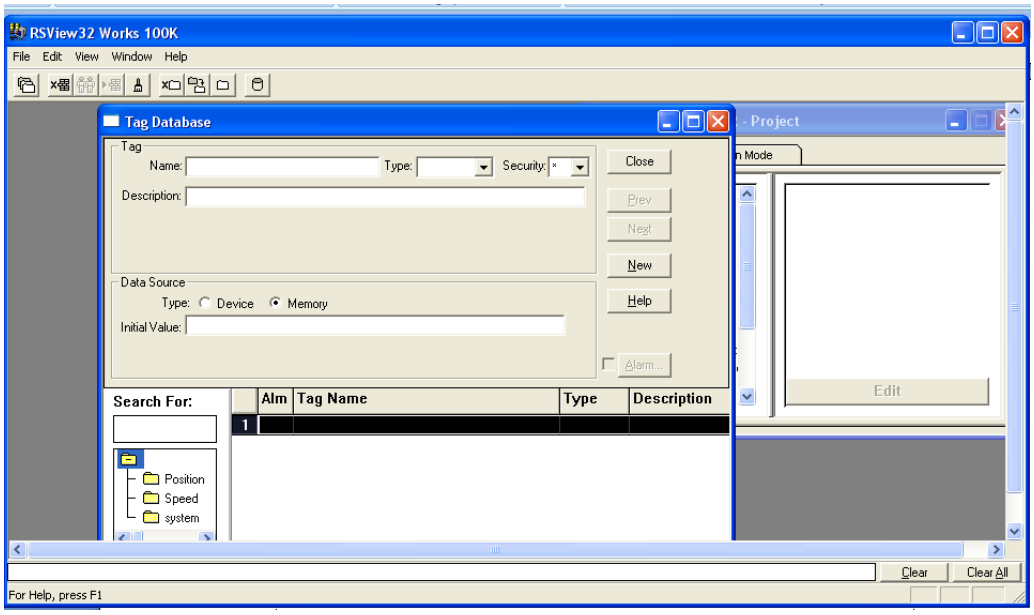
Tujuan untuk mengkonfigurasi awal node ini agar PLC yang akan dipakai selama proses pengontrolan akan dikenali. Ini sangat penting karena PLC selama bekerja akan selalu berkomunikasi dengan perangkat lunak ini melalui antarmuka yang telah dibuat. Maka perangkat lunak ini akan merujuk ke PLC tersebut dan akan terjadi proses interaksi apabila ada perubahan pada antarmuka maupun pada PLC.



Gambar II.24. Tampilan jendela *Node*

II.1.2.2.2 *Tag Database*

Tag atau pelabelan instruksi merupakan inti dari perangkat lunak antarmuka ini, karena kesalahan penggunaan *tagname* dapat mengakibatkan operasi program menjadi tidak normal dan bahkan mengalami kesalahan (*error*). Pada saat program dijalankan (*Runtime*), seluruh database pada objek grafis berisikan nilai-nilai saat antarmuka belum dijalankan, untuk itulah *tagname* digunakan agar informasi dari seluruh variabel yang telah dirancang dapat dikenali antarmuka sesuai dengan posisi kerjanya. Misalnya untuk *tagname* tipe masukan analog, antarmuka memerlukan informasi yang konkrit berapa integer pada PLC dan akan menampilkannya pada antarmuka yang telah dibuat.



Gambar II.25 Tampilan jendela *Tag Database*

Beberapa tipe label yang dapat dibuat adalah sebagai berikut ;

- *Digital tags*

Pelabelan digital akan menyimpan nilai numerik antara 0 atau 1. Label ini digunakan untuk mewakili alat yang mengondisikan dua keadaan: “ON”

atau “OFF” atau keadaan “*false*” atau “*true*”, seperti *switches*, kontak atau relay.

