

**APLIKASI DASAR MIKROKONTROLER
ATMEGA8535 DENGAN BAHASA BASIC
MENGUNAKAN *DIGITAL OUT (LED, 7 SEGMENT,
DOT Matrik, LCD), DAN DIGITAL IN (SWITCH,
KEYPAD, KEYBOARD)***

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika



Oleh :

Nama : Muhammad Khaldun Rasyidi

No. Mahasiswa : 05.523.035

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA 2011**

LEMBAR PENGESAHAN PEMBIMBING

**APLIKASI DASAR MIKROKONTROLER ATMEGA8535
DENGAN BAHASA BASIC MENGGUNAKAN *DIGITAL OUT*
(*LED, 7 SEGMENT, DOT Matrik, LCD*), DAN *DIGITAL IN*
(*SWITCH, KEYPAD, KEYBOARD*)**

TUGAS AKHIR



Yogyakarta, 27 Oktober 2011

Pembimbing

Zainudin Zukhri,ST., MIT

**LEMBAR PENGESAHAN PENGUJI
APLIKASI DASAR MIKROKONTROLER ATMEGA8535
DENGAN BAHASA BASIC MENGGUNAKAN *DIGITAL OUT*
(*LED, 7 SEGMENT, DOT Matrik, LCD*), DAN *DIGITAL IN*
(*SWITCH, KEYPAD, KEYBOARD*)
TUGAS AKHIR**

Oleh :

Nama : Muhammad Khaldun Rasyidi

No. Mahasiswa : 05.523.035

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika Fakultas
Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 27 Oktober 2010

Tim Penguji

Zainudin Zuhri S.T.,M.I.T

Ketua

Syarif Hidayat, S.Kom.,MIT

Anggota I

Izzati Muhimmah, ST., M.Sc., Ph.D

Anggota II

**Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia**

Yudi Prayudi, S.Si., M.Kom.

PERSEMBAHAN

Puji syukur kehadirat Allah SWT yang telah memberiku begitu banyak waktu untuk dapat menikmati hidup yang pahit tetapi begitu indah.

Tugas akhir ini kupersembahkan dengan penuh rasa sayang kepada :

AYAH dan alm. IBU tercinta

Terima kasih atas kasih sayang, pengorbanan, kerja keras, doa dan semua hal yang kalian lakukan untuk membahagiakan keluarga.

Juga pelajaran yang sangat berharga yang kalian tinggalkan kepadaku.

Khususnya buat ibu tercinta, I miss u mom.

Keluarga besar saya yang ada di tegal, Sulawesi dan Maluku tenggara. Terima kasih atas segala pengorbanan, perhatian, kasih sayang, dan pelajaran hidup yang kalian berikan kepadaku.

Semoga ALLAH selalu meridhoi kalian.

Mba ufi dan kakak-kakakku sekalian yang udah memberiku kebebasan dan kepercayaan kepadaku untuk melakukan apa yang ku mau. Juga support materil dan spirituil kalianlah yang bisa membuatku sampai di jenjang pendidikan ini.

Semua teman-teman terbaik yang pernah aku dapatkan...

MOTTO

“... Jadikanlah sabar dan shalat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar”.

(Q.S. Al-Baqarah ayat 153)

“Sesungguhnya sesudah kesulitan itu ada kemudahan; Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain”.

(Q.S. Alam Nasyrat ayat 6 & 7)

Hidup akan lebih bahagia kalau kita dapat menikmati apa yang kita miliki. Karena itu bersyukur merupakan kualitas hati yang tertinggi. Kalau kamu belum bisa memiliki apa yang kamu cintai,

Cintailah apa yang telah kamu miliki...

(Little-Fairy)

Sempurnakan upayamu, maka ia akan membatalkan ketidakmungkinan. Kamu pasti bisa!

(Mario teguh)

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya sehingga penyusun dapat menyelesaikan tugas akhir dengan judul “Aplikasi dasar mikrokontroler ATMEGA8535 dengan bahasa *BASIC*”. Shalawat dan salam senantiasa tercurahkan kepada Rasulullah Muhammad SAW beserta seluruh keluarga dan sahabatnya.

Laporan tugas akhir ini adalah salah satu syarat guna menyelesaikan jenjang kesarjanaan Strata-1 (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penyusun menyadari bahwa dalam penyusunan laporan tugas akhir ini masih terdapat kelemahan dan kekurangan. Untuk itu saran dan kritik membangun dari para pembaca senantiasa diharapkan agar dapat lebih baik lagi di masa yang akan datang.

Selama dalam pelaksanaan tugas akhir dan pembuatan laporan, penyusun telah mendapat bimbingan dan bantuan dari berbagai pihak, untuk itu penyusun memberikan ucapan terima kasih yang sebesar - besarnya kepada :

1. Allah SWT. Atas segala hidayah, barokah, taufiq, ridha' serta rizki Nya yang tiada henti.
2. Kedua orangtuaku, yang selalu melimpahkan kasih sayang yang tulus, doa yang tiada henti-hentinya, serta dukungan maupun pengorbanan yang begitu besar dalam kehidupanku.

3. Bapak Ir. Gumbolo HS., M.Sc, selaku Dekan Fakultas Teknologi Industri.
4. Bapak Yudi Prayudi, S.Si., M.kom., selaku ketua Jurusan Teknik Informatika.
5. Bapak Zainudin Zuhri S.T.,M.I.T, selaku Dosen Pembimbing.
6. Keluarga Besar Kost Enau Jambusari.
7. Semua pihak yang telah turut serta mendukung pelaksanaan dan penyusunan laporan tugas akhir ini.

Semoga laporan tugas akhir ini dapat memberi manfaat bagi kita semua, Amin.

Wassalamu'alaikum Wr.Wb.



Yogyakarta, 27 Oktober 2011

Penyusun

TAKARIR

<i>screenshot</i>	sebuah tampilan jendela dalam sebuah aplikasi
<i>siklus clock</i>	denyutan yang mengkoordinasi/mensinkronisasikan setiap aksi-aksi pada komponen perangkat elektronika
<i>switching</i>	penyambung arus
<i>reset</i>	tombol yang digunakan untuk mengulang
<i>trafficlight</i>	lampu lalu lintas
<i>software</i>	perangkat lunak atau program komputer
<i>variable</i>	masukan data
<i>port</i>	lubang yang digunakan sebagai aliran data
<i>downloader</i>	alat yang digunakan untuk mentransfer data
<i>compile</i>	penyusunan data
<i>peripheral</i>	perangkat/peralatan yang berfungsi sebagai perangkat tambahan

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
PERSEMBAHAN	iv
MOTTO	v
KATA PENGANTAR	vi
TAKARIR	viii
DAFTAR ISI	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metodologi Penelitian	3
1.6.1 Metode pengumpulan data	3
1.6.2 Metode pengembangan sistem	3
1.7 Sistematika Penulisan	3
BAB II LANDASAN TEORI	6
2.1 Mikrokontroler	6
2.1.1 Fitur AVR ATmega8535	6
2.1.2 Konfigurasi pin ATmega8535	7

2.1.3 Register I/O	8
2.2 Bahasa Pemrograman BASIC	10
2.2.6 Struktur Pemilihan	14
2.2.7 Struktur Perulangan.....	16
2.2 LED (light-emitting diode)	18
2.3 Seven Segment Display.....	18
2.4 Dot Matrik 5x7	18
2.5 LCD 16x2.....	19
2.6 Switch.....	19
2.7 Keypad 3x4	20
2.8 Sensor LM35	20
2.9 Downloader K-125R	20
2.10 Kabel Pelangi	21
2.11 Software compiler dan Software downloader	21
2.11.1 Compiler BASCOM-AVR	21
2.11.2 Downloader AVR OSP II.....	21
BAB III METODOLOGI.....	23
3.1 Analisis Kebutuhan	23
3.2 Perancangan Perangkat Keras (<i>hardware</i>).....	24
3.2.1 Rangkaian Mikrokontroler ATmega8535	24
3.2.2 Rangkaian LED.....	25
3.2.3 Rangkaian Traffic Light.....	25
3.2.4 Rangkaian LCD.....	26

3.2.5 Rangkaian Seven Segment.....	26
3.2.6 Rangkaian Dot Matrik.....	27
3.2.7 Rangkaian Sensor Suhu LM35	27
3.2.8 Rangkaian Switch.....	28
3.2.9 Rangkaian Keypad	28
3.3 Perancangan Perangkat Lunak (<i>software</i>).....	29
3.3.1 Perangkat Lunak.....	29
3.3.2 Diagram Alir Rangkaian	29
BAB IV HASIL DAN PEMBAHASAN	40
4.1 Hasil	40
4.1.1 Cara meng- <i>Compile</i> dan men- <i>Download</i> Program.....	42
4.1.2 LED Berjalan Berurutan	43
4.1.3 Trafficlight	44
4.1.4 LCD.....	45
4.1.5 Dotmatrik	46
4.1.6 Seven segment.....	47
4.1.7 Switch dan LED	48
4.1.8 Keypad dan LED.....	49
4.1.9 Interupsi Eksternal	50
4.1.10 Timer Counter	51
4.1.11 PWM.....	53
4.1.12 Serial komunikasi.....	54
4.1.13 ADC dengan sensor suhu LM35	56

4.1.14 EEPROM	57
4.2 Troubleshooting	58
4.3 Analisis Kelebihan dan Kekurangan Sistem	61
4.3.1 Kelebihan	61
4.3.2 Kekurangan	62
BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan	63
5.2 Saran.....	63

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR GAMBAR

Gambar 2. 1 Konfigurasi pin ATmega8535	8
Gambar 2. 2 LED	18
Gambar 2. 3 Seven segment display	18
Gambar 2. 4 Dot matrik 5x7	19
Gambar 2. 5 LCD	19
Gambar 2. 6 Switch.....	19
Gambar 2. 7 Keypad 3x4	20
Gambar 2. 8 Sensor LM35	20
Gambar 2. 9 Alur Pemrograman Mikrokontroler	22
Gambar 3. 1 Skema minimum ATmega8535	24
Gambar 3. 2 Rangkaian LED.....	25
Gambar 3. 3 Rangkaian Traffic Light	25
Gambar 3. 4 Rangkaian LCD.....	26
Gambar 3. 5 Rangkaian seven segment	26
Gambar 3. 6 Rangkaian dot matrik	27
Gambar 3. 7 Rangkaian sensor suhu LM35	27
Gambar 3. 8 Rangkaian switch	28
Gambar 3. 9 Rangkaian keypad	28
Gambar 3. 10 <i>Flowchart</i> led berurutan	29
Gambar 3. 11 <i>Flowchart</i> traffic light	30
Gambar 3. 12 <i>Flowchart</i> LCD	31
Gambar 3. 13 <i>Flowchart</i> dot matrik.....	32
Gambar 3. 14 <i>Flowchart</i> seven segment.....	33
Gambar 3. 15 <i>Flowchart</i> swith dan LED	33
Gambar 3. 16 <i>Flowchart</i> keypad dan LED	34

Gambar 3. 17 <i>Flowchart</i> interupsi eksternal.....	35
Gambar 3. 18 <i>Flowchart</i> timer.....	36
Gambar 3. 19 <i>Flowchart</i> Counter	36
Gambar 3. 20 <i>Flowchart</i> PWM.....	37
Gambar 3. 21 <i>Flowchart</i> serial komunikasi.....	37
Gambar 3. 22 <i>Flowchart</i> ADC.....	38
Gambar 3. 23 <i>Flowchart</i> EEPROM.....	39
Gambar 4. 1 Hasil Aplikasi Mikrokontroler	40
Gambar 4. 2 Kabel Pelangi	40
Gambar 4.3 Schematic dan Rangkaian Minimum ATmega8535	41
Gambar 4.4 Screenshot Compile.....	42
Gambar 4.5 Screenshot Download.....	43
Gambar 4. 6 LED Berjalan Berurutan.....	44
Gambar 4. 7 Trafficlight	45
Gambar 4. 8 LCD.....	46
Gambar 4. 9 Dotmatrik	47
Gambar 4. 10 Seven Segment.....	48
Gambar 4. 11 Switch dan LED	49
Gambar 4. 12 Keypad dan LED.....	50
Gambar 4. 13 Interupsi Eksternal	51
Gambar 4. 14 Timer	52
Gambar 4. 15 Counter.....	53
Gambar 4. 16 PWM	54
Gambar 4. 17 Serial Komunikasi.....	55
Gambar 4. 18 Hyperterminal.....	56
Gambar 4. 19 ADC dengan Sensor Suhu LM35.....	57
Gambar 4. 20 EEPROM.....	58

DAFTAR TABEL

Tabel 2.1 Register I/O	8
Tabel 2.2 Tipe Data.....	10
Table 4.1 Switch dan LED	48
Tabel 4.2 Keypad dan LED.....	49
Tabel 4.3 Serial Komuknikasi.....	54
Tabel 4.4 Troubleshooting	58



BAB I

PENDAHULUAN

1.1 Latar Belakang

Manusia dalam kehidupannya cenderung ingin suatu hal yang serba otomatis tanpa harus sibuk untuk mengeluarkan banyak tenaga dalam menyelesaikan pekerjaannya, maka akhir-akhir ini banyak sekali pemanfaatan tenaga kerja manusia yang semakin sedikit. Tenaga kerja manusia sudah banyak tergantikan oleh mesin atau robot.

Teknologi mikrokontroler adalah salah satu sistem yang mampu melakukan setiap instruksi dalam bentuk program-program yang disimpan dalam chip tunggal sebagai otak kendali atau pemroses untuk melakukan setiap instruksi yang diinginkan.

