

## PROGRAMMABLE LOGIC CONTROL (PLC) BERBASIS MIKROKONTROLER AT89C51

Darmono (L2F097620)  
Jurusan Teknik Elektro UNDIP

### Abstrak

Pada era sekarang ini, *Programmable Logic Control (PLC)* banyak dipakai dalam dunia industri sebagai unit kontrol utama pada mesin-mesin produksi. *PLC* merupakan sekumpulan saklar elektronik yang dapat diatur kapan harus ON dan kapan harus OFF sesuai dengan hukum-hukum kontrol logic yang telah diprogram. Salah satu keunggulan dari *PLC* adalah dapat digunakan menjadi control mesin yang berlainan hanya dengan mengubah program yang ada dalam *PLC* tanpa harus mengubah perangkat kerasnya.

Konfigurasi dari *PLC* antara lain terdiri dari *Power Supply*, *CPU* dan *Memory*. Konfigurasi ini dapat kita temukan pada mikrokontroler antara lain *AT89C51*, sehingga mikrokontroler *AT89C51* dapat kita buat atau kita program menjadi *PLC*. Untuk memasukkan program ke dalam *PLC* kita menggunakan *Programming Console*. *Programming Console* adalah suatu peralatan yang terdiri dari keypad yang digunakan untuk memilih program dan *LCD* yang digunakan untuk menampilkan program yang kita masukkan ke dalam *PLC*. Program yang dimasukkan ke dalam *PLC* berbentuk mnemonic. *Programming Console* ini akan berkomunikasi dengan *PLC* sehingga program yang kita masukkan dapat disimpan dalam memori *PLC*.

### 1. Pendahuluan

Mikrokontroler merupakan suatu sistem minimum dari mikrokomputer, yang terdiri atas sebuah mikroprosesor, memori dan unit keluaran masukan. Dengan menggunakan mikrokontroler didapat suatu keuntungan baik secara material maupun kemudahan dalam pengembangan aplikasinya. Hal ini dimungkinkan karena aplikasi dengan menggunakan mikrokontroler merupakan gabungan dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang menggerakannya.

Pada dunia industri mikrokontroler banyak diterapkan sebagai sistem kontrol industri mengingat aplikasinya yang mudah dan murah. Dengan sebuah mikrokontroler dan beberapa komponen analog kita sudah dapat membuat suatu sistem kontrol untuk dunia industri yang diterapkan pada mesin-mesin industri. Disamping mikrokontroler, *PLC* merupakan kontrol industri yang juga banyak digunakan. *PLC* digunakan karena memiliki kehandalan-kehandalan antara lain mudah diprogram dan diaplikasikan, pengawatan (*wiring*) lebih sedikit, *troubleshooting* sistem lebih sederhana, konsumsi daya relatif lebih rendah, modifikasi sistem lebih sederhana dan cepat. Suatu *PLC* dapat diprogram untuk menggantikan puluhan atau ratusan rangkaian kontrol *logic* yang saling tidak bergantung. I/O pada *PLC* memungkinkan untuk *interfacing* langsung dengan proses yang sebenarnya (*riil*).

### 2. Teori Programmable Logic Control (PLC) dan Mikrokontroler AT89C51

#### 2.1 Programmable Logic Control (PLC)

*Programmable Logic Control (PLC)* merupakan peralatan kontrol yang dirancang untuk menggantikan rangkaian kontrol *logic*, *relay* dan *timer* yang merupakan tulang punggung bagi sistem kontrol untuk proses yang kompleks.

Suatu *PLC* dapat diprogram untuk menggantikan puluhan atau ratusan rangkaian kontrol *logic* yang saling tidak bergantung. I/O pada *PLC* memungkinkan untuk *interfacing*

langsung dengan proses yang sebenarnya (*riil*). *PLC* menerima input data yang berupa sinyal dari peralatan input luar (*external input device*) dari sistem yang dikontrol. Peralatan input luar itu antara lain berupa sakelar, tombol dan sensor. Data-data yang masih berupa sinyal analog akan diubah oleh modul input *analog to digital module (A/D)* menjadi sinyal digital. Selanjutnya unit prosesor sentral atau *CPU* akan mengolah sinyal digital tersebut sesuai dengan program yang telah disimpan pada memori. Selanjutnya *CPU* mengambil keputusan dan memberikan perintah melalui modul output dalam bentuk sinyal digital. Modul *digital to analog (D/A)* akan mengubah sinyal digital ke dalam sinyal analog. Sinyal analog inilah yang mampu menggerakkan peralatan output luar (*external output device*) dari sistem yang dikontrol antara lain berupa *kontaktor*, *relay*, *solenoid valve*, *heater* dan *alarm*.

#### 2.1.1 Konfigurasi Programmable Logic Control (PLC) <sup>[1]</sup>

Konfigurasi dari *Programmable Logic Control (PLC)* terdiri atas *Power Supply Unit*, *Central Processing Unit (CPU)*, *Memory Unit*, *I/O Unit* dan *Peripheral*.

##### 2.1.1.1 Power Supply Unit

Unit ini berfungsi memberikan sumber daya pada *PLC* sehingga memungkinkan *PLC* untuk dapat bekerja.

##### 2.1.1.2 Central Processing Unit (CPU)

Unit ini merupakan otak dari *PLC*. *CPU* berfungsi untuk mengolah program sesuai dengan hukum kontrol *logic*, melakukan pengawasan atas semua operasional kerja dari *PLC*, dan transfer informasi melalui *internal bus* antara *PLC*, memori, dan unit I/O. Irama kerja dari *CPU* *PLC* dikendalikan oleh suatu *generator clock* di luar *CPU* yang bervariasi sesuai dengan tipe dari *PLC*.

##### 2.1.1.3 Memori Unit

Memori berfungsi untuk menyimpan program yang telah dibuat, menyimpan data dan status input/output (*interfacing information*), dan menyimpan data/informasi untuk fungsi-fungsi internal (*timer*, *counter*, *marker relay*, dan lain-

lain). Unit memori yang dipakai bisa berupa *Random Acces Memory* (RAM), *Erasable Programmable Read Only Memory* (EPROM) atau *Electronic Erasable Programmable Read Only Memory* (EEPROM). Untuk menjaga agar program kerja dan data-data yang penting tidak hilang jika terdapat suatu gangguan *power supply*, maka pada PLC sering disertai dengan suatu *battery backup*.

#### 2.1.1.4 Input/Output (I/O) Unit

Pada umumnya informasi data pada PLC dinyatakan dalam bentuk tegangan listrik antara 5 – 15 VDC, sedangkan system tegangan di luar bervariasi dari 24 – 240 VDC maupun AC. Unit I/O dimaksudkan untuk interfacing antara kedua besaran tersebut.