- *Analog tags*

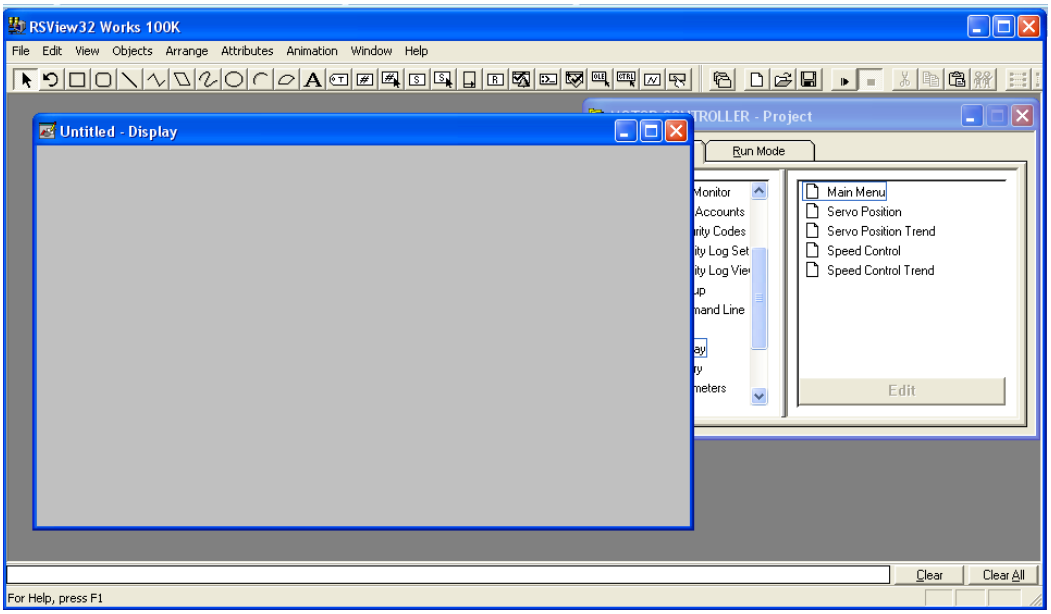
  Pelabelan analog akan menyimpan nilai numerik dari jarak nilai yang didefinisikan oleh label tersebut. Biasanya digunakan untuk mewakili alat yang mempunyai jarak nilai seperti tempratur, tekanan, *flow*, atau posisi kendali rotasi.

- *String tags*

  Pelabelan string akan menimpan ASCII string yang berupa karekter string atau keseluruhan kata. Maksimum string yang diperbolehkan yaitu 255 karekter.

II.1.2.2.3 *Display Window*

Pada jendela *Display* akan diberikan keleluasan kepada programmer agar membuat antarmuka sesuai yang diperlukan. Disini juga terdapat beberapa *tools* untuk pembuatan antarmuka tersebut dari pembuatan teks sampai ke pembuatan animasi. Objek tersebut akan berisikan label (*Tags*) yang telah dibuat di *Tag Database* sehingga setiap perubahan parameter akan terjadi pula di antarmuka tersebut. Setelah pembuatan antarmuka selesai, pada jendela ini juga dapat kita janlankan (*Run*) untuk melihat apakah animasi yang dibuat sesuai dengan perubahan parameter yang terjadi pada PLC atau tidak.



Gambar II.26. Tampilan jendela *Display*

II.2. Motor DC Servo Trainer

DC Servo Trainer ED-4400B dari ED Co., LTD. adalah sistem loop tertutup servo DC yang didesain untuk modul praktikum. Inti konsep pembuatan sistem trainer ED-4400B adalah untuk memberikan pengetahuan praktik kerja pada sistem loop tertutup servo DC kepada pengguna dengan mengintegrasikan dasar teori dan langkah demi langkah percobaan dengan satu subjek.

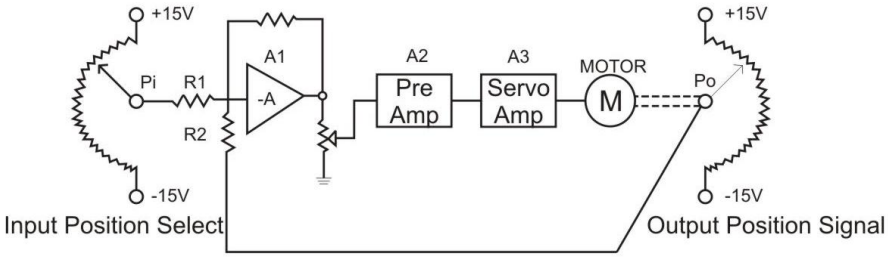
Modul praktikum pada ED-4400B sangat sederhana dan mudah untuk diuji cobakan pada sistem. Yang hanya dibutuhkan adalah untuk menghubungkan antara modul seperti yang diinstruksikan pada gambar rangkaian pada setiap seksi dengan menggunakan *jumper* yang telah disediakan.

II.2.1. Pengendali Sudut Dengan Loop Tertutup

Dalam sistem pengendali servo posisi, informasi posisi dari potensiometer yang dihubungkan pada motor menjadi umpan balik pada penguat kontrol *(control amplifier)*. Kemudian, pengaturan posisi masukan dari masukan potensiometer

dikombinasikan dengan sinyal balik pada *input amplifier* yang menjalankan motor pada bagian yang berbeda antara dua sinyal. Ketika kedua posisi teridentifikasi, keluaran pada amplifier akan menjadi nol.