Aplikasi dasar ini di tujukan untuk pemula yang ingin belajar mikrokontroler. diharapkan pemula dapat dengan mudah mempelajari *port - port* mikrokontroler , *output*, dan *input hardware* seperti *LED*, *segment display*, *dot matrik 5x7*, *LCD*, *switch*, dan *keypad 3x4*. Cara kerja aplikasi ini dengan menuliskan program dan meng-*compile* program sehingga *output hardware* sesuai dengan keinginan pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka masalah yang akan di rumuskan sebagai berikut:

Bagaimana membuat sebuah aplikasi dasar mikrokontroler ATmega8535 dengan bahasa BASCOM, sehingga memudahkan pemula untuk memahami *input* dan *output* dari mikrokontroler itu sendiri maupun dari hardware yang akan digunakan.

1.3 Batasan Masalah

Batasan masalah dalam skripsi ini adalah :

- a. Mengetahui fitur - fitur dan kegunaan *port - port* mikrokontroler Atmega8535 yang terdiri dari 40 *port*.
- b. Menggunakan bahasa BASCOM yang di aplikasikan pada *digital out (LED, 5 segment display, dot matrik 5x7, LCD 2 * 16)* dan *digital in (switch, keypad 3x4, keyboard)*.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah :

- a. Membuat program dan *hardware* agar para pemula yang ingin belajar mikrokontroler dapat belajar dengan mudah.
- b. Mengetahui dasar – dasar dari pembuatan robot.

1.5 Manfaat Penelitian

Penelitian ini akan memiliki beberapa manfaat, yaitu :

- a. Memberikan pengetahuan dasar tentang mikrokontroler Atmega8535..
- b. Memberikan pengetahuan dasar tentang cara membuat robot, khususnya jurusan teknik informatika yang semester ini sudah ada mata kuliah informatika robotika.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan sistem.

1.6.1 Metode pengumpulan data

Pengumpulan data yang diperlukan menggunakan metode sebagai berikut:

1. Observasi

metode pengumpulan data ini digunakan untuk mendapatkan data yang berkaitan tentang mikrokontroler dan bahasa BASIC yang efektif untuk di terapkan pada aplikasi ini.

2. Studi pustaka

Metode ini digunakan untuk mendapatkan informasi tambahan berupa referensi buku-buku literatur yang digunakan sebagai acuan dalam pengembangan sistem.

1.6.2 Metode pengembangan sistem

Metode pengembangan sistem yang digunakan meliputi analisis kebutuhan, perancangan perangkat keras dan perancangan perangkat lunak, implementasi perangkat keras beserta perangkat lunak dan analisis kinerja perangkat keras dan perangkat lunak.

1.7 Sistematika Penulisan

Untuk memudahkan pembahasan penyusunan laporan tugas akhir ini serta memberikan gambaran secara menyeluruh terhadap masalah yang akan dibahas, maka sistematika penulisan laporan tugas akhir akan dibagi menjadi lima bab yaitu :

BAB I. PENDAHULUAN

Bab ini berisi tentang teknologi robot dapat banyak membantu kegiatan manusia. Pada teknologi aplikasi ini menggunakan mikrokontroler ATMega8535 karena mempunyai *fitur* yang lengkap dengan harga terjangkau. Adapun sub-sub bagian tersebut akan dibahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dibagi menjadi dua metode pengumpulan data dan metode pengembangan sistem, dan sistematika penelitian.

BAB II. LANDASAN TEORI

Membahas mengenai uraian dasar teori yang berhubungan dengan objek penelitian tugas akhir ini. Teori tersebut akan digunakan penulis dalam melakukan perancangan, pembuatan aplikasi mikrokontroler Atmega8535 yang menggunakan hardware *digital out (LED, 5 segment display, dot matrik 5x7, LCD 2 * 16)* dan *digital in (switch, keypad 3x4, keyboard)*, dan cara menggunakan *software* secara umum.

BAB III. METODOLOGI

Membahas tentang analisis kebutuhan dan gambaran perancangan dari aplikasi mikrokontroler ATMega8535 yang akan dibuat, baik dari segi rangkaian dan aliran program.

BAB IV. HASIL DAN PEMBAHASAN

Berisi mengenai penguraian hasil dari implementasi pembuatan aplikasi dasar mikrokontroler ATMega8535 berdasarkan kesesuaian antara hasil akhir sistem dengan *konsep* yang dijadikan landasan dalam pengembangan sistem, memuat tampilan *screenshot* dari aplikasi yang telah dibuat. Bagian hasil memuat tentang hasil akhir sistem dan pembahasan memuat tentang hasil aktifitas yang diperoleh.

BAB V. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menguraikan kesimpulan dari penelitian Aplikasi mikrokontroler ATmega8535 serta dijelaskan beberapa saran untuk dilaksanakan lebih lanjut guna pengembangan dan perbaikan penelitian tugas akhir ini.



BAB II

LANDASAN TEORI

2.1 Mikrokontroler

Dalam perancangan aplikasi dasar ini menggunakan mikrokontroler keluarga AVR dari ATMEL. Digunakan mikrokontroler generasi AVR (*Alf and Vegard's RISC processor*) karena mikrokontroler ini memiliki desain sistem dengan teknologi yang memiliki kapabilitas yang amat maju, tetapi dengan biaya ekonomis yang cukup minimal.

Mikrokontroler AVR memiliki RISC (*reduced instruction set computing*) 8 bit, dimana semua instruksi dikemas dalam satu *siklus clock*. Berbeda dengan mikrokontroler generasi sebelumnya, yaitu MCS51 yang hanya menggunakan arsitektur CISC (*complex instruction set computing*) yang membutuhkan 12 *siklus clock*. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATTiny, keluarga AT90Sxx, keluarga ATMega, dan AT86RFxx. Dari semua keluarga AVR tersebut pada dasarnya yang membedakan masing – masing kelas adalah memori, *peripheral* dan fungsinya, sedangkan dari arsitektur dan instruksi yang digunakan semua keluarga AVR dapat dikatakan hampir sama. (Iswanto, 2008:1)

2.1.1 Fitur AVR ATMega8535

Mikrokontroler ATMega8535 memiliki fitur sebagai berikut : (Atmel, 2006)

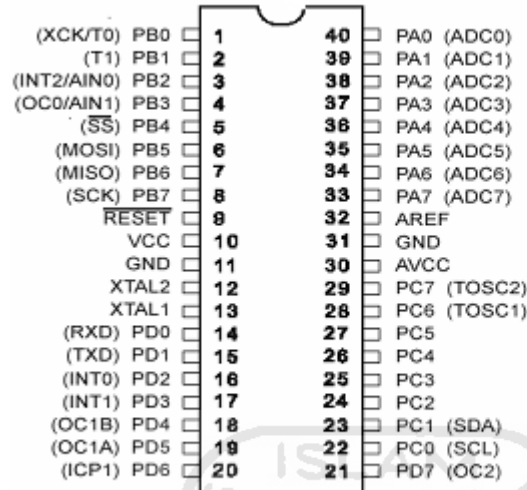
1. Port I/O 32 jalur (*Port A, Port B, Port C, Port D*)
2. ADC 10 bit 8 channel
3. 3 buah timer / counter
4. Osilator internal 1Mhz
5. Memori flash sebesar 8 kb
6. EEPROM 512 byte

7. SRAM 512 byte
8. Interupsi eksternal dan internal
9. *Port* USART untuk serial komunikasi

2.1.2 Konfigurasi pin ATmega8535

Konfigurasi pin dari mikrokontroler ATmega8535 sebanyak 40 pin dapat dilihat pada Gambar 2.1. Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut: (Iswanto, 2008:5)

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. *Port* A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. *Port* B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI.
5. *Port* C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog dan *Timer Oscillator*.
6. *Port* D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
7. *RESET* merupakan pin yang digunakan untuk *me-reset* mikrokontroler .
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.



Gambar 2. 1 Konfigurasi pin ATmega8535

2.1.3 Register I/O

Register I/O berfungsi untuk mengatur kontrol *peripheral* mikrokontroler. Definisi masing-masing *register* sebagai berikut: (Inkubator, 2008:4)

Tabel 2.1 Register I/O

addres	nama	Fungsi
\$3F	SREG	Status register
\$3E	SPH	Stack pointer <i>high</i>
\$3D	SPL	Stack pointer <i>low</i>
\$3C	OCR0	Ouput compare register (0)
\$3B	GICR	General interrupt <i>control</i> register
\$3A	CIFR	General interrupt flag <i>control</i>
\$39	TIMSK	Timer/counter interrupt mask register
\$38	TIFR	Timer/counter interrupt flag register
\$73	SPMCR	Store program memory <i>control</i> register
\$36	TWCR	Twi <i>control</i> register

\$35	MCUCR	Mcu <i>control</i> register
\$34	MCUCSR	Mcu <i>control dan status</i> register
\$33	TCCR0	Timer/counter <i>control</i> register
\$32	TCNT0	Timer/counter register
\$31	OSCCAL	Oscillator <i>calibration</i> register
\$30	SFIOR	Special function I/O register
\$2F	TCCR1A	Timer/counter 1 <i>control</i> register A
\$2E	TCCR1B	Timer/counter 1 <i>control</i> register B
\$2D	TCNT1H	Timer/counter 1 <i>high</i>
\$2C	TCNT1L	Timer/counter 1 <i>low</i>
\$2B	OCR1AH	Output compare register 1A <i>high</i>
\$2A	OCR1AL	Output compare register 1A <i>low</i>
\$29	OCR1BH	Output compare register 1B <i>high</i>
\$28	OCR1BL	Output compare register 1B <i>low</i>
\$27	ICR1H	Input capture register 1 <i>high</i>
\$26	ICR1L	Input capture register 1 <i>low</i>
\$25	TCCR2	Timer counter 2 <i>control</i> register
\$24	TCNT2	Timer/counter 2 register
\$23	OCR2	Output <i>control</i> register 2
\$22	ASSR	Asynchronous status register
\$21	WDTCSR	Watchdog timer <i>control</i> register
\$1F	EEARH	EEPROM address register, <i>high</i>
\$1E	EEARL	EEPROM address register, <i>low</i>
\$1D	EEDR	EEPROM data register
\$1C	EEDR	EEPROM <i>control</i> register
\$1B	PORTA	Data register, port A
\$1A	DDRA	Data <i>direction</i> register, portA
\$19	PINA	Input pin, portA
\$18	PORTB	Data register, port B
\$17	DDRB	Data <i>direction</i> register, portB
\$16	PINB	Input pin, portB
\$15	PORTC	Data register, port C
\$14	DDRC	Data <i>direction</i> register, portC
\$13	PINC	Input pin, portC
\$12	PORTD	Data register, port D
\$11	DDRD	Data <i>direction</i> register, portD
\$10	PIND	Input pin, portD
\$0F	SPDR	SPI data register
\$0E	SPSR	SPI status register
\$0D	SPCR	SPI <i>control</i> register
\$0C	UDR	USART I/O data register

\$0B	UCSRA	USART <i>control & status register A</i>
\$0A	UCSRB	USART <i>control & status register B</i>
\$09	UBRAL	USART baud rate register <i>low</i>
\$08	ACSR	Analog comparator <i>control & status register</i>
\$07	ADMUX	ADC multiplexer <i>selection</i>
\$06	ADCSRA	ADC <i>control & status register A</i>
\$05	ADCH	ADC data register <i>high</i>
\$04	ADCL	ADC data register <i>low</i>
\$03	TWDR	TWI data register
\$02	TWAR	TWI address register
\$01	TWSR	TWI status register
\$00	TWBR	TWI bit rate register

2.2 Bahasa Pemrograman BASIC

Salah satu bahasa yang digunakan pada program mikrokontroler adalah bahasa *BASIC*. Bahasa *BASIC* banyak digunakan untuk aplikasi mikrokontroler karena kemudahan dan kompetibel terhadap mikrokontroler jenis AVR dan di dukung oleh *compiler software* berupa BASCOM-AVR. (Fahmizal, 2010)

2.2.1 Tipe Data

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efisien dan efektif.

Tabel 2.2 Tipe Data

No	Tipe data	Ukuran (byte)	Jangkauan
1	Bit	1/8	0 atau 1
2	Byte	1	0 s/d 255
3	Integer	2	-32.768 s/d 32.767
4	Word	2	0 s/d 65535
5	Long	4	-2147483648 s/d 2147483647
6	Single	4	$1.5 \times 10^{-45} - 3.4 \times 10^{38}$
7	Double	8	$5.0 \times 10^{-324} \text{ to } 1.7 \times 10^{308}$

8	String	s/d 254	-
---	--------	---------	---

2.2.2 Konstanta

Konstanta merupakan suatu nilai dengan tipe data tertentu yang tidak dapat diubah-ubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program.

Contoh : $Kp = 35$, $Ki=15$, $Kd=40$

2.2.3 Variabel

Variabel adalah suatu pengenal (*identifier*) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari variable terserah sesuai dengan yang diinginkan namun hal yang terpenting adalah setiap variabel diharuskan :

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, max 32 karakter.
2. Tidak boleh mengandung spasi atau symbol-simbol khusus seperti : \$, ?, %, #, !, &, *, (,), -, +, = dan lain sebagainya kecuali *underscore*.

2.2.4 Deklarasi

Deklarasi sangat diperlukan bila akan menggunakan pengenal (*identifier*) dalam suatu program.