#### 2.1.1.5 Peripheral

Untuk memasukkan program ke dalam PLC diperlukan suatu unit pemrogram. Unit ini dapat berupa :

##### 1. Programming Console

*Programming Console* berguna untuk membuat, mengubah, memasukkan, memantau dan menjalankan program pada PLC. Bagian utama dari *Programming Console* adalah *LCD Display-monitor*, *mode selector switch (Program-Monitor-Run)*, *keyboard* (kunci numeric, kunci instruksi dan kunci operasi) dan *connector lead*. Program yang dimasukkan ke dalam PLC berbentuk *mnemonic*.

##### 2. Sysmate Support Software (SSS)

*Sysmate Support Software* (SSS) adalah suatu software khusus yang memungkinkan *programmer* dapat mensimulasikan program kerja dari PLC dengan menggunakan *Personal Computer* (PC)..

### Mikrokontroler AT89C51

#### 2.2.1 Arsitektur dan Organisasi AT89C51 [8]

AT89C51 terdiri dari sebuah CPU (*Central Processing Unit*), dua buah jenis memory data dan program, port *input/output*, dan *register-register*.

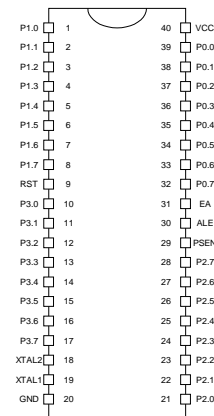
Spesifikasi Mikrokontroler AT89C51 dapat dijelaskan sebagai berikut :

- Kompatibel dengan keluarga MCS-51 dari produk Intel
- Memiliki satu prosessor 8 bit
- Kemampuan untuk mengalami memory program hingga 64 Kbyte
- Kemampuan untuk mengalami memory data hingga 64 kbyte
- Memiliki Flash PEROM internal 4096 Byte
- Memiliki RAM 128 Byte
- Memiliki struktur detak internal dan rangkaian pewaktuan
- Memiliki 4 buah terminal masukan dan keluaran masing-masing 8 jalur
- Memiliki 2 *timer/counter* 16 bit (T0 dan T1)
- Mempunyai 2 interupsi external, 2 interupsi timer, interupsi serial dan reset
- Mampu beroperasi pada pengoperasian dengan frekuensi 24 MHz
- Kemampuan melaksanakan suatu instruksi matematika seperti operasi penambahan, pengurangan, perkalian, pembagian dan operasi *Boolean*

#### 2.2.2 Pena AT89C51 [16]

Susunan pena-pena mikrokontroler AT89C51 diperlihatkan pada gambar 2.1. Penjelasan dari masing-masing pena adalah sebagai berikut :

- Pena 1 – 8 (Port 1) merupakan port parallel 8 bit dua arah (*bi-directional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*)
- Pena 9 (reset) adalah masukan reset (aktif tinggi). Pulsa transisi dari rendah ke tinggi akan mereset mikrokontroler AT89C51.



Gambar 2.1. Pena Mikrokontroler AT89C51

- Pena 10 – 17 (Port 3) adalah port parallel 8 bit dua arah yang memiliki fungsi pengganti sebagai berikut :
  - Pena 10 (RxD) : *Receive Data* (untuk port serial)
  - Pena 11 (TxD) : *Transmit data* (untuk port serial)
  - Pena 12 (INT0) : *External Interrupt 0*
  - Pena 13 (INT1) : *External Interrupt 1*
  - Pena 14 (T0) : *Timer/Counter 0*
  - Pena 15 (T1) : *Timer/Counter 1*
  - Pena 16 (WR) : *External Data Write*
  - Pena 17 (RD) : *External Data Read*
- Pena 18 (XTAL2) adalah pena keluaran ke osilator luar. Sebuah osilator luar atau kristal dapat digunakan
- Pena 19 (XTAL1) adalah pena masukan ke osilator luar.
- Pena 20 (GND) dihubungkan ke ground
- Pena 21 – 28 (Port 2) adalah port parallel 8 bit dua arah (*bi-directional*)
- Pena 29 (PSEN = *Program Store Enable*) yang merupakan sinyal pengontrol yang memperbolehkan program memory luar masuk ke dalam bus selama proses pemberian atau pengambilan instruksi (*fetching*)
- Pena 30 (ALE = *Address Lacth Enable*) yang digunakan untuk menahan alamat memory luar selama pelaksanaan instruksi
- Pena 31 (EA = *External Access*) , input dari pena ini bisa tinggi dan bisa rendah. Bila EA diberi logika tinggi, mikrokontroler akan melaksanakan instruksi dari Flash PEROM dan bila pin ini diberi logika rendah maka mikrokontroler akan melaksanakan instruksi dari memori luar

- k. Pena 32 – 39 (Port 0) merupakan port parallel 8 bit *open drain* dua arah. Bila digunakan untuk mengakses memori luar port ini akan memultipleks alamat memori dengan data
- l. Pena 40 (Vcc) dihubungkan ke Vcc (5 volt)

### 2.2.3 Register AT89C51 <sup>[8]</sup>

Menurut fungsinya, register pada mikrokontroler AT89C51 digolongkan menjadi dua yaitu Register Serbaguna (*General Purpose Register*) dan Register Fungsi Khusus (*Special Function Register/SFR*).

Register serbaguna berfungsi untuk menyimpan data sementara yang akan diolah oleh processor yang berupa operasi logika ataupun aritmatika. Register serbaguna hanya dapat berfungsi sebagai nilai operand kedua pada perhitungan aritmatika maupun logika dengan operand utama register akumulator. Mikrokontroler AT89C51 mempunyai register serbaguna delapan bit yaitu R0 – R7 yang menempati delapan bank alamat pada lokasi RAM Internal.

Register Fungsi Khusus (SFR) menempati RAM internal pada alamat 128 (80h) sampai 255 (ffh). Register-register tersebut adalah :

- a. Register A ( Akumulator )  
Merupakan register delapan bit yang digunakan untuk operasi aritmatika dan logika dan tempat untuk menyimpan operasi tersebut.
- b. Register B  
Merupakan register delapan bit yang digunakan bersama dengan register A untuk operasi perkalian ataupun pembagian.
- c. Register Penunjuk Tumpukan (SP = *Stack Pointer*)  
*Stack Pointer* adalah register delapan bit yang menunjukkan alamat data terakhir yang dimasukkan (*push*) ke dalam *stack*, juga sebagai alamat dari byte selanjutnya yang akan dikeluarkan (*pop*).
- d. Register Penampung Data Serial (SBUF = *Serial Buffer*)  
Register ini berfungsi menampung sementara data dari hasil penerimaan atau pengiriman data serial dari atau ke terminal data serial.
- e. Register Penunjuk Data (DPTR = *Data Pointer*)  
Merupakan register 16 bit yang berfungsi untuk mengakses lokasi kode alamat (*code address space*) dan lokasi alamat external. DPTR terdiri dari dua buah register yaitu DPH dan DPL, masing-masing 8 bit.
- f. Register Pencacah Program (PC = *Program Counter*)  
Merupakan register 16 bit yang berisi alamat dari instruksi berikutnya yang akan dilaksanakan oleh unit CPU
- g. Register Penyimpan Program Status (PSW = *Program Status word*)  
Merupakan register 8 bit yang berfungsi sebagai penanda status (*status flag*) dari pelaksanaan suatu instruksi.
- h. Register Pengatur Ragam Kerja Pewaktu/Pencacah (TMOD = *Timer/Counter Mode Register*)  
Merupakan register 8 bit yang berfungsi untuk memilih mode waktu/pencacah pada mikrokontroler
- i. Register Pengatur Kerja Pencacah/Pewaktu (TCON= *Timer Counter*)  
Merupakan register 8 bit yang berfungsi untuk mengatur kerja pewaktu atau pencacah