Sebuah diagram sederhana dari sistem pengontrolan posisi putaran dapat dilihat pada gambar II.26.



Gambar II.27. Sebuah loop tertutup pengontrol posisi servo

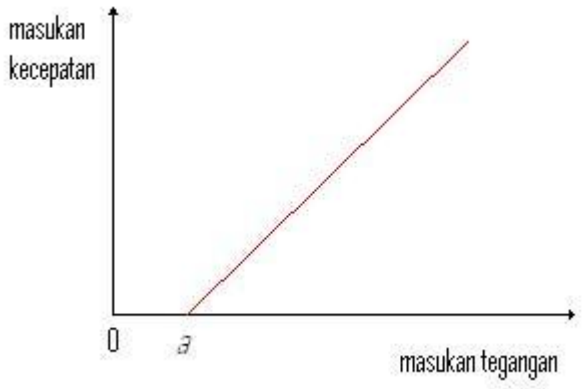
Terdapat tiga penguat pada gambar. A1 adalah sinyal dari generator, A2 adalah sinyal masukan penguat, dan A3 adalah *driver* untuk motor M. P1 berlawanan arah dengan P0. Perbedaan antara dua tegangan potensiometer menjadi masukan ke A1. Sinyal terus brlanjut melewati A2 dan A3, kemudian menjalankan motor dan tegangan pada P1 dan P0 saling mengurangi. Keadaan ini terus berlanjut sampai keluaran pada A1 sama dengan nol. Dalam hal ini, tegangan pada P1 dan P0 adalah sama tetapi dengan polaritas yang berbeda. Misalnya P1 = +3V dan P0 = -3V, inilah membuat keluaran pada A1 menjadi nol.

Posisi akhir dari P1 da P0 sangat relatif, bergantung pada ada tidaknya penambahan tegangan pada penguat. Untuk penambahan yang besar, posisi P0 hampir sama dengan P1, tetapi ketika penambahan tidak cukup maka posisi antara

keduanya tidak akan sama. Hal ini disebut sebagai kerugian pada pengontrolan posisi.

II.2.2. Kecepatan Motor dan Karakteristik Input

Secara umum motor merupakan sebuah mesin yang mengkonversi energi listrik menjadi energi gerak. Elemen kunci dari motor DC adalah medan (*field*) dan armatur. Aliran arus listrik yang melewati medan akan menimbulkan torka (tenaga putaran) pada armatur. Pada motor ED-4400B ini, efek medan dibangkitkan oleh magnet permanen. Magnet permanen ini menciptakan fluks magnet yang nilainya konstan. Sehingga kecepatan motor hanya bergantung pada tegangan masukan yang sampai pada armatur. Hubungan antara kecepatan motor dengan tegangan masukan dapat dilihat pada Gambar II.27.

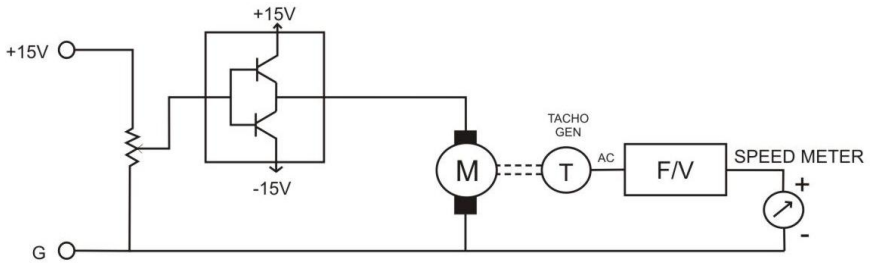


Gambar II.28. Hubungan antara kecepatan motor dan tegangan masuk

Pada Gambar II.23, *point a* terjadi karena motor membutuhkan tegangan minimum untuk mengatasi gesekan mesin dengan sikat-sikat, sampai akhirnya motor dapat berputar. Ketika tegangan masukan telah melebihi tegangan minimum (yang dibutuhkan), maka kecepatan motor akan naik secara linier sesuai

dengan bertambahnya tegangan masukan. Meskipun demikian, karakteristik linier akan berhenti pada saat terjadi saturasi. Hal ini disebabkan karena adanya batasan kapasitas bahan pada gulungan armatur, sehingga pada saat tegangan dinaikkan terus menerus maka arus yang dihasilkan pada armatur tidak akan berambah (*saturation*).