1. Deklarasi Variabel

Bentuk umum pendeklarasian suatu variable adalah **Dim** *nama_variabel* **As** *tipe_data*

Contoh : **Dim** *x* **As** *Integer* 'deklarasi *x* bertipe *integer*

2. Deklarasi Konstanta

Dalam Bahasa Basic konstanta di deklarasikan langsung.

Contoh : *S* = "Hello world" **Assign string**

3. Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat dipanggil di manapun di dalam program. Fungsi dalam Bahasa Basic ada yang sudah disediakan sebagai fungsi pustaka seperti print, input data dan untuk menggunakannya tidak perlu dideklarasikan.

4. Deklarasi Buatan

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah :

Sub Test (**byval** *variabel* **As** *type*)

Contoh : **Sub Pwm**(**byval** *Kiri* **As** *Integer* , **Byval** *Kanan* **As** *Integer*)

2.2.5 Operator

1. Operator Penugasan

Operator Penugasan (*Assignment operator*) dalam Bahasa Basic berupa “=”.

2. Operator Aritmatika

a. * : untuk perkalian

b. / : untuk pembagian

c. + : untuk penambahan

d. - : untuk pengurangan

e. % : untuk sisa pembagian (modulus)

3. Operator Hubungan (Perbandingan)

Operator hubungan digunakan untuk membandingkan hubungan dua buah operand atau sebuah nilai / variable, misalnya :

a. = 'Equality $X = Y$

b. < 'Less than $X < Y$

c. > 'Greater than $X > Y$

d. <= 'Less than or equal to $X <= Y$

e. >= 'Greater than or equal to $X >= Y$

4. Operator Logika

Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada empat macam, yaitu :

a. NOT *'Logical complement*

b. AND *'Conjunction*

c. OR *'Disjunction*

d. XOR *'Exclusive or*

5. Operator Bitwise

Operator bitwise digunakan untuk memanipulasi bit dari data yang ada di memori. Operator bitwise dalam Bahasa Basic :

a. Shift A, Left, 2 : *Pergeseran bit ke kiri*

b. Shift A, Right, 2 : *Pergeseran bit ke kanan*

c. Rotate A, Left, 2 : *Putar bit ke kiri*

d. Rotate A, right, 2 : *Putar bit ke kanan*

2.2.6 Struktur Pemilihan

1. IF-THEN

Merupakan pernyataan untuk menguji apakah kondisi bernilai benar atau salah untuk melakukan sebuah intruksi. Syntak penulisannya:

If <kondisi> **then** <perintah>

If <kondisi> **then**

<perintah 1>

<perintah 2>

End if

2. IF –THEN –ELSE

Untuk keadaan dimana kedua kondisi (benar maupun salah) tetap dikenai perintah. Syntax penulisannya:

If <kondisi> **then**

<perintah 1>

Else

<perintah 2>

End if

3. IF-THEN-ELSEIF

Digunakan ketika terdapat lebih dari satu pengujian kondisi. Syntax penulisannya sebagai berikut:

If <kondisi 1> **then**

<perintah 1>

Elseif <kondisi 2> **then**

<perintah 2> **then**

Elseif <kondisi 3> **then**

<perintah 3>

End if

4. SELECT-CASE

Digunakan untuk pengujian kondisi yang banyak, maka akan lebih sederhana menggunakan *select-case*. Cara penulisannya:

Select case <variable>

Case 1 : <perintah 1>

Case 2: <perintah 2>

End select

2.2.7 Struktur Perulangan

1. FOR-NEXT

Perintah ini digunakan untuk melaksanakan perintah secara berulang.

Syntax penulisannya:

For <variable=nilai awal> **to** <nilai akhir> <step penambahan>

<pernyataan>

Next

2. DO-LOOP

Pernyataan ini melakukan perulangan selama kondisi terpenuhi. Syntax penulisannya:

Do

<pernyataan>

Loop

Jika perulangan terbatas, sesuai kondisi yang diinginkan, maka caranya sebagai berikut:

Do

<pernyataan>

Loop until <kondisi>

3. WHILE-WEND

Bentuk perulangan jika sebuah syarat kondisi terpenuhi. Syntax penulisannya:

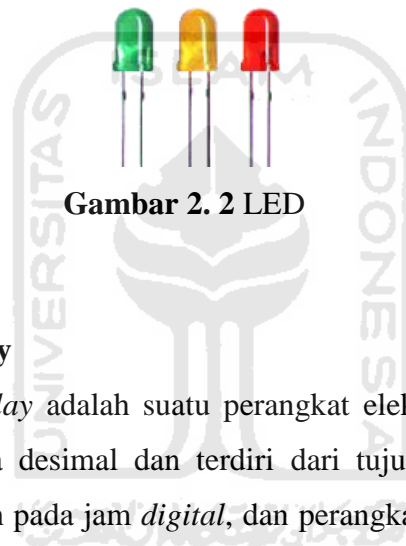
While <kondisi>

<perintah>

Wend

2.2 LED (light-emitting diode)

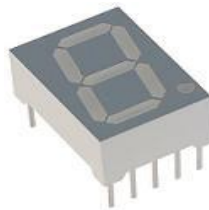
LED adalah Suatu lampu indikator dalam perangkat elektronika yang biasanya memiliki fungsi untuk menunjukkan status dari perangkat elektronika tersebut. Lampu LED terbuat dari plastik dan *dioda semi-konduktor* yang dapat menyala apabila dialiri tegangan listrik rendah (sekitar 1.5 volt DC). Berbagai macam warna dan bentuk dari lampu LED, disesuaikan dengan kebutuhan dan fungsinya. (Wik, 2011)



Gambar 2. 2 LED

2.3 Seven Segment Display

Seven segment display adalah suatu perangkat elektronik yang kegunaannya untuk menampilkan angka desimal dan terdiri dari tujuh elemen. *Seven segment display* biasanya digunakan pada jam *digital*, dan perangkat elektronik lainnya untuk menampilkan informasi *numerik*. (Wik, 2011)

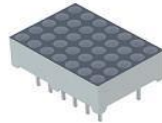


Gambar 2. 3 Seven Segment Display

2.4 Dot Matrik 5x7

Dot matrik 5x7 memiliki arti bahwa satu dot matrik berukuran 5 kolom x 7

baris susunan LED. Jadi satu dot matrik terdapat 35 buah LED. Biasa digunakan untuk membuat tulisan berjalan baik kanan kiri atau sebaliknya maupun atas bawah atau sebaliknya. (Wik, 2011)



Gambar 2. 4 Dot Matrik 5x7

2.5 LCD 16x2

LCD 16x2 memiliki arti bahwa di dalam LCD terdapat 16 kolom x 2 baris. Lcd ini dapat menampilkan campuran huruf dan angka sekaligus. Jika tidak ada *icon* atau huruf pada keyboard dapat menggunakan LCD design untuk membuatnya. (Wik, 2011)



Gambar 2. 5 LCD

2.6 Switch

Sebuah tombol saklar yang membuka atau menutup sirkuit listrik. (Wik, 2011)



Gambar 2. 6 Switch

2.10 Kabel Pelangi

Kabel pelangi terdiri dari 10 buah kabel yang di jadikan satu. Kabel ini akan digunakan sebagai penghubung rangkaian mikrokontroler dengan rangkaian *hardware* yang akan dibuat. (Wik, 2011)



Gambar 2.9 Kabel Pelangi

2.11 Software compiler dan Software downloader

Bahasa BASIC merupakan salah satu bahasa yang cukup populer dan handal untuk pemrograman mikrokontroler . Dalam melakukan pemrograman mikrokontroler di perlukan *software* pemrograman, salah satunya yang mendukung bahasa BASIC adalah BASCOM–AVR.

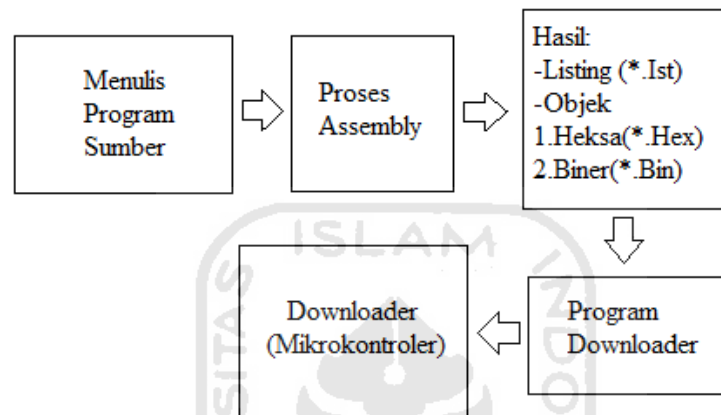
2.11.1 Compiler BASCOM-AVR

Buat lembar kerja baru, *file -> new* (CTRL+N), kemudian buat *listing* program dengan bahasa BASIC. Setelah selesai simpan dengan cara *file -> save* (CTRL+S). *compile* program tersebut *program -> compile* (F7). Jika terdapat *error*, maka perbaiki lagi program kemudian *compile* sampai tidak ada tanda *error*.

2.11.2 Downloader AVR OSP II

AVR OSP dipilih karena cukup mudah penggunaannya dan dapat beroperasi tanpa menginstalnya. Cara menggunakannya pertama-tama buka AVR OSP II, sebelum di gunakan AVR OSP II harus di *setting* terlebih dahulu, cara *setting* pilih

configure -> lakukan *setting port* sesuai *port* yang terdeteksi dan *baudrate* pilih *115200bps*. Setelah melakukan *setting* kembali ke menu *program* dan tekan *auto detect* untuk mengetahui *downloader* siap digunakan. Tekan *browse* untuk memilih *file *.hex* yang ingin di *download* kemudian tekan *program*.



Gambar 2.10 Alur Pemrograman Mikrokontroler

BAB III

METODOLOGI

3.1 Analisis Kebutuhan

Dalam pembuatan aplikasi dasar mikrokontroler ini membutuhkan perangkat keras (*hardware*), dan perangkat lunak (*software*). Perangkat keras yang dibutuhkan meliputi mikrokontroler ATmega8535 sebagai otak kendali, 17 LED, 5 switch, 2 seven segment, 1 dotmatrik 5x7, 1 LCD 16*2, 1 keypad 3x4, 1 LM35 dan *downloader serial k-125R*. Perangkat keras tersebut akan di rangkai sehingga menjadi satu kesatuan agar dapat dioperasikan melalui mikrokontroler ATmega8535 sebagai otak kendalinya. *Downloader serial k-125R* bertugas sebagai penghubung antara perangkat keras dan perangkat lunak. Adapun piranti elektronika yang wajib dimiliki seperti *multimeter, solder, dan cutter*.

Setelah perangkat keras terkumpul maka selanjutnya ke tahap perancangan perangkat keras (*hardware*) yang terdiri dari rangkaian mikrokontroler ATmega8535, rangkaian LED, rangkaian Traffic Light, rangkaian LCD, rangkaian Seven Segment, rangkaian Dot Matrik, rangkaian sensor suhu LM35, rangkaian switch, rangkaian Keypad.

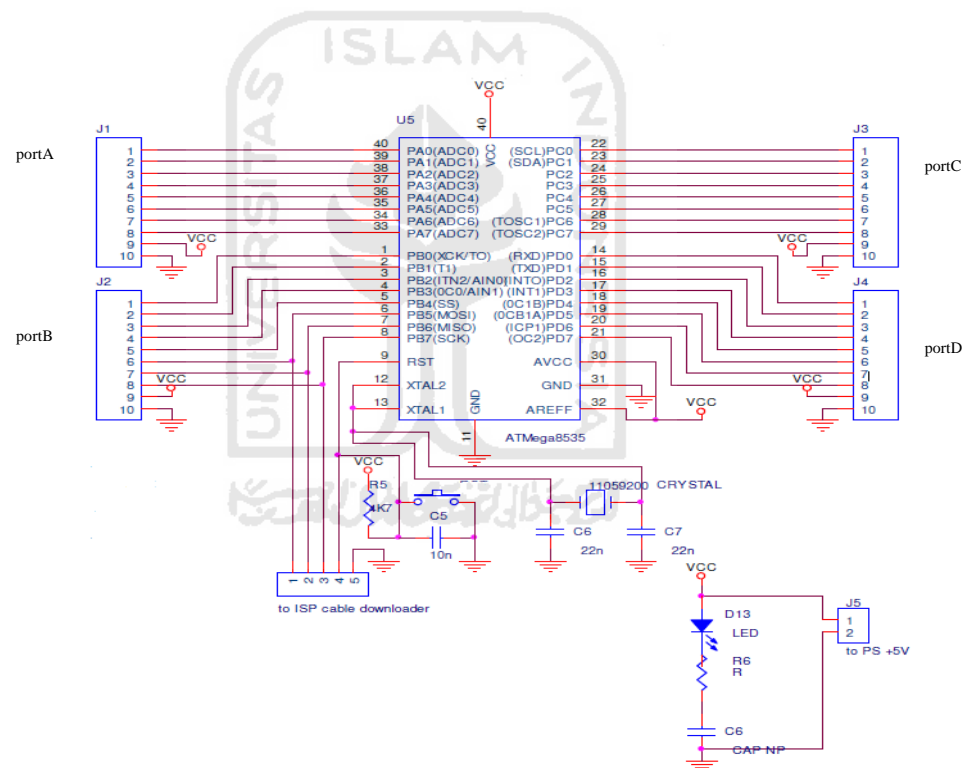
Tahap selanjutnya perancangan perangkat lunak (*software*) terdiri dari perangkat lunak, dan rangkaian diagram alir. Perangkat lunak yang digunakan *Compiler BASCOM-AVR*, dan *Downloader AVR OSP II*. Rangkaian diagram alir akan menentukan *script program* yang akan dibuat dengan melihat rangkaian perangkat keras terlebih dahulu. Rangkaian diagram alir terdiri dari, *flowchart* Led Berjalan Berurutan, *flowchart* Traffic Light, *flowchart* LCD, *flowchart* Dot Matrik, *flowchart* Seven Segment, *flowchart* Switch dan *flowchart* LED, *flowchart* Keypad dan LED, *flowchart* Interupsi Eksternal, *flowchart* Timer Counte, *flowchart* PWM (Pulse Width Modulation), *flowchart* Serial Komunikasi, *flowchart* ADC dengan Sensor Suhu Lm35 dan *flowchart* EEPROM.