- j. Register Pengatur Serial (SCON = *Serial Port Control*)  
Merupakan register 8 bit yang berfungsi untuk mengatur ragam penerimaan atau pengiriman data serial melalui terminal serial
- k. Register Pengatur Interupsi (IE = *Interupsi Enable*)  
Merupakan register 8 bit yang berfungsi sebagai pengatur interupsi. Pena *interupsi* yang dimaksudkan pada bit IE difungsikan atau tidak
- l. Register Pengontrol Prioritas Interupsi (IP = *Interupsi Priority*)  
Merupakan register 8 bit yang berfungsi untuk menentukan prioritas interupsi sesuai dengan yang diinginkan.

### 2.2.4 Timer <sup>[4],[16]</sup>

AT89C51 dilengkapi dengan dua perangkat *Timer/Counter* 16 bit, yaitu *Timer/counter 0* dan *Timer/counter 1*. Pencacah biner untuk *Timer 0* dibentuk dengan register TL0 (*Timer 0 Low Byte*) dan register TH0 (*Timer 0 High Byte*). Pencacah biner untuk *Timer 1* dibentuk dengan register TL1 (*Timer 1 Low Byte*) dan register TH1 (*Timer 1 High Byte*).

Apabila *timer/counter* diaktifkan pada frekuensi kerja mikrokontroler 12 MHz, *timer/counter* akan melakukan hitungan waktu sekali setiap satu mikrodetik secara *independen*, tidak tergantung pada pelaksanaan suatu instruksi. Satu siklus pencacahan waktu berpadanan dengan satu siklus pelaksanaan instruksi, sedangkan satu siklus dilaksanakan dalam waktu satu mikrodetik. Bila dimisalkan suatu urutan instruksi telah selesai dilaksanakan dalam waktu 5 mikrodetik, pada saat itu pula *timer/counter* telah menunjukkan perioda waktu 5 mikrodetik.

Apabila perioda waktu tertentu telah terlampaui, *timer/counter* segera menginterupsi mikrokontroler untuk memberitahukan bahwa perhitungan perioda waktu telah selesai dilaksanakan.

### 2.2.5 Interupt <sup>[4],[16]</sup>

Pada mikrokontroler AT89C51 terdapat beberapa saluran interupsi, interupsi tersebut dibedakan menjadi dua yaitu :

- a. Interupsi yang tidak dapat dihalangi oleh perangkat lunak (*non maskable interrupt*), misalnya reset
- b. Interupsi yang dapat dihalangi oleh perangkat lunak (*maskable interrupt*), yaitu INT0 dan INT1 yang merupakan interupsi *external* serta *Timer/Counter 0*, *Timer/Counter1*, dan interupsi port serial yang merupakan interupsi *internal*

Tabel 2.1. Lokasi Interupsi

Nama	Lokasi	Alat Interupsi
Reset	00 H	<i>Power on reset</i>
Int 0	03 H	<i>INT 0</i>
Timer 0	0B H	<i>Timer 0</i>
Int 1	13 H	<i>INT 1</i>
Timer 1	1B H	<i>Timer 1</i>
Sint	23 H	<i>Port I/O Serial</i>

### 2.2.6 Serial <sup>[14]</sup>

Port seri pada AT89C51 dapat bekerja pada 4 mode, satu mode bekerja secara *sinkron* dan tiga mode lainnya bekerja secara *asinkron*. Keempat mode tersebut adalah :

a. Mode 0

Mode 0 bekerja secara *sinkron*, *clock* dikirim bersama dengan data seri. Data seri dikirim dan diterima melalui kaki P3.0 (RxD), dan kaki P3.1 (TxD) dipakai untuk menyalurkan clock pendorong data seri yang dibangkitkan AT89C51. Data dikirim/diterima 8 bit sekaligus, dimulai dari bit yang bobotnya paling kecil (bit 0) dan diakhiri dengan bit yang bobotnya paling besar (bit 7).

b. Mode 1

Mode ini dan mode-mode berikutnya bekerja secara *asinkron*, data dikirim melalui kaki P3.1 (TxD) dan diterima melalui kaki P3.0 (RxD). Pada Mode 1 data dikirim/diterima 10 bit sekaligus, diawali dengan 1 bit *start*, disusul dengan 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), diakhiri dengan 1 bit *stop*. Pada AT89C51 yang berfungsi sebagai penerima bit *stop* ditampung pada RB8 dalam register SCON.

c. Mode 2

Data dikirim/diterima 11 bit sekaligus, diawali dengan 1 bit *start*, disusul 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), kemudian bit ke 9 yang bisa diatur lebih lanjut, diakhiri dengan 1 bit *stop*. Pada MCS51 yang berfungsi sebagai pengirim, bit 9 tersebut berasal dari bit TB8 dalam register SCON. Pada AT89C51 yang berfungsi sebagai penerima, bit 9 ditampung pada bit RB8 dalam register SCON, sedangkan bit stop diabaikan tidak ditampung.

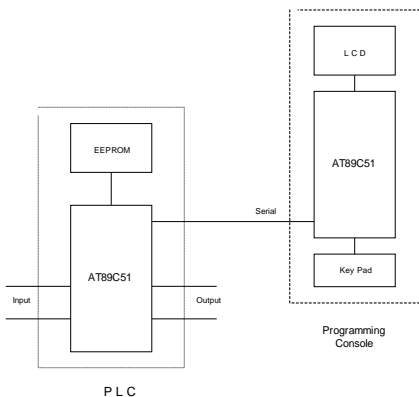
d. Mode 3

Mode ini sama dengan Mode 2, hanya saja kecepatan pengiriman data (*baud rate*) bisa diatur sesuai dengan keperluan, seperti halnya Mode 1.

### 3. Perancangan Sistem

#### 3.1 Blok Diagram Sistem

Sistem PLC berbasis mikrokontroler AT89C51 yang dibuat terdiri dari 2 bagian utama yaitu PLC dan *Programming console*. PLC dilengkapi dengan EEPROM *External* untuk menyimpan program, 6 jalur masukan dan 4 jalur keluaran. *Programming Console* dilengkapi dengan *keypad* yang berfungsi untuk memasukkan program dan LCD yang digunakan untuk menampilkan program yang diisikan.