Motor ED-4400B *system* dijalankan dengan *motor driver amplifier U-154* dan *Attenuator U-151* sebagai pengontrol tegangan. Kecepatan motor dapat di deteksi dan dilihat langsung pada *output tacho U-159.*



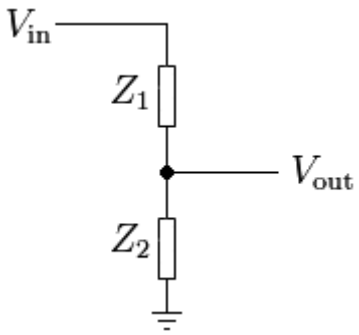
Gambar II. 29 Rangkaian ekivalen pengontrolan motor

II.3. Rangkaian Pembagi Tegangan

Pada elektronika, rangkaian pembagi tegangan (juga dikenal sebagai pembagi potensial) adalah rangkaian linear sederhana yang menghasilkan tegangan *output* (V_{out}) yang berasal dari tegangan input (V_{in}). Pembagi tegangan merujuk pada pembagian dari tegangan tergantung dari komponen yang pembaginya.

Rumus pembagi tegangan sama dengan rumus pembagi arus, namun rasio pembagi tegangan berasal dari impedansi tegangan yang ingin di cari, tidak seperti pembagi arus yang berasal dari komponen seberang yang akan dicari.

Contoh sederhana untuk rangkaian pembagi tegangan adalah terbagi atas dua resistor atau potensiometer yang dipasang seri. Pada umumnya ini digunakan untuk membuat tegangan referensi, atau untuk mendapatkan proporsi sinyal tegangan yang rendah ke tegangan yang akan terukur dan bisa juga digunakan sebagai sinyal attentuator dari frekuensi rendah.



Gambar II.30. Rangkaian pembagi tegangan

Rangkaian pembagi tegangan dapat dibuat dengan merangkai seri dua komponen impedansi listrik seperti Gambar II.29. Tegangan yang masuk dapat diterapkan bersebrangan impedansi seri dari Z_1 dan Z_2 dan tegangan yang keluar berasal dari Z_2. Z_1 and Z_2 dapat dirangka dengan setiap kombinasi komponen seperti resistor, induktor dan kapasitor.

Dengan menerapkan hukum Ohm, hubungan antara tegangan masukan (V_{in}) dan tegangan keluaran (V_{out}) dapat dilihat sebagai berikut :

$$V_{\text{out}} = \frac{Z_2}{Z_1 + Z_2} \cdot V_{\text{in}} \quad \text{.....(II.1)}$$

Bukti rumus :

$$V_{\text{in}} = I \cdot (Z_1 + Z_2)$$

$$V_{\text{out}} = I \cdot Z_2$$

$$I = \frac{V_{\text{in}}}{Z_1 + Z_2}$$

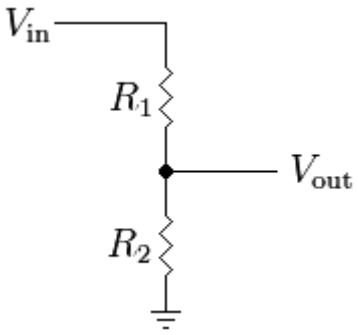
$$V_{\text{out}} = V_{\text{in}} \cdot \frac{Z_2}{Z_1 + Z_2}$$

Fungsi transfer (atau juga dikenal sebgai pembagi rasio tegangan) dari rangkaian ini dapat disederhanakan menjadi :

$$H = \frac{V_{out}}{V_{in}} = \frac{Z_2}{Z_1 + Z_2}$$ .....(II.2)

Pada umumnya fungsi transfer ini adalah komplesifitas, fungsi rasio dari frekuensi.

Resistive divider



Gambar II.31. Resistor sederhana pembagi tegangan

Pembagi resistif adalah kasus khusus dimana antara impedansi, Z_1 and Z_2, adalah murni resistor (Gambar II.30).

Dengan mengubah $Z_1 = R_1$ dan $Z_2 = R_2$ dengan rumus sebelumnya hingga menjadi :

$$V_{\text{out}} = \frac{R_2}{R_1 + R_2} \cdot V_{\text{in}}$$ .....(II.3)

Pada kasus yang lebih umum, R_1 dan R_2 mungkin bisa dari beberapa kombinasi dari resistor dengan rangkaian seri atau peralel.