3.2 Perancangan Perangkat Keras (*hardware*)

Pada perancangan perangkat keras ini akan di bahas bagaimana merangkai *hardware-hardware* yang di perlukan dalam pengerjaan aplikasi mikrokontroler ini.

3.2.1 Rangkaian Mikrokontroler ATmega8535

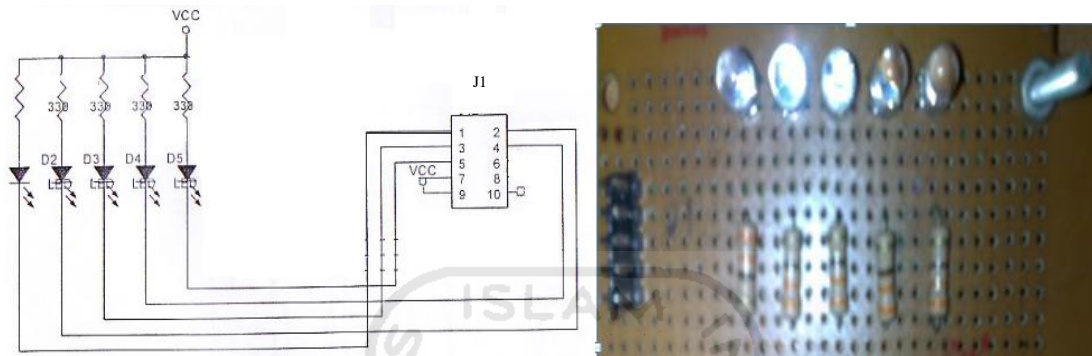
Rangkaian ini menggunakan ATmega8535 yang digunakan sebagai *minimum system*. Rangkaian ini berfungsi sebagai otak yang mengatur jalannya rangkaian secara keseluruhan.



Gambar 3. 1 Skema minimum ATmega8535

3.2.2 Rangkaian LED

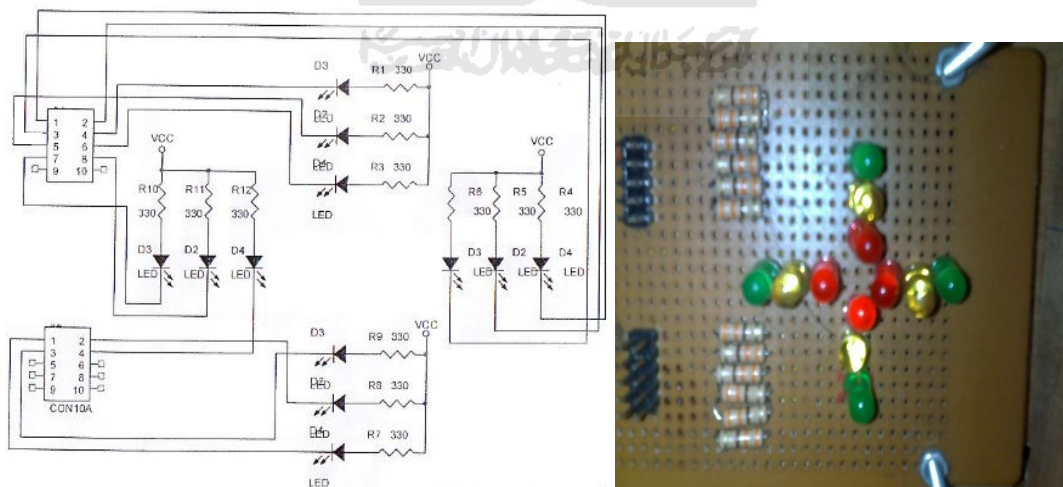
LED ini akan digunakan untuk *output* dari salah satu *port* (misalnya *port A* saja), *output* dari switch, *output* dari keypad, *output* dari komunikasi serial, dan akan digunakan untuk simulasi pada fitur PWM.



Gambar 3. 2 Rangkaian LED

3.2.3 Rangkaian Traffic Light

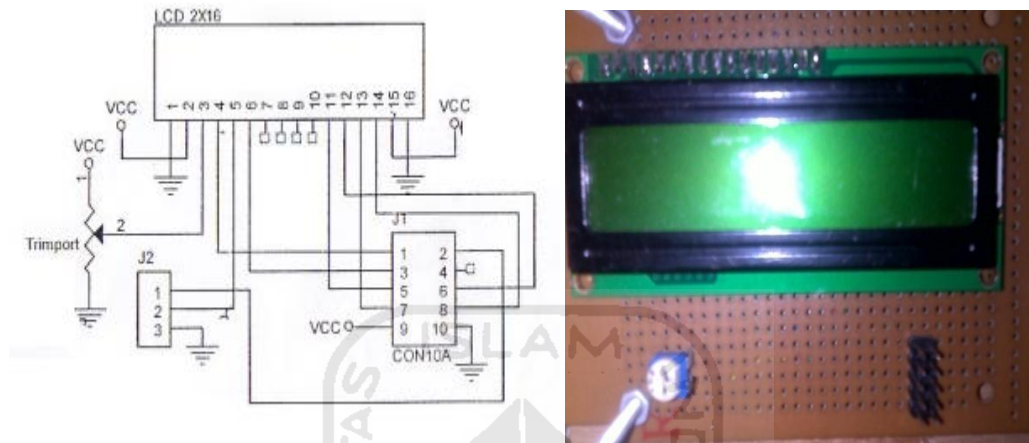
LED ini digunakan untuk *output* dari dua *port* (misalnya *port A*, dan *port B*), dan akan disimulasikan seperti *trafficlight* yang biasa kita lihat sehari – hari.



Gambar 3. 3 Rangkaian Trafic Light

3.2.4 Rangkaian LCD

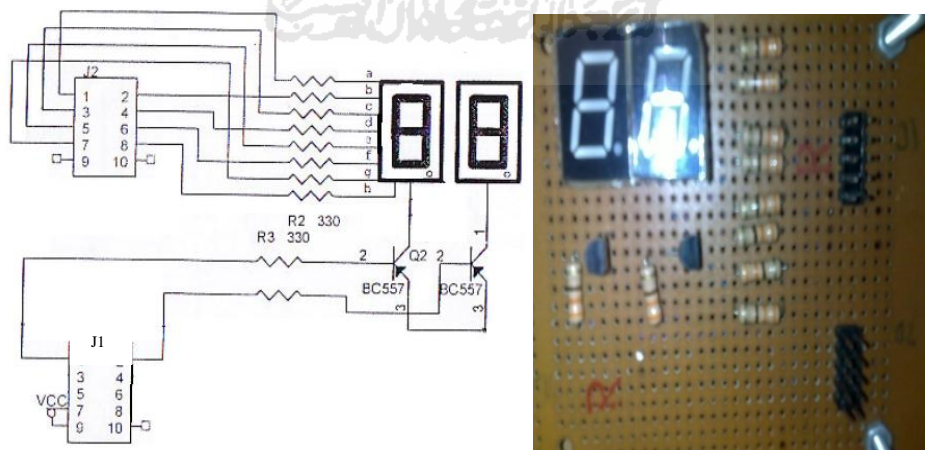
Disini kita akan bahas tentang cara menampilkan tulisan ke LCD, dan akan digunakan untuk menampilkan *timer dan counter* dan *ADC*.



Gambar 3. 4 Rangkaian LCD

3.2.5 Rangkaian Seven Segment

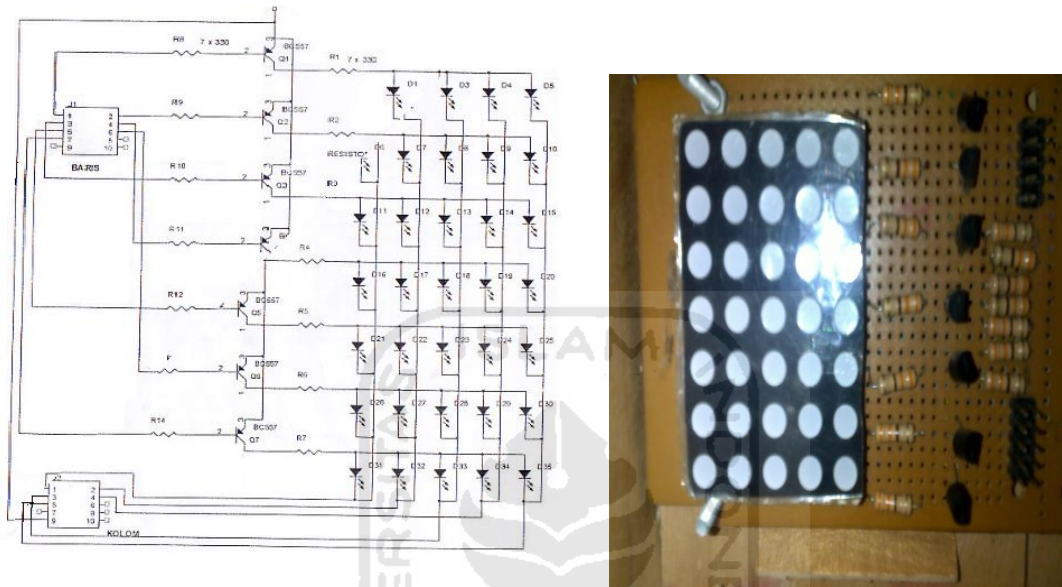
Disini kita akan bahas tentang cara menampilkan seven segment yang terdiri dari dua buah seven segment.



Gambar 3. 5 Rangkaian Seven Segment

3.2.6 Rangkaian Dot Matrik

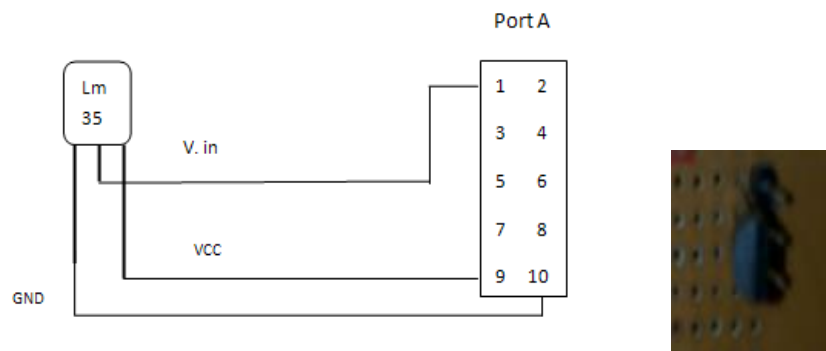
Pada rangkaian dot matrik kita akan bahas tentang cara menampilkan salah satu huruf abjad dan menampilkan simulasi tulisan berjalan.



Gambar 3. 6 Rangkaian Dot Matrik

3.2.7 Rangkaian Sensor Suhu LM35

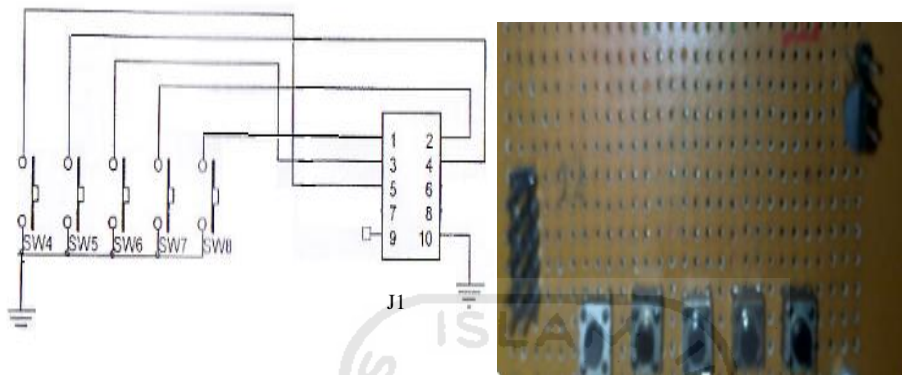
Sensor suhu ini akan digunakan untuk menampilkan fitur ADC (*Analog-to-digital Converter*) pada LCD.



Gambar 3. 7 Rangkaian Sensor Suhu LM35

3.2.8 Rangkaian Switch

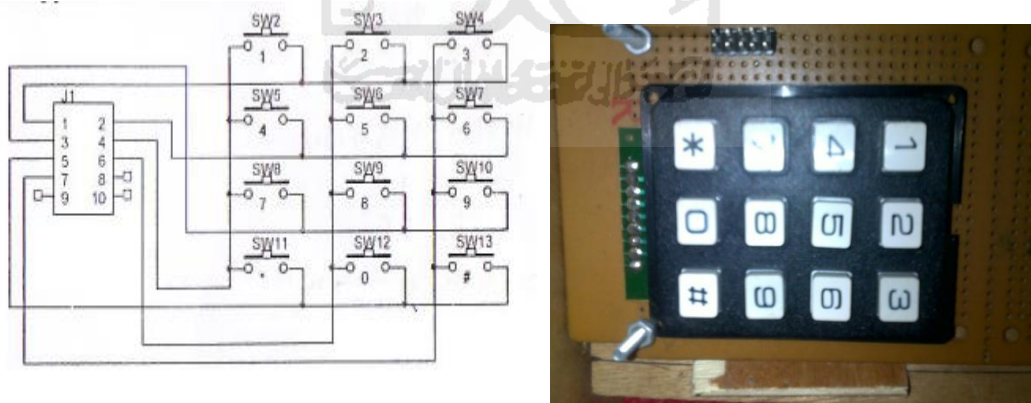
Switch ini akan digunakan *input* yang *output* akan di lihat pada LED dan akan digunakan juga pada fitur interupsi eksternal.