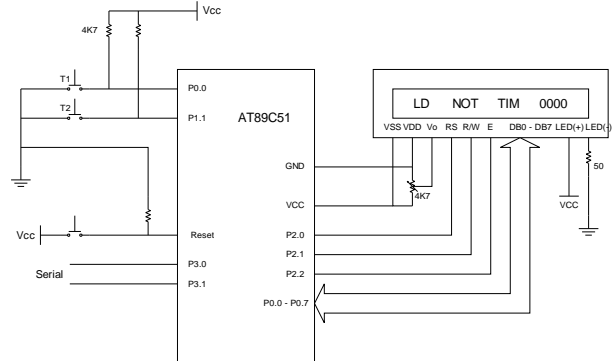
Gambar 3.1 Blok Diagram Sistem

#### 3.2 Perancangan Perangkat Keras

Pembuatan sistem perangkat keras terdiri dari 2 blok yaitu blok PLC dan blok *Programming console*.

##### 3.2.1 Perangkat Keras Pada *Programming Console*

Blok diagramnya adalah sebagai berikut :

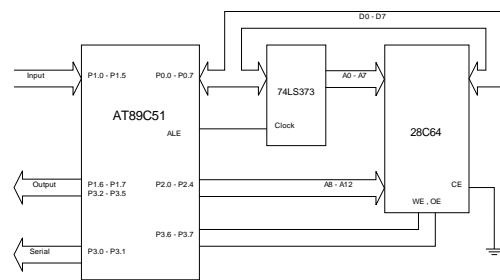


Gambar 3.2 Rangkaian *Programming console*

Dari blok diagram di atas dapat kita lihat bagian-bagian dari *Programming console* yaitu AT89C51, LCD dan Keypad. AT89C51 merupakan otak dari *Programming console* yang akan mengirimkan data ke dalam PLC melalui kaki serial (port 3.0 dan port 3.1), *keypad* berfungsi memberikan kode pada AT89C51 tentang program yang harus dikirimkan ke dalam PLC dan LCD menampilkan program yang dikirim ke dalam PLC. Tombol yang digunakan pada *keypad* berjumlah 2 buah yang diset aktif rendah. Satu tombol digunakan untuk memilih program yang dimasukkan dan satu tombol enter. LCD dihubungkan dengan AT89C51 pada Port 0 dan Port 2.0 sampai Port 2.2.

AT89C51 akan diprogram sebagai *Programming console* dengan memperhatikan masukan *keypad* dan menampilkan program pada LCD. Program ini akan dikirim ke dalam PLC melalui kaki serial pada port 3.1 dan port 3.2. Data yang dikirim ke PLC berupa data-data kode dari perintah-perintah PLC yang akan disimpan dalam EEPROM external dari PLC (EEPROM 28C64).

##### 3.2.2 Perangkat Keras Pada PLC



Gambar 3.3 Perangkat keras pada PLC

Perangkat keras pada PLC terdiri atas mikrokontroler AT89C51, EEPROM 28C64, 74LS373 dan 4N25. EEPROM 28C64 berfungsi untuk menyimpan program PLC yang dimasukkan dengan *Programming console*. 74LS373 merupakan IC *Latch* yang menahan byte bawah alamat dari program yang akan disimpan dalam EEPROM. Program yang disimpan dalam 28C64 berupa kode-kode dari perintah yang

telah dimasukkan melalui *Programming console*. Saat menjalankan program (*running*) AT89C51 akan mengambil program/kode dari 28C64 yang kemudian diterjemahkan ke dalam bahasa AT89C51.

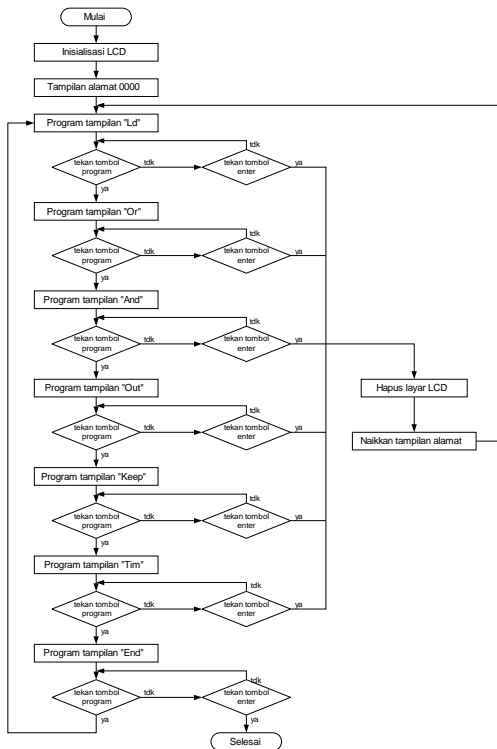
### 3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada PLC berbasis mikrokontroler ini juga dibagi menjadi dua yaitu perancangan perangkat lunak pada *Programming Console* dan perancangan perangkat lunak pada PLC

#### 3.3.1 Perangkat Lunak Pada *Programming console*

##### 3.4.1.1 Program keypad dan LCD

*Flowchart* untuk keypad dan LCD :



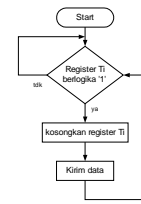
Gambar 3.4 *Flowchart* keypad dan tampilan LCD

Prinsip kerja dari LCD dan *Keypad* adalah sebagai berikut:

Saat tombol power ditekan AT89C51 akan menginisialisasi LCD dan selanjutnya LCD akan menampilkan alamat "0000" pada baris pertama dan pada baris kedua akan berisi program tampilan "Ld". Selanjutnya mikrokontroler akan menunggu tombol yang ditekan, apakah tombol program atau tombol enter. Jika tombol program yang ditekan maka akan masuk pada program tampilan "Or", dan jika tombol enter yang ditekan maka program pada LCD akan dihapus dan pada baris pertama akan berisi tampilan "0001", dan *routine* akan kembali ke awal. Jika tombol program yang terus ditekan maka akan masuk pada program tampilan "And", program tampilan "Out", program tampilan "Keep", program tampilan "Tim" dan program tampilan "End". Dan jika pada *routine* program tampilan tertentu tombol enter ditekan maka tampilan pada layar LCD akan terhapus dan tampilan bilangan alamat pada baris pertama akan naik satu bilangan dan *routine* akan kembali ke awal.

#### 3.4.1.2 Pengiriman Data Serial ke PLC

*Flowchart* untuk pengiriman data secara serial :



Gambar 3.5 *Flowchart* pengiriman data secara serial

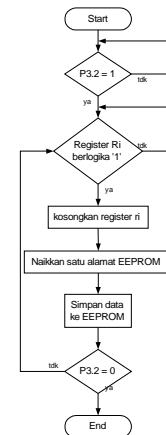
Data serial yang akan dikirimkan berupa kode-kode dari perintah PLC. Data yang akan dikirim dimasukkan ke register accumulator dan AT89C51 menunggu sampai TI berlogika '1'. Setelah TI berlogika '1' maka isi dari TI akan diclear menjadi '0' dan kemudian data dikirim ke PLC.