Gambar 3. 8 Rangkaian Switch

3.2.9 Rangkaian Keypad

Pada rangkaian keypad akan dibahas cara menggunakan keypad sebagai *input* yang *outputnya* akan di lihat pada LED.



Gambar 3. 9 Rangkaian Keypad

3.3 Perancangan Perangkat Lunak (*software*)

Pada perancangan perangkat lunak ini membahas tentang *software – software* yang digunakan dan membahas aliran data yang akan dibuat pada aplikasi mikrokontroler ini.

3.3.1 Perangkat Lunak

1. *Compiler* BASCOM-AVR

Aplikasi ini digunakan untuk menulis program yang akan dibuat yang akan disimpan dalam ekstensi *.bas, kemudian dapat meng-*compile* menjadi ekstensi *.hex.

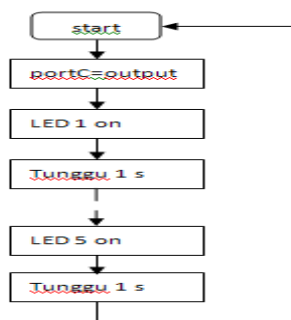
2. *Downloader* AVR OSP II

Aplikasi ini digunakan untuk *download listing* program yang *berekstensi* *.hex yang sudah di *compile* dari BASCOM-AVR ke mikrokontroler .

3.3.2 Diagram Alir Rangkaian

pada diagram alir ini sebagai langkah awal yang akan menerangkan bagaimana alur dari program yang akan digunakan pada aplikasi mikrokontroler.

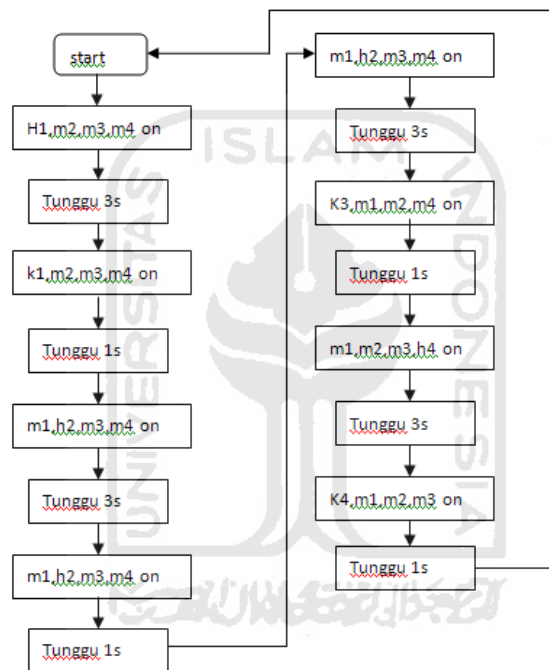
1. Led Berjalan Berurutan



Gambar 3. 10 *Flowchart* Led Berurutan

Menentukan inisialisasi *port output* pada LED, kemudian mengatur program agar LED berjalan berurutan dari LED 1 sampai dengan LED 5. Ketika LED 1 menyala, dibutuhkan satu detik untuk LED berikutnya dapat menyala. Setelah LED 5 menyala maka akan kembali lagi ke LED 1 dan seterusnya.

2. Traffic Light

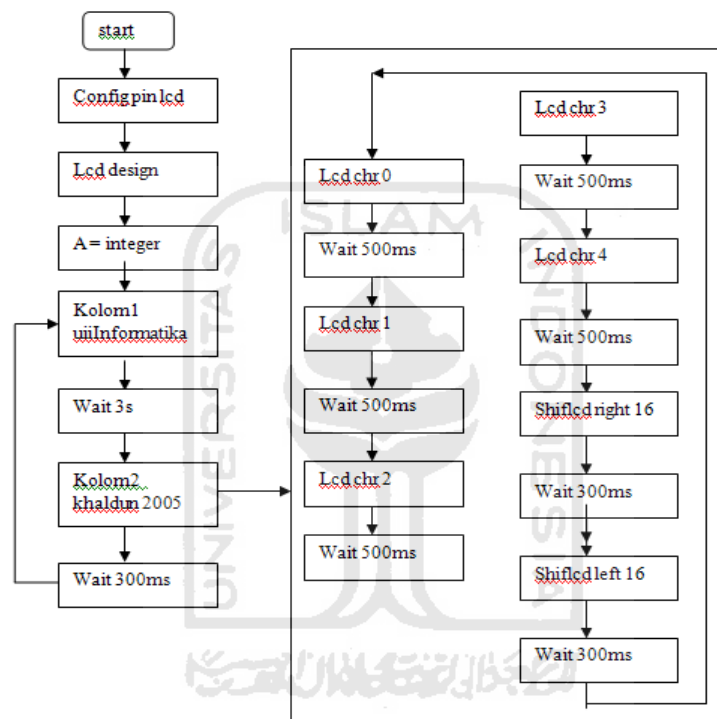


Gambar 3. 11 *Flowchart Trafficlight*

Pada gambar 3.11 diterangkan H sebagai hijau, M sebagai merah, dan K sebagai kuning. Ada 4 deret lampu *trafficlight*, ketika deret pertama LED hijau menyala maka deret yang lain LED merah yang akan menyala demikian pula ketika LED kuning pada deret pertama menyala maka deret yang lainnya tetap LED merah yang menyala demikian seterusnya setelah berganti deret. Dibutuhkan waktu 3 detik

dari LED hijau ke kuning, dan dari kuning ke LED hijau deret selanjutnya membutuhkan waktu 1detik.

3. LCD

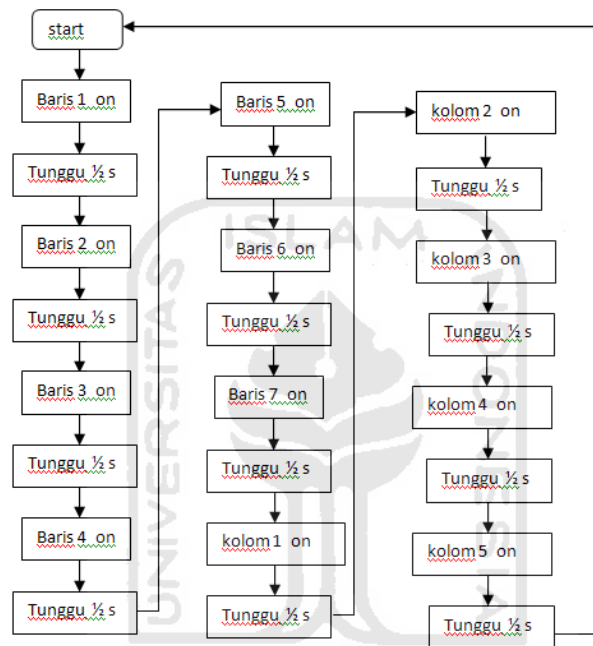


Gambar 3. 12 Flowchart LCD

Pertama-tama menentukan pin LCD yang dihubungkan dengan mikrokontroler . Masukan LCD design yang sudah dibuat kemudian tentukan *variable* yang akan di gunakan. Waktu yang dibutuhkan antar kolom satu ke kolom dua adalah 3 detik. Setelah itu LCD akan di geser ke kanan sebanyak 16 baris tunggu 0,3 detik dan LCDdi gesere kembali ke kiri sebanyak 16 baris, kemudian tunggu 0,3 detik dan kembali ke kolom satu kembali. didalam kolom dua terdapat *script* animasi

yang seolah-olah pada tampilan LCD terlihat bergerak dan berjalan. Antara karakter satu dengan lainnya di beri waktu 0,5 detik agar karakter terlihat hidup atau bergerak.

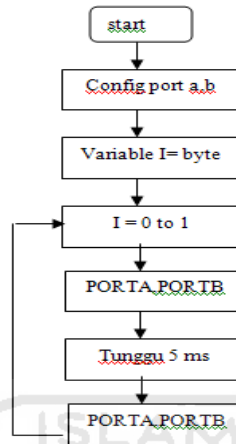
4. Dot Matrik



Gambar 3. 13 Flowchart Dot Matrik

Baris 1 menyala tunggu 0,5 detik baris 2 menyala dan seterusnya sampai baris ke 7. Setelah baris 7 menyala tunggu 0,5 detik dan kolom 1 menyala sampai seterusnya ke kolom 5 dengan waktu 0,5 detik pula dan akan kembali ke baris 1 dan seterusnya.

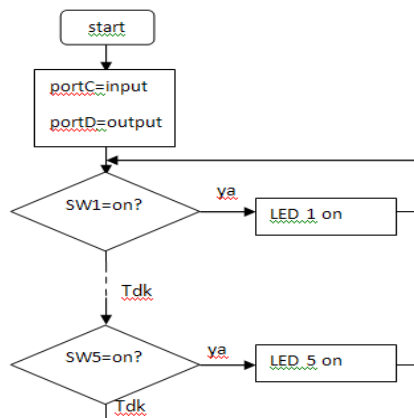
5. Seven Segment



Gambar 3. 14 Flowchart Seven Segment

Menentukan port yang akan digunakan dan menentukan *variable*. Waktu yang dibutuhkan dari PORTA.1, PORTB. ke PORTA.2, PORTB adalah 5 ms, sehingga LED pada seven segment akan terlihat seperti menyala bersamaan.

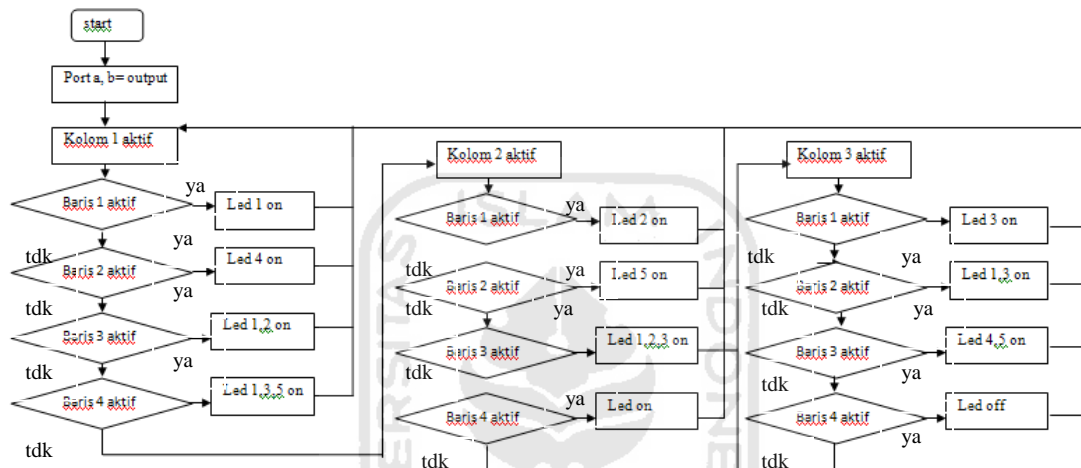
6. Switch dan LED



Gambar 3. 15 Flowchart Swith dan LED

Menentukan *port input* dan *output*, *input* pada switch dan *output* pada LED. Ketika switch 1 menyala, maka LED 1 menyala, jika tidak akan di teruskan ke switch 2. Switch 2 menyala maka LED 2 menyala dan seterusnya sampai switch 5.

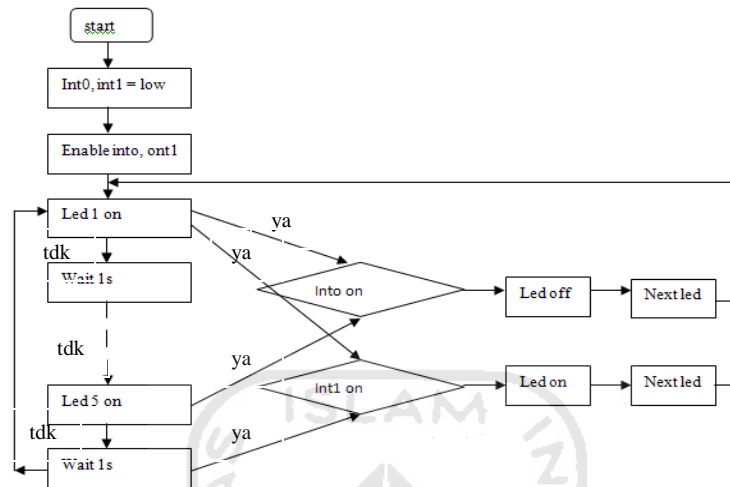
7. Keypad dan LED



Gambar 3. 16 Flowchart Keypad dan LED

Ketika kolom 1 aktif, baris 1 aktif LED 1 *on*, jika tidak akan mengecek kebaris 2. Baris 2 aktif maka LED 4 *on*, jika tidak akan mengecek ke baris 3. Baris 3 aktif maka LED 1,2 *on*, jika tidak akan mengecek ke baris 4 dan seterusnya sampai ke kolom 3.