#### 3.3.2 Perangkat Lunak Pada PLC

Perancangan perangkat lunak pada PLC dibagi menjadi dua bagian yaitu perancangan perangkat lunak untuk penerimaan data serial dan penyimpanan data di EEPROM serta perancangan perangkat lunak untuk *running* PLC.

##### 3.3.2.1 Penerimaan Data Serial dan Penyimpanan Data di EEPROM

*Flowchart* untuk penerimaan data secara serial dan penyimpanan data di EEPROM adalah sebagai berikut :

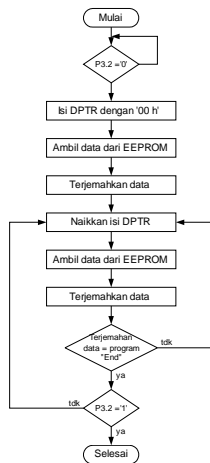


Gambar 3.6 *Flowchart* penerimaan data serial dan penyimpanan data di EEPROM

P3.2 merupakan pena yang mengindikasikan pengisian program ke dalam PLC dan *running* PLC. Jika p3.2 berlogika tinggi maka *Programming Console* siap mentransmisikan data ke dalam PLC. PLC siap menerima data dengan ditandai logika tinggi pada register "ri". "Ri" akan diclear dan alamat pada EEPROM dinaikkan satu dan kemudian data akan disimpan di EEPROM. Jika p3.2 masih berlogika tinggi maka PLC siap menerima data dari programming console dan jika p3.2 telah berlogika rendah maka PLC akan menjalankan rutin "*running* PLC". Saat proses *running* PLC, AT89C51 akan mengambil data yang telah disimpan pada EEPROM dan kemudian diterjemahkan menjadi perintah-perintah PLC.

#### 3.4.2.2 *Running* PLC

*Flowchart* untuk pengambilan data dari EEPROM dan *running* PLC :



Gambar 3.7 Flowchart pengambilan data dari EEPROM dan running PLC

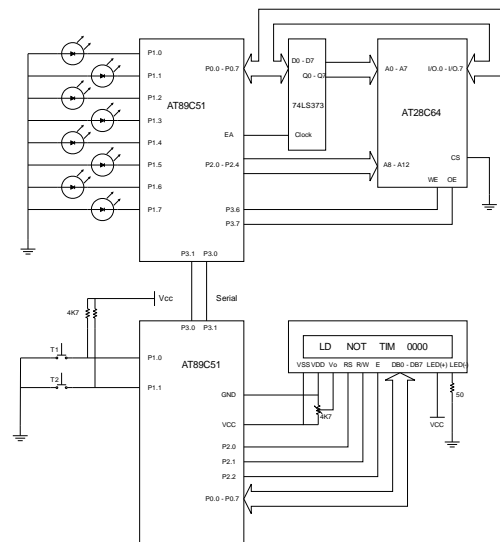
Untuk mengawali proses *Running* PLC, AT89C51 akan menunggu samapai p3.2 berlogika rendah. Jika p3.2 sudah berlogika rendah maka AT89C51 akan memberi perintah untuk mengisi register DPTR dengan alamat 0000 h dan kemudian mengambil data dari EEPROM. Data kode yang diambil dari EEPROM akan diterjemahkan oleh AT89C51 dan kemudian program akan dijalankan. Mikrokontroler memerintahkan untuk menaikkan alamat DPTR dan kemudian mengambil data dan menerjemahkannya. Jika hasil terjemahan data sama dengan program “End” maka mikrokontroler akan memeriksa keadaan logika pada p3.2. Jika p3.2 masih berlogika rendah maka AT89C51 akan mengambil data di EEPROM pada alamat 0000 h dan kemudian diterjemahkan lagi. Jika terjemahan data bukan program “End” maka mikrokontroler akan menaikkan alamat DPTR dan mengambil data serta menerjemahkannya. Hasil terjemahan data sama dengan program “End” dan p3.2 berlogika tinggi maka proses *running* PLC telah selesai dan mikrokontroler siap menerima data serial. Data yang berada pada alamat “0000” dan alamat “genap” akan diterjemahkan menjadi 3 perintah program yaitu program pada kolom1, kolom2 dan kolom3 sedangkan data pada alamat “ganjil” akan diterjemahkan menjadi perintah yang ada pada kolom4.

#### 4. Pengujian Alat

##### 4.1 Pengujian Pengisian Program ke dalam PLC

Pada pengujian ini *Programming Console* akan mengirimkan kode-kode dari perintah-perintah PLC seperti perintah “Ld”, “Or”, “And” dan sebagainya. Tombol *keypad* akan menginstruksikan pemilihan program dan LCD akan menampilkan program yang telah dipilih. *Programming Console* akan mengirimkan kode perintah yang telah dipilih ke PLC melalui komunikasi serial. PLC akan menerima kode perintah tersebut dan akan menyimpannya pada EEPROM 28C64.

Rangkaian ujinya adalah sebagai berikut :



Gambar 4.1 Rangkaian uji pengiriman data serial dan penyimpanan data di EEPROM

Pada rangkaian di atas LED yang dihubungkan dengan port 1 pada AT89C51 yang berfungsi sebagai PLC digunakan untuk menampilkan data yang diambil dari EEPROM sehingga dapat dideteksi kesalahan yang terjadi pada pengiriman data serial dan penyimpanan data di EEPROM.

Program yang digunakan adalah sebagai berikut:

```
<<<<< Program Penerimaan Data Serial, Simpan di EEPROM, baca EEPROM dan tampilkan pada port 1 >>>>>
start: mov scon,#52h
      mov tmod,#20h
      mov th1,#0e6h
      setb tr1
      mov dptr,#00h
      jnb ri,$
      clr ri
      mov a,sbuf
      movx @dptr,a
      acall delay
      movx a,@dptr
      mov p1,a
      acall delay
terima: jnb ri,$
      clr ri
      inc dptr
      mov a,sbuf
      movx @dptr,a
      acall delay
      movx a,@dptr
      mov p1,a
      acall delay
      ajmp terima
delay: mov r0,#0
delay1: mov r5,#1
      djnz r5,$
      djnz r0,delay1
      ret
```

Pada rutin program ini data pertama yang diterima akan disimpan pada EEPROM dengan alamat 0000h dan akan dibaca serta ditampilkan pada port 1. Untuk data berikutnya akan disimpan pada alamat 0001h dan juga akan dibaca dan ditampilkan pada LED di port 1.

Kode dari satu baris program akan disimpan ke dalam dua alamat dari EEPROM. Kode dari perintah pada kolom1

sampai kolom3 pada LCD akan dijumlahkan dan selanjutnya akan dikirim ke PLC dan kemudian disimpan pada satu alamat EEPROM. Kode dari kolom4 pada LCD akan disimpan pada alamat berikutnya.