8. Interupsi Eksternal



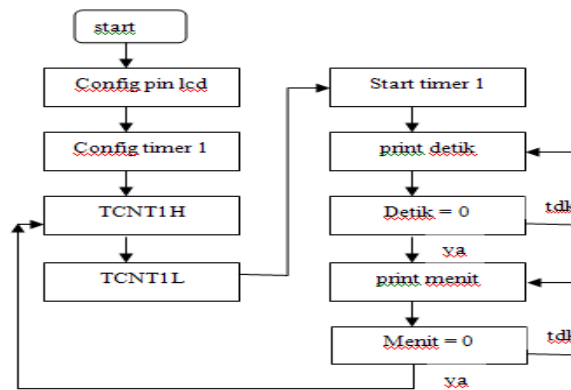
Gambar 3. 17 Flowchart Interupsi Eksternal

Interupsi di *setting* pada posisi *low*, kemudian mengaktifkan *register interupsi*. LED 1 akan menyala berurutan sampai ke LED 5 dan waktu yang dibutuhkan 1 detik antara LED 1 dengan LED lainnya. Di misalkan ketika LED 3 menyala dan *int0* di aktifkan maka semua LED akan mati, sebaliknya ketika *int1* di aktifkan maka semua LED akan menyala. Ketika *int0* maupun *int1* di non-aktifkan maka LED 4 akan menyala kembali berurutan ke LED 5 dan kembali ke LED 1 dan seterusnya.

9. Timer Counter

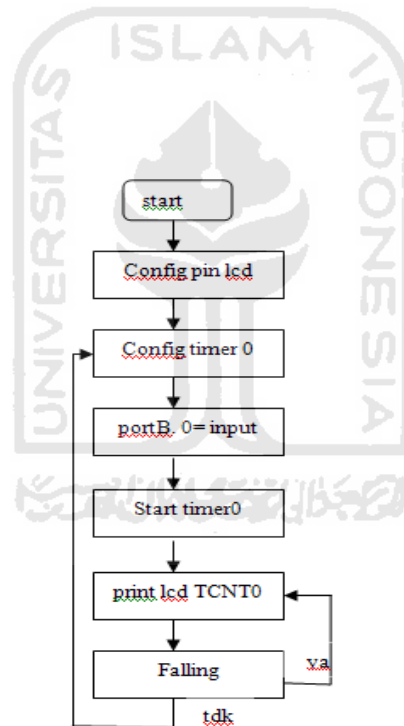
a. Timer

Pada gambar 3.18, pin di konfigurasi terlebih dahulu kemudian menentukan *TCNT1H(high)*, *TCNT1L(low)*. *Timer* dijalankan dari 1 detik sampai 60 detik, ketika pada 60 detik maka detik akan kembali ke 0 dan menit akan bertambah 1.



Gambar 3. 18 Flowchart Timer

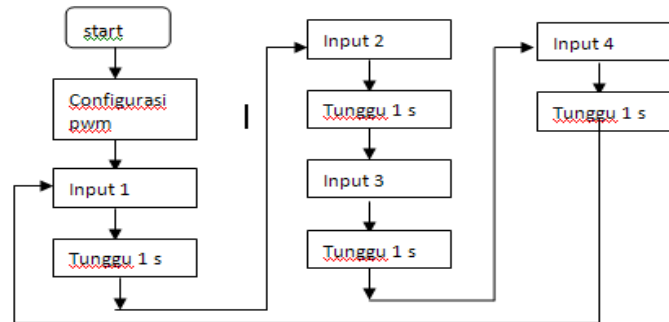
b. Counter



Gambar 3. 19 Flowchart Counter

Pin pada LCD dan timer di konfigurasi dan PORTB.0 sebagai *input*. Ketika timer di jalankan akan di *print* ke LCD. Ketika PORTB.0 di tekan pada posisi *falling* maka timer akan bertambah, jika tidak maka tidak akan terjadi apapun.

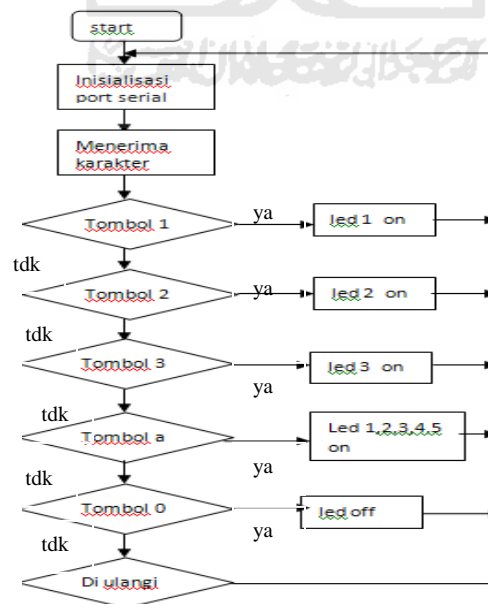
10. PWM (Pulse Width Modulation)



Gambar 3. 20 Flowchart PWM

Pin pwm di konfigurasi terlebih dahulu kemudian input 1 di jalankan kemudian tunggu 1 detik dan input 2 akan berjalan dan seterusnya sampai input 4 dan akan kembali lagi ke input 1.

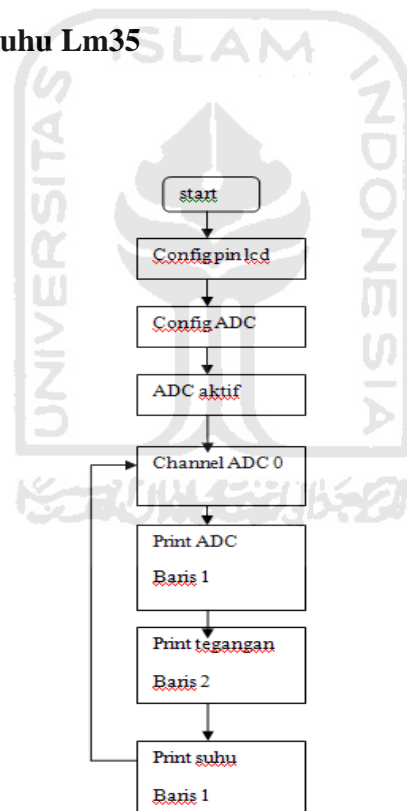
11. Serial Komunikasi



Gambar 3. 21 Flowchart Serial Komunikasi

Menentukan *port serial* kemudian mengkoneksikan antara mikrokontroler dengan PC/laptop. tombol yang digunakan menggunakan keyboard. Ketika tombol 1 di tekan maka LED 1 *on*, jika tidak akan mengecek ke tombol 2. Tombol 2 di tekan LED 2 *on*, jika tidak akan mengecek ke tombol 3. Tombol 3 di tekan LED 3 *on*, jika tidak akan mengecek ke tombol a. Tombol a di tekan LED 1,2,3,4,5 *on*, jika tidak akan mengecek ke tombol 0. Tombol 0 di tekan LED *off*, jika tidak akan mengecek kembali ke tombol 1 dan seterusnya.

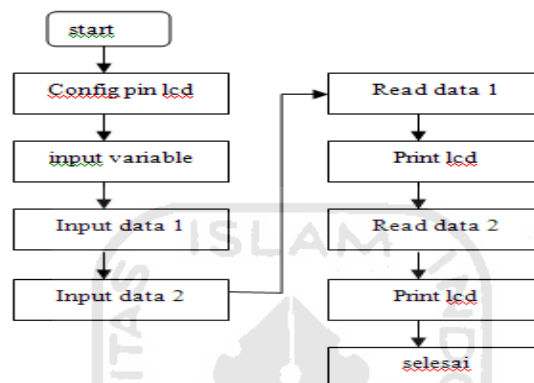
12. ADC dengan Sensor Suhu Lm35



Gambar 3. 22 Flowchart ADC

Pertama-tama menentukan pin LCD yang dihubungkan dengan mikrokontroler dan menentukan *setting* ADC. Kemudian ADC diaktifkan dan dipilih ADC 0. Hasil dari ADC akan di tampilkan pada LCD.

13. EEPROM



Gambar 3. 23 Flowchart EEPROM

Pertama-tama menentukan pin LCD yang dihubungkan dengan mikrokontroler dan menentukan *variable input*. Kemudian masukan data 1 dan 2, ketika di *read* data maka data akan ditampilkan ke LCD.

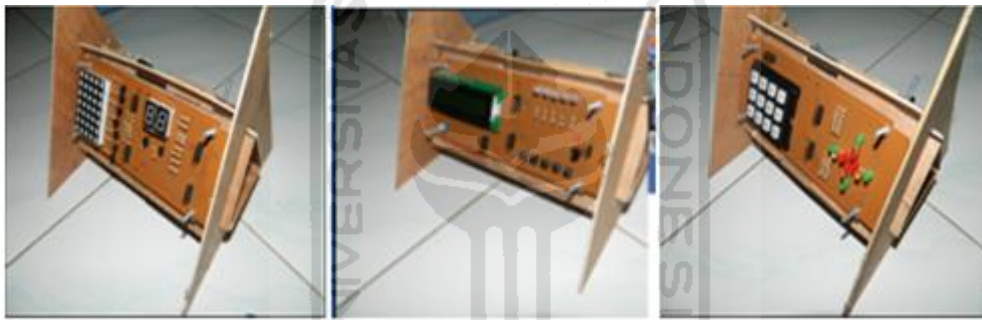
BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

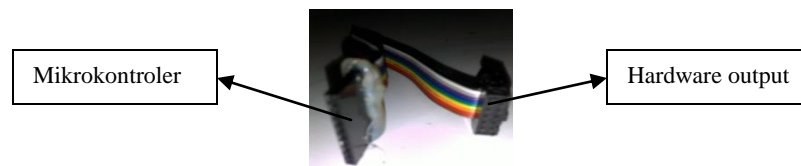
Hasil dari aplikasi mikrokontroler ini akan ditampilkan *output* dari hardware itu sendiri yang terdiri dari led berjalan berurutan, *trafficlight*, *LCD*, *seven segment*, *dot matrik*, *switch dan LED*, *keypad dan LED*, *interupsi eksternal*, *timer counter*, *PWM*, *serial komunikasi*, *ADC dengan Sensor Suhu Lm35*, *EEPROM*.

Dapat kita lihat hasil jadi secara keseluruhan pada gambar 4. 1 dibawah ini:

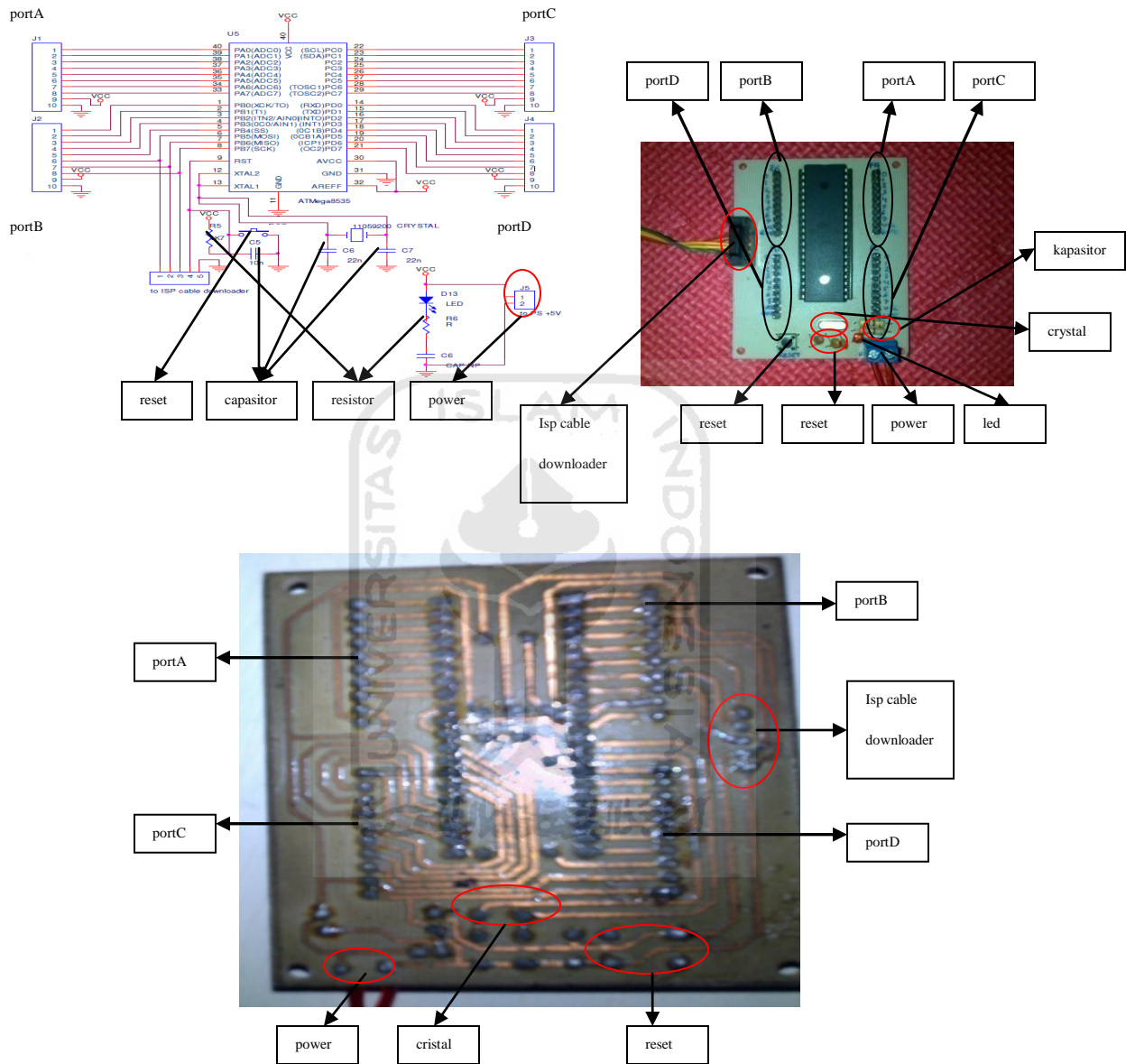


Gambar 4. 1 Hasil Aplikasi Mikrokontroler

Rangkaian minimum ATmega8535 yang telah terangkai dan terdiri dari 4 port yaitu portA, portB, portC, portD. *Hardware output* dan rangkaian mikrokontroler akan dihubungkan dengan kabel pelangi (lihat gambar 4.2) sesuai dengan isi dari listing program.



Gambar 4. 2 Kabel Pelangi



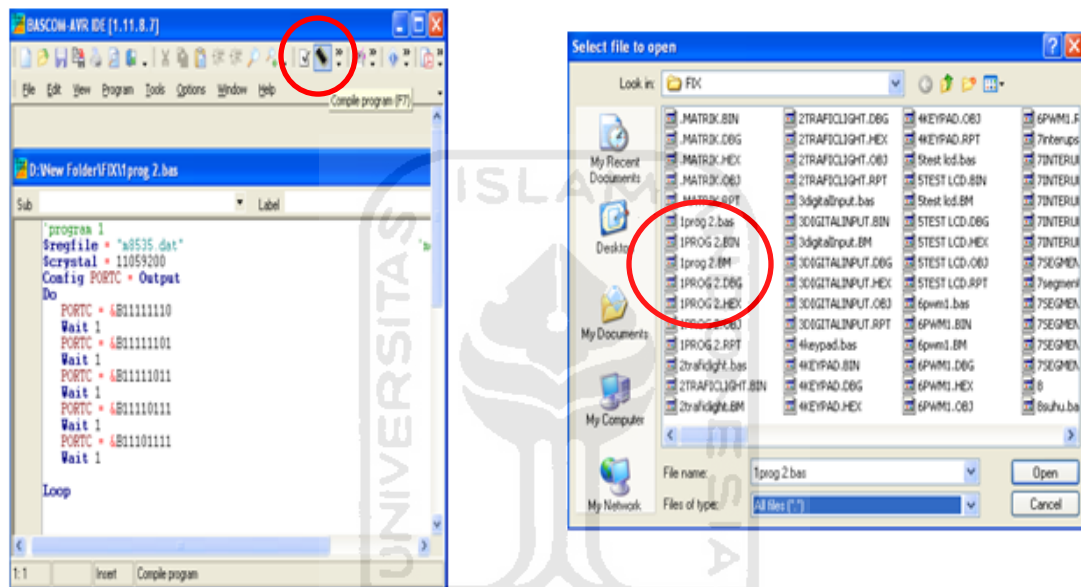
Gambar 4.3 Schematic dan Rangkaian Minimum ATmega8535

LED pada gambar 4.3 berfungsi sebagai tanda *on /off*. Ketika power dan kabel *downloader* diaktifkan LED akan *on* dan ketika di *non*-aktifkan akan *off*.

4.1.1 Cara meng-Compile dan men-Download Program

a. Meng - Compile

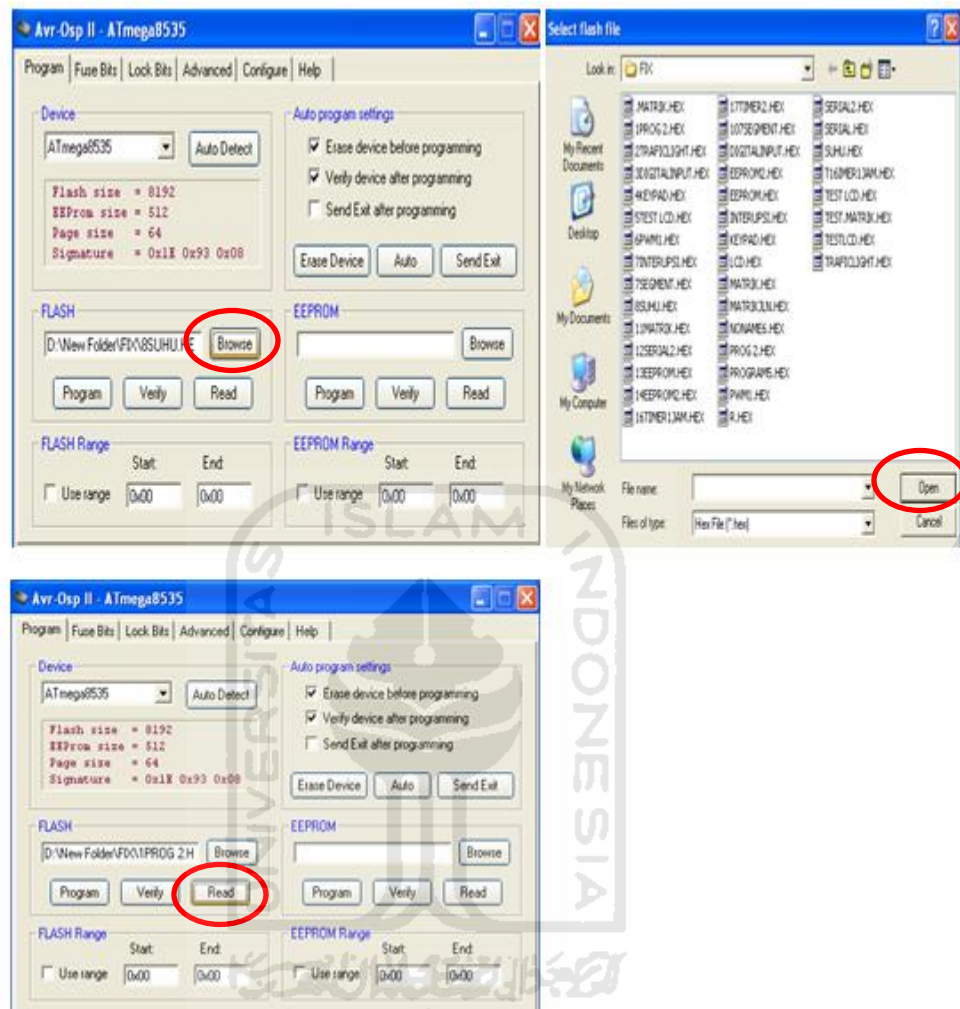
Meng-Compile dilakukan pada *software* BASCOM AVR. Setelah program selesai dibuat klik **compile program (F7)** dan hasil *compile* terdiri dari .Hex dan .Bas. Selengkapnya dapat kita lihat pada gambar 4.2.



Gambar 4.4 Screenshot Compile

b. Men - Download

Men-download program dilakukan pada *software* AVR OSP II. Buka AVR OSP II klik **browse** dan akan muncul window baru kemudian pilih program target dan klik **ok**. Setelah program target selesai dipilih kemudian klik **read** dan program pun akan ter-download. Selengkapnya dapat kita lihat pada gambar 4.4.



Gambar 4.5 Screenshot Download

4.1.2 LED Berjalan Berurutan

Pada gambar 4.3 di bawah akan dilihat LED akan menyala secara berurutan dari kanan ke kiri. LED di bawah di contohkan hanya berjalan pada satu port mikrokontroler, misalnya di *PORTC* saja.

PortC pada mikrokontroler (lihat gambar 4.2) dihubungkan dengan j1 pada rangkaian LED (lihat gambar 3.2) menggunakan kabel pelangi. (*Script* program lihat lampiran hal.1)

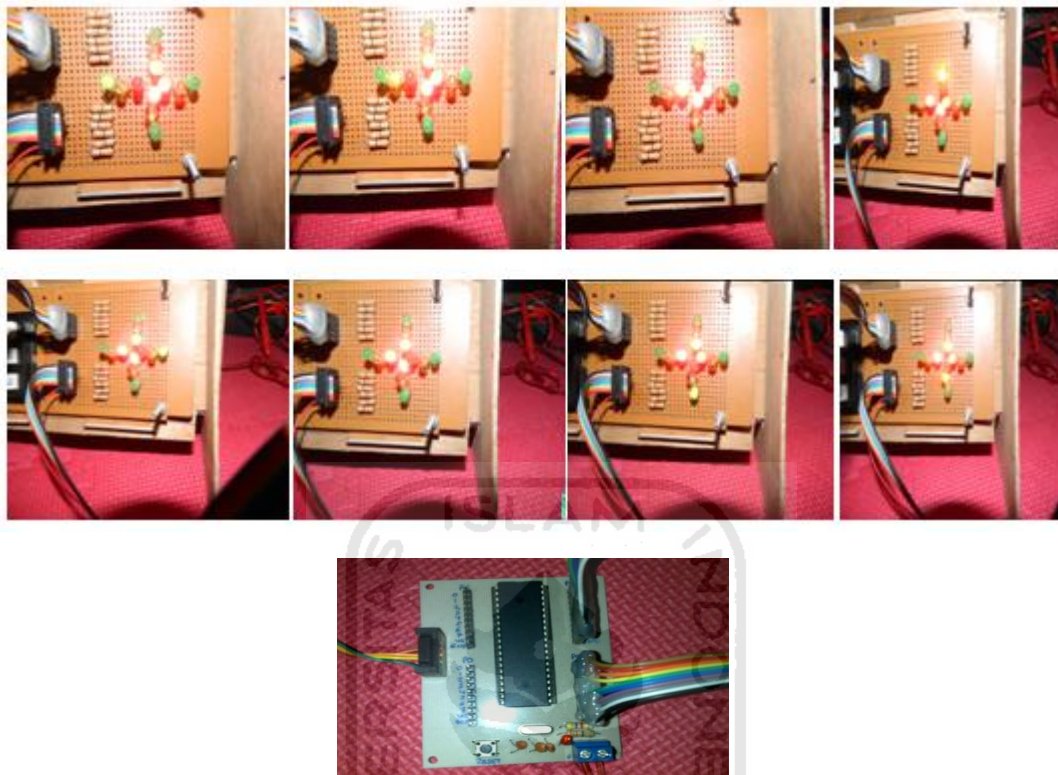


Gambar 4. 6 LED Berjalan Berurutan

4.1.3 Trafficlight

Pada gambar 4. 3 di bawah LED di simulasikan seperti *trafficlight*. Dimulai dari LED *trafficlight* sebelah kiri, atas, kanan, kemudian ke bawah, dan dilakukan secara berulang. Berbeda dengan LED berjalan berurutan diatas, LED *trafficlight* ini menggunakan dua *port* pada mikrokontroler , dikarenakan dalam satu *port* mikrokontroler hanya terdiri dari 8 pin I/O sedangkan *trafficlight* membutuhkan 12 pin I/O.

Pada *trafficlight* port yang digunakan adalah portA dihubungkan dengan j1 dan portC dihubungkan dengan j2 (lihat gambar 4.2 dan gambar 3.3). (*Script* program lihat lampiran hal.2)

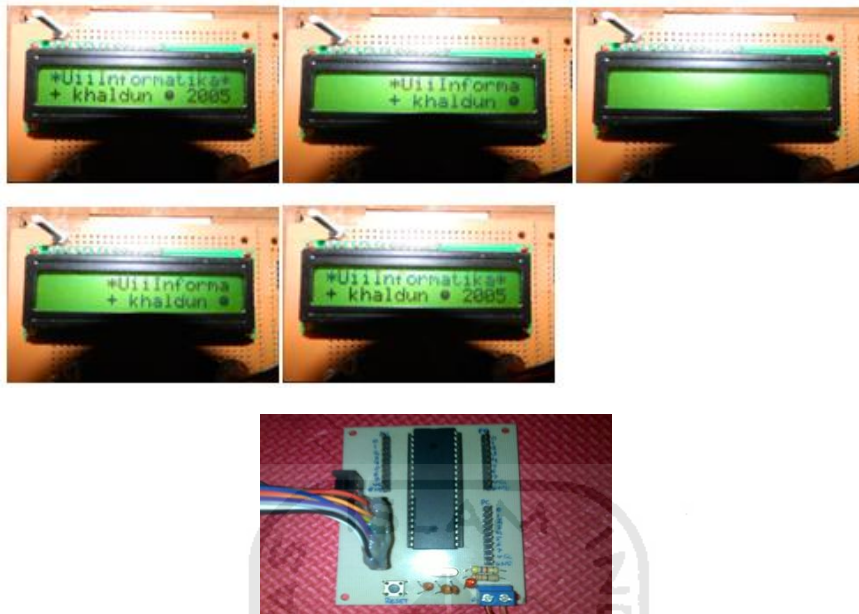


Gambar 4.7 Trafficlight

4.1.4 LCD

Pada gambar 4.4 dibawah kita akan melihat tampilan LCD yang bergeser kekanan terlebih dahulu dan akan bergeser kekiri sehingga tampilannya akan sama seperti di awal. Pada tampilan LCD di bawah dapat terlihat gambar *icon* yang tidak ada pada shortcut pada keyboard. Kita dapat membuatnya dengan cara buka menu *tools* pada BASCOM kemudian klik *LCD designer*.