Program yang akan dikirim ke PLC melalui komunikasi serial adalah sebagai berikut :

Tabel 4.1 Kode program yang dimasukkan ke EEPROM

Program	Kode			
Kolom1 – Kolom4	Kolom1	Kolom2	Kolom3	Kolom4
Ld 0000	20 h	02 h	00 h	0b0 h
Ld Not Tim 000	20h	04 h	01 h	0a0 h
Or Tim 000	40 h	02h	01 h	0a0 h
And Not 0001	60h	04h	00 h	0b1 h
Out 1000	80h	02 h	00 h	0c0 h
Out Not 1001	80h	04 h	00 h	0c1 h
Keep 1410	0a0 h	02 h	00 h	0d0 h
Tim 000	0.1 s	0c0 h	08 h	0e0 h
Or Not 1411	0e0 h	04h	00 h	0d1 h
End				

Dari pengujian didapat hasil sebagai berikut :

Tabel 4.2 Hasil pengujian pengiriman data melalui programming console dan disimpan di EEPROM

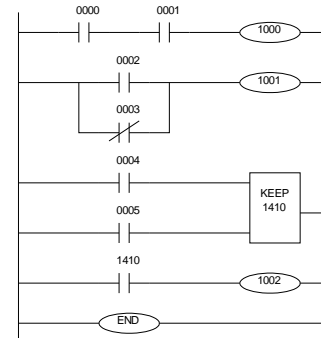
Program	Tegangan ( V )	P	P	P	P	P	P	P	P
Kol1 - Kol3	Kol4	1.	1.	1.	1.	1.	1.	1.	1.
		7	6	5	4	3	2	1	0
Ld		0	0	1	0	0	0	1	0
	0000	1	0	1	1	0	0	0	0
Ld Not Tim		0	0	1	0	0	1	0	1
	000	1	0	1	0	0	0	0	0
Or Tim		0	1	0	0	0	0	0	1
	000	1	0	1	0	0	0	0	0
And Not		0	1	1	0	0	1	0	0
	0001	1	0	1	1	0	0	0	1
Out		1	0	0	0	0	0	1	0
	1000	1	1	0	0	0	0	0	0
Out Not		1	0	0	0	0	1	0	0
	1001	1	1	0	0	0	0	0	1
Keep		1	0	1	0	0	0	1	0
	1410	1	1	0	1	0	0	0	0
Tim 000		1	1	0	0	1	0	0	0
	0.1 s	1	1	1	0	0	0	0	0
Or Not		0	1	0	0	0	1	0	0
	1411	1	1	0	1	0	0	0	1
End		1	1	1	0	0	0	0	0

Pada pengujian didapatkan hasil yang telah sesuai dengan kode yang telah dikirim oleh *Programming Console*. Perintah “Ld Not Tim” diindikasikan dengan menyala LED pada p1.6, p1.5, p1.2 dan p1.0 yang menandakan bahwa kode yang telah di baca pada EEPROM adalah “25h”, sedangkan perintah “000” diindikasikan dengan menyala LED pada p1.7 dan p1.5 yang menandakan bahwa kode yang telah dibaca adalah “ a0 h “. Pada tabel kode program perintah “Ld Not Tim 000” dikodekan dengan bilangan “20h”, “04h”, “01h” dan “a0h”. Jika kode perintah “Ld Not Tim” dijumlahkan maka didapatkan hasil “25h”. Hasil penjumlahan ini sesuai dengan hasil yang telah diperoleh dari pengujian.

## 4.2 Running PLC

### 4.2.1 Uji Running PLC1

Program berikut ini adalah program uji yang dimasukkan ke dalam PLC melalui *Programming Console*.



Gambar 4.2 Ladder uji Running PLC 1

Simbol mnemonicnya adalah sebagai berikut:

Ld		0000
And		0001
Out		1000
Ld		0002
Or	Not	0003
Out		1001
Ld		0004
Ld		0005
Keep		1410
Ld		1410
Out		1002
End		

Dari hasil pengujian didapat hasil sebagai berikut :

Tabel 4.4 Hasil uji PLC1

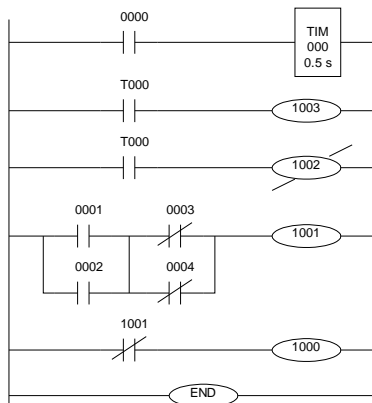
Input						Output		
0000	0001	0002	0003	0004	0005	1000	1001	1002
0	0	-	-	-	-	0	-	-
1	0	-	-	-	-	0	-	-
0	1	-	-	-	-	0	-	-
1	1	-	-	-	-	1	-	-
-	-	0	0	-	-	-	1	-
-	-	1	0	-	-	-	1	-
-	-	0	1	-	-	-	0	-
-	-	1	1	-	-	-	1	-
-	-	-	-	1	0	-	-	1
-	-	-	-	0	0	-	-	1
-	-	-	-	0	1	-	-	0
-	-	-	-	0	0	-	-	0
-	-	-	-	1	0	-	-	1
-	-	-	-	0	0	-	-	0

Dari *Ladder diagram* dapat kita lihat bahwa output “1000” hanya akan menyala bila kedua kondisi input yaitu input “0000” dan “0001” berlogika tinggi, dan jika salah satu input tersebut berlogika rendah maka output “1000” akan berlogika rendah. Hal ini juga dapat kita lihat pada hasil pengujian, output “1000” akan berlogika rendah jika salah satu input “0000” atau “0001” berlogika rendah dan output “1000” akan berlogika tinggi bila kedua input “0000” dan “0001” berlogika tinggi. Kondisi kedua input yaitu “0000” dan “0001” tidak akan mempengaruhi kondisi output “1001” dan “1002”. Kondisi output “1001” dipengaruhi oleh input “1002” dan “1003”, pada keadaan awal yaitu kedua input berlogika rendah maka output “1001” akan berlogika tinggi. Hal ini disebabkan karena adanya perintah not pada input “0003”. Output “1001” akan berlogika rendah jika kondisi input “0002” berlogika rendah dan input “0003” berlogika tinggi, selain kondisi tersebut output “1001” akan selalu berlogika tinggi. Output

“1002” dipengaruhi oleh input “0004” dan “0005”. Output “1002” akan berlogika tinggi bila input “0004” berlogika tinggi atau terjadi transisi dari “rendah-tinggi-rendah” dan input “0005” dalam keadaan berlogika rendah. Output “1002” ini akan tetap berlogika tinggi walaupun input “0004” sudah berlogika rendah selama input “0005” berlogika rendah. Output “1002” akan selalu berlogika rendah jika input “0005” berlogika tinggi. Hal ini disebabkan karena input “0005” dihubungkan dengan pin “reset” pada perintah “Keep”, jika “reset” berlogika tinggi maka “Keep” akan selalu berlogika rendah walaupun pin “set” berlogika tinggi.

#### 4.5.2.2 Uji Running PLC 2

Ladder diagram uji running PLC 2 :



Gambar 4.3 Ladder Uji PLC2

Mnemonicnya adalah sebagai berikut :

```

Ld          0000
Tim        000      0.5s
Ld          000
Out        Tim      1003
Ld          Tim      000
Out        Not      1002
Ld          0001
Or          0002
Ld        Not      0003
Or        Not      0004
And        Ld
Out        1001
Ld        Not      1001
Out        1000
End
  
```

Dari ladder diagram di atas dapat dilihat bahwa input “0000” akan menghidupkan “timer 000”. “Timer 000” akan menghitung selama “0,5 detik” dan akan menghidupkan output “1003” dan mematikan output “1002”. Output “1001” akan hidup bila salah satu input “0001” atau “0002” berlogika tinggi dan akan mati jika kedua input “0003” dan “0004” berlogika tinggi. Output “1000” akan mati jika output “1001” hidup dan juga sebaliknya output “1000” akan hidup jika output “1001” mati.

Tabel 4.5 Hasil Uji PLC2

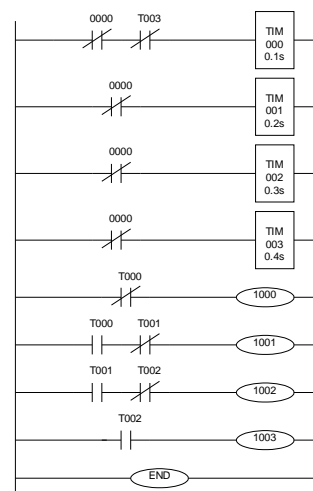
Time	0000	0001	0002	0003	0004	1003	1002	1001	1001
T0	1	-	-	-	-	0	1	-	-
T1	1	-	-	-	-	1	0	-	-
	-	0	0	0	0	-	-	0	1
	-	0	0	0	1	-	-	0	1
	-	0	0	1	0	-	-	0	1
	-	0	0	1	1	-	-	0	1
	-	0	1	0	0	-	-	1	0
	-	0	1	0	1	-	-	1	0
	-	0	1	1	0	-	-	1	0
	-	0	1	1	1	-	-	0	1
	-	1	0	0	0	-	-	1	0
	-	1	0	0	1	-	-	1	0
	-	1	0	1	0	-	-	1	0
	-	1	0	1	1	-	-	0	1
	-	1	1	0	0	-	-	1	0
	-	1	1	0	1	-	-	1	0
	-	1	1	1	0	-	-	1	0
	-	1	1	1	1	-	-	0	1

Dari tabel di atas dapat kita lihat bahwa output “1003” akan menyala setelah “0,5 detik” setelah input “0000” berlogika tinggi dan output “1002” akan mati seiring dengan hidupnya output “1003”. Output “1003” akan tetap menyala sampai tombol reset pada PLC diaktifkan. Output “1001” akan menyala jika salah satu input “0001” atau “0002” berlogika tinggi dan salah satu input “0003” atau “0004” berlogika rendah. Output “1001” ini akan tetap mati jika kedua input “1003” dan “1004” berlogika tinggi. Output “1000” akan menyala jika output “1001” mati dan akan mati jika output “1001” menyala.

#### 4.5.2.3 Uji Running PLC 3 (Motor Stepper)

Pada pengujian ini output dari PLC yang berjumlah empat buah dihubungkan dengan empat buah input dari motor stepper.

Program ujinya adalah sebagai berikut :



Gambar 4.4 Ladder diagram uji PLC3

Mnemonic :

```

Ld          Not      0000
  
```

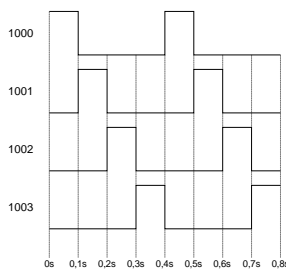


```

And    Not    Tim    003
Tim    000    0.1s
Ld     Not    0000
Tim    001    0.2s
Ld     Not    0000
Tim    002    0.3s
Ld     Not    0000
Tim    003    0.4s
Ld     Not    Tim    000
Out    1000
Ld     Tim    000
And    Not    Tim    001
Out    1001
Ld     Tim    001
And    Not    Tim    002
Out    1002
Ld     Tim    002
Out    1003
End

```

Dari *ladder diagram* di atas dapat kita lihat bahwa output “1000” akan menyala waktu PLC dihidupkan. *Timer* 000 menghitung sampai 0,1 detik dan setelah waktu tersebut tercapai maka output “1000” akan mati dan output “1001” akan menyala. *Timer* 001 akan menghitung sampai 0,2 detik dan selanjutnya akan mematikan output “1001” dan menghidupkan output “1002”. Setelah *timer* 002 mencapai harga 0,3 detik maka output “1002” akan mati dan output “1003” akan menyala. Jika *timer* 003 mencapai harga 0,4 detik maka proses akan dimulai dari awal lagi.



Gambar 4.5 *Timing diagram* uji PLC3

Setelah dilakukan pengujian didapatkan hasil bahwa motor *stepper* akan berputar searah dengan jarum jam. Tabel di bawah ini adalah tabel output dari PLC yang dapat kita lihat pada indicator Led yang ada di PLC.

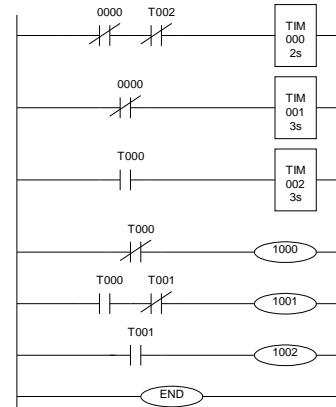
Tabel 4.6 Hasil pengujian *running* PLC3

Waktu	Output			
	1000	1001	1002	1003
0 s	1	0	0	0
0.1 s	0	1	0	0
0.2 s	0	0	1	0
0.3 s	0	0	0	1

Dari hasil pengujian didapatkan hasil yang sama dengan analisa program yang telah dimasukkan ke dalam PLC. Pada waktu PLC hidup output “1000” akan menyala dan yang lainnya mati, dan setelah waktu mencapai 0,1 detik maka output “1001” akan menyala dan yang lainnya mati. Setelah waktu mencapai 0,2 detik maka output “1002” yang menyala dan setelah waktu mencapai 0,3 detik maka output “1003” yang menyala. Proses akan kembali ke keadaan awal setiap perioda 0,4 detik. Output dari PLC ini membangkitkan pulsa yang berurutan pada tiap outputnya dengan selang waktu 0,1 detik. Pulsa inilah yang menggerakkan motor *stepper*.

#### 4.5.2.4 Pengujian *Running* PLC4

Program yang dimasukkan ke dalam PLC adalah sebagai berikut :



Gambar 4.6 *Ladder diagram* uji PLC4

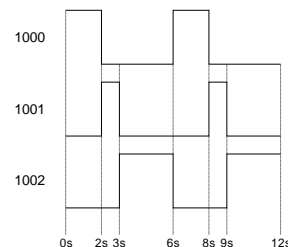
*Mnemonic* :

```

Ld     Not    0000
And    Not    Tim    002
Tim    000    2s
Ld     Not    0000
Tim    001    3s
Ld     Tim    000
Tim    002    3s
Ld     Not    Tim    000
Out    1000
Ld     Tim    000
And    Not    Tim    001
Out    1001
Ld     Tim    001
Out    1002
End

```

Dari *ladder diagram* di atas output “1000” dihubungkan dengan Led warna “Hijau”, output “1001” dihubungkan dengan Led warna “Kuning” dan output “1002” dihubungkan dengan Led warna “Merah”. Output “1000 (hijau)” akan menyala waktu PLC dihidupkan. *Timer* 000 akan mencacah sampai 2 detik dan kemudian output “1000 (hijau)” akan mati dan output “1001 (kuning)” akan menyala. Satu detik kemudian output “1001 (kuning)” akan mati dan output “1002 (merah)” akan menyala. Output “1002 (merah)” ini akan menyala selama 3 detik dan kemudian setelah waktu tersebut proses akan dimulai dari awal lagi.



Gambar 4.7 *Timing diagram* uji PLC4

Dari hasil pengujian didapatkan hasil sebagai berikut :

Tabel 4.7 Hasil uji PLC4

Waktu	Output		
	1000 (Hijau)	1001 (Kuning)	1002 (Merah)
0 detik	1	0	0
T0 (2 detik )	0	1	0
T1 (3 detik )	0	0	1
T2 (6 detik)	1	0	0

Dari hasil pengujian didapatkan hasil yang telah sesuai dengan program yang telah dimasukkan ke dalam PLC. Output “1000 (Hijau)” akan menyala selama 2 detik, selanjutnya output “1001 (Kuning)” akan menyala selama 1 detik dan kemudian output “1002 (Merah)” akan menyala selama 3 detik. Proses akan selalu berulang untuk perioda 6 detik.

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil perancangan dan pembuatan *Programmable Logic Control* (PLC) berbasis mikrokontroler AT89C51 serta pengujian yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Mikrokontroler AT89C51 dapat digunakan untuk membuat rangkaian *Programmable Logic Control* (PLC) sederhana.
2. Indikator LED pada input dan output *Programmable Logic Control* (PLC) berbasis mikrokontroler AT89C51 dapat digunakan untuk mendeteksi kesalahan program pada saat dimasukkan ke dalam PLC melalui *Programming Console*
3. Jika terjadi kesalahan dalam proses memasukkan program ke *Programmable Logic Control* (PLC) berbasis mikrokontroler AT89C51 maka proses pemasukan program harus diulang dari awal karena tidak adanya fasilitas hapus pada *Programming Console*
4. *Programmable Logic Control* (PLC) berbasis mikrokontroler AT89C51 dapat digunakan sebagai pembangkit pulsa dengan lebar pulsa minimal 0,1 detik sehingga pada pengujian menggunakan motor *stepper*, kecepatan yang dihasilkan maksimal 0,05 putaran tiap detik.

### 5.2 Saran

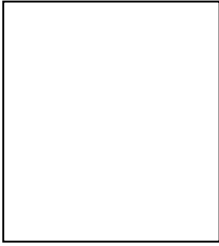
Saran- saran yang dapat diberikan untuk penggunaan dan pengembangan peralatan *Programmable Logic Control* (PLC) berbasis mikrokontroler AT89C51 adalah sebagai berikut :

1. Pembuatan *Programming Console* dengan fasilitas pemrograman yang lebih lengkap dapat dilakukan dengan cara penambahan *keypad*, penggunaan mikrokontroler dengan kapasitas PEROM yang lebih besar dan penggunaan LCD dengan jumlah karakter yang lebih banyak
2. Agar PLC dapat digunakan untuk aplikasi kontrol yang lebih besar diperlukan penambahan input/output, penggunaan mikrokontroler dengan kapasitas PEROM yang lebih besar dan penambahan kapasitas EEPROM

3. PLC harus dilengkapi dengan fasilitas untuk berinteraksi dengan komputer agar proses memasukkan program ke dalam PLC dan proses pengawasannya menjadi lebih mudah .

## DAFTAR PUSTAKA

1. ...., *C200HS Programmable Controllers Operation Manual* , 1994
2. ...., [http://www. Atmel. Com](http://www.Atmel.Com), *Configure Logic Microcontroller Non Volatile Memory*, Atmel Productcs, 1997
3. ...., [http://www. Stts. Edu](http://www.Stts.Edu) , *Instruksi MCS51 Bagian I, II dan III*, Sekolah Tinggi Teknik Surabaya, Surabaya, 2001
4. ...., [http://www. Stts. Edu](http://www.Stts.Edu) , *Interrupt & Timer*, Sekolah Tinggi Teknik Surabaya, Surabaya, 2001
5. ...., [http://www. Stts. Edu](http://www.Stts.Edu) , *Seiko Instrument M1632 LCD Module*, Sekolah Tinggi Teknik Surabaya, Surabaya, 2001
6. ...., *Sysmac C 200 HS Programmable Controller Installation guide*, OMRON , April 1997
7. ...., *Sysmac C-Series and CVMI PCs Sysmate Support Software Operation manual Basic*, OMRON, Mei 1996
8. Ahmad Subaidi, *Aplikasi Mikrokontroler AT89C51 Sebagai Pengatur Waktu Penyalaan Lampu Rumah Yang Dapat Diprogram*, Teknik Elektro UNDIP, Semarang 2001
9. Albert Paul Malvino & Hanafi Gunawan, *Prinsip-Prinsip Elektronik*, Erlangga, Jakarta, 1985
10. Charles A. Schuler & William L. McNamee, *Industrial Electronics and Robotics*, Mc Graw-Hill, Singapore, 1986
11. David A. Hodges – Horace G. Jackson, *Analisis dan Desain Rangkaian Terpadu Digital*, Erlangga, Jakarta, 1987
12. ECG Semiconductor, *Master Replacement Guide*, ECG Semiconductor, 1996
13. El-Tech, *LCD Module User Manual*, El-Tech Electronics, Surabaya,
14. I Scott MacKenzie, *The 8051 Microcontroller*, Prentice Hall, New Jersey, 1995
15. LCD & LCM, Wintek Corporation
16. Moh. Ibnu Malik & Anistardi, *Bereksperimen dengan Mikrokontroler 8031*, Elex Media Komputindo, Jakarta, 1997
17. Rodney Zaks & S. H. Nasution, *Dari Chip ke Sistem*, Erlangga, Jakarta, 1991
18. Ronald J Tocci, *Digital system*, Mc-Graw Hill Co, Singapore, 1998



Darmono (L2F 097 620), lahir di Boyolali 13 Oktober 1979. Saat ini masih berstatus sebagai Mahasiswa Angkatan '97 Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang dengan konsentrasi Kontrol.

Disetujui oleh,

Pembimbing I

Pembimbing II

Ir. Agung Warsito, DHET

NIP : 131 668 485

Darjat, ST

NIP : 132 231 135