Pada LCD port yang digunakan adalah portD (lihat gambar 4.2) yang dihubungkan dengan j1 pada rangkaian LCD (lihat gambar 3.4). (*Script* program lihat lampiran hal.4)



Gambar 4. 8 LCD

4.1.5 Dotmatrik

Gambar 4.6 dotmatrik dibawah dapat dilihat baris pertama akan *on* dan dilanjutkan ke baris berikutnya sampai ke baris tujuh, kemudian kolom pertama *on* dilanjutkan ke kolom berikutnya sampai kolom lima. Gambar *script* di bawah menunjukkan pergeseran baik baris dan kolom membutuhkan waktu 500ms, sehingga LED terlihat seperti berjalan dari atas ke bawah, dan dari samping kiri ke samping kanan.

Pada *dotmatrik* menggunakan dua port pada rangkaian mikrokontroler yaitu, portA dihubungkan pada j1 dan portC dihubungkan dengan j2 (lihat gambar 4.2 dan gambar 3.6). (*Script* program lihat lampiran hal.7)



Gambar 4. 9 Dotmatrik

4.1.6 Seven segment

Gambar 4.7 *seven segment* dibawah terlihat ada 2 digit angka 8. Yang mana waktu *on* antara digit membutuhkan 5 ms. Sehingga antara digit 1 dengan digit 2 terlihat menyala secara bersamaan. *Seven segment* banyak digunakan untuk membuat jam *digital* dan pula dapat digunakan sebagai penghitung waktu.

Pada *seven segment* juga menggunakan dua *port* sama halnya dengan *dotmatrik* yaitu, portA dihubungkan dengan j1 dan portB dihubungkan dengan j2 (lihat gambar 4.2 dan gambar 3.5). (*Script* program lihat lampiran hal.10)



Gambar 4. 10 Seven Segment

4.1.7 Switch dan LED

Prinsip dasar dari *digital input* adalah membaca data yang masuk dari mikrokontroler. Tentu PORTx harus dibuat sebagai *port* input, dapat dengan cara:

Config PORTx = input, atau

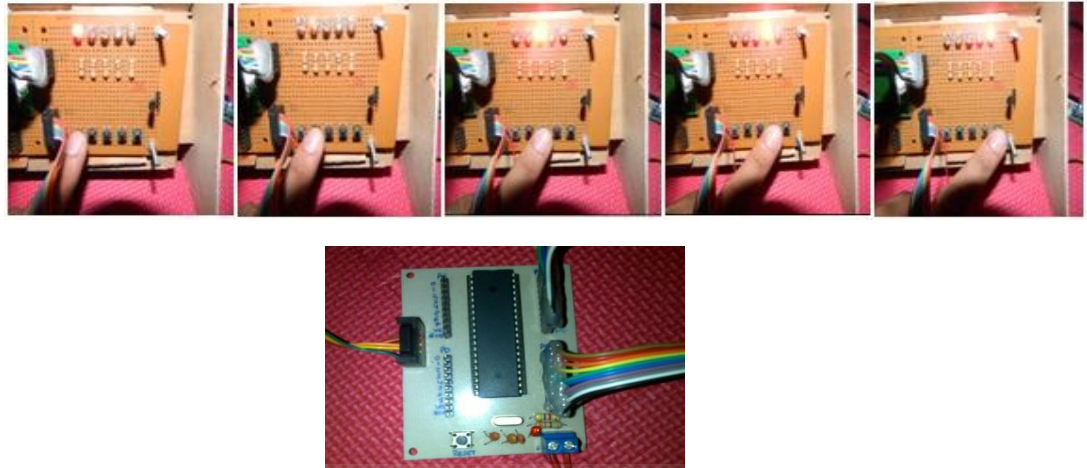
DDRx = 0

Gambar 4.8 menunjukkan ketika *switch* 1 di tekan maka LED 1 on, *switch* 2 di tekan LED 2 on, dan seterusnya berurutan sampai *switch* 5 dan LED 5. Dapat kita lihat pada tabel di bawah.

Table 4.1 Switch dan LED

No	switch	LED on
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

Pada *switch* dan LED menggunakan portA (lihat gambar 4.2) dihubungkan dengan j1 pada LED (lihat gambar 3.2), dan portC (lihat gambar 4.2) dihubungkan dengan j1 pada *switch* (lihat gambar 3.8). (*Script* program lihat lampiran hal.11)



Gambar 4. 11 Switch dan LED

4.1.8 Keypad dan LED

Keypad dan LED ini sama halnya dengan *Switch* dan LED termasuk pada *digital input* maka PORTx harus di set sebagai *input*.

Pada gambar 4.9 dapat dilihat ada 12 tombol pada keypad dan 5 buah LED, ketika salah satu keypad di tekan maka LED akan on. Lebih jelasnya dapat kita pada Table 3.2 di bawah ini:

Tabel 4.2 Keypad dan LED

No	Keypad	LED on
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	1, 5
7	7	1, 2
8	8	1, 2, 3
9	9	4, 5

10	*	1, 3, 5
11	0	1, 2, 3, 4, 5
12	#	-

Pada *keypad* dan LED menggunakan portA (lihat gambar 4.2) dihubungkan dengan j1 pada LED (lihat gambar 3.2), dan portC (lihat gambar 4.2) dihubungkan dengan j1 pada *keypad* (lihat gambar 3.9). (*Script* program lihat lampiran hal.13)



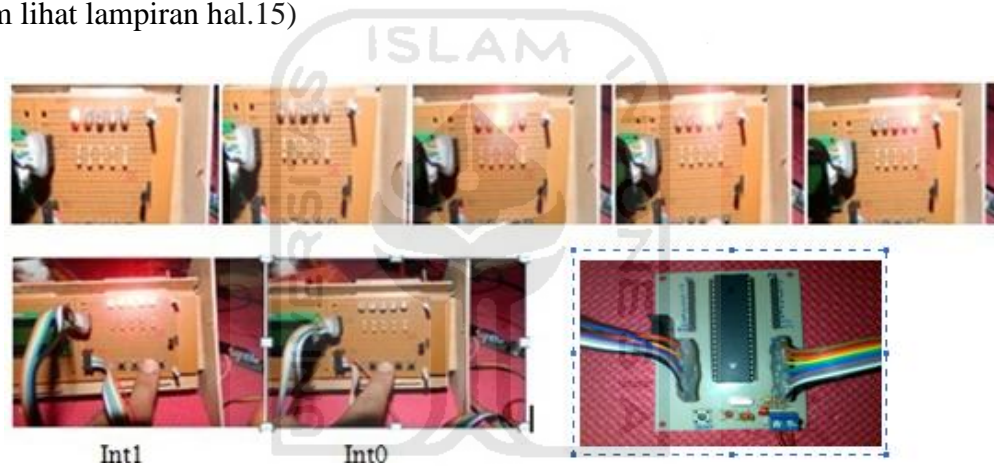
Gambar 4. 12 Keypad dan LED

4.1.9 Interupsi Eksternal

Interupsi merupakan sebuah kondisi dimana CPU berhenti dari program yang sedang dijalankan untuk melayani permintaan suatu rutin yang terjadi.

Gambar 4. 5 dibawah menunjukkan LED akan berjalan berurutan dari LED 1 sampai LED 5 secara terus menerus sampai ada interupsi pada switch 2 dan switch 3. Ketika switch 2 di tekan terjadi interupsi 0 dan LED akan *off*, ketika switch 3 di tekan terjadi interupsi 1 dan semua LED akan *on*. Ketika interupsi selesai, LED akan berjalan seperti awal mula ketika interupsi belum terjadi.

Pada *interupsi eksternal* menggunakan portC (lihat gambar 4.2) yang dihubungkan dengan j1 pada rangkaian LED (lihat gambar 3.2) dan portD (lihat gambar 4.2) dihubungkan dengan j1 pada rangkaian *switch* (lihat gambar 3.8). (*Script* program lihat lampiran hal.15)



Gambar 4. 13 Interupsi Eksternal

4.1.10 Timer Counter

a. Timer

Timer adalah hitungan jumlah sinyal *input* pada *interval* konstan untuk menunjukkan waktu yang telah berlalu. Pada gambar 4.11 dapat kita lihat seperti simulasi pada jam *digital*. Detik akan berjalan dari 1 sampai 60. Ketika detik sudah mencapai 60, detik akan kembali ke 1 dan menit akan bertambah 1 dan seterusnya.

Pada *timer* menggunakan portD (lihat gambar 4.2) yang dihubungkan dengan j1 pada LCD (lihat gambar 3.4). (*Script* program lihat lampiran hal.17)

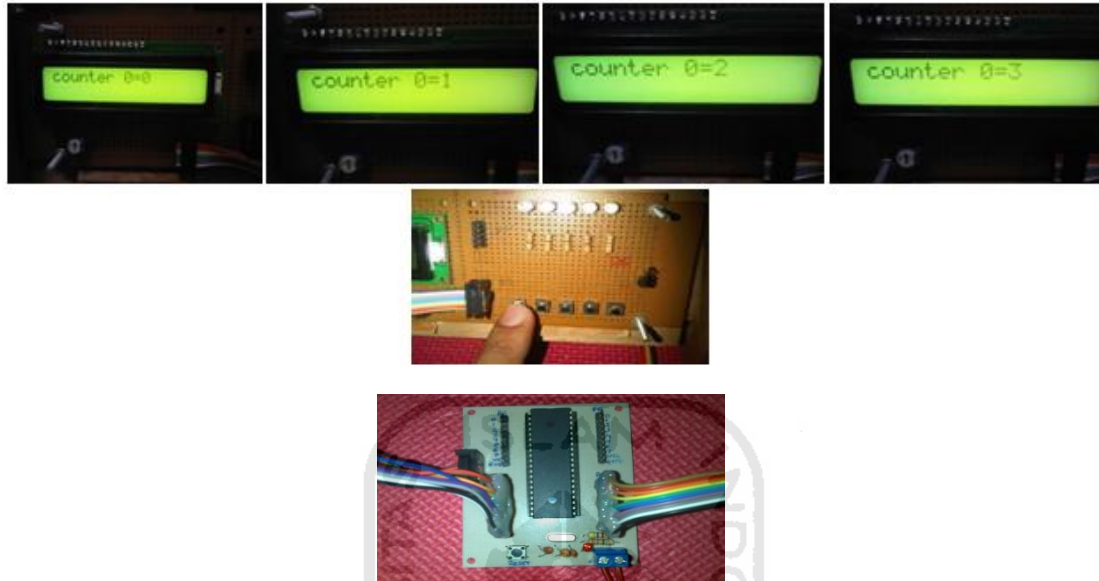


Gambar 4. 14 Timer

b. Counter

Counter adalah hitungan dan menunjukkan jumlah sinyal (peristiwa) masukan pada setiap *interval*. Pada gambar dibawah nilai *counter* akan bertambah satu ketika *switch* ditekan. Inilah yang akan dapat membedakan antara *timer* dan *counter*. Jika *counter* membutuhkan *event* dalam menjalankannya.

Pada *counter* menggunakan portB (lihat gambar 4.2) yang dihubungkan dengan rangkaian switch dan portD dihubungkan pada LCD (lihat gambar 3.4). (*Script* program lihat lampiran hal.19)

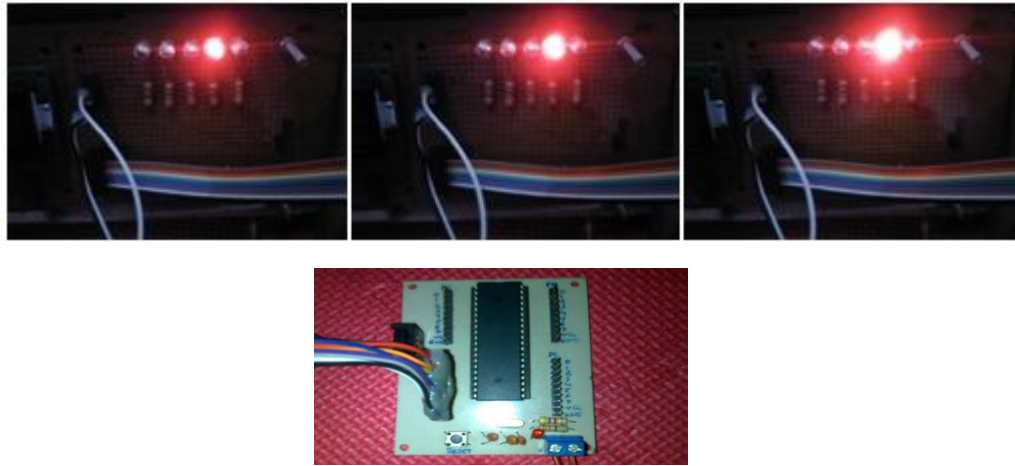


Gambar 4. 15 Counter

4.1.11 PWM

Keunggulan ATmega8535 salah satunya adalah fasilitas PWM. Fasilitas ini dapat digunakan untuk mengontrol motor DC, motor servo, atau mengatur terang redup LED. Pwm mempunyai sinyal frekuensi yang tetap, tetapi lebar pulsa 'high' dan lebar pulsa 'low' dapat diatur. (Inkubatek,2008)

Pada gambar 4.13 akan terlihat LED seolah-olah terlihat redup kemudian akan semakin terang dan kembali redup kemudian terang lagi dan seterusnya. Pada *PWM* menggunakan portD pin.6 (lihat gambar 4.2) yang dihubungkan dengan j1 pin.2, dan Vcc dihubungkan pada j1 pin.9 pada rangkaian LED, kabel yang digunakan hanya 2 helai saja, tidak perlu menggunakan kabel pelangi yang utuh. (*Script* program lihat lampiran hal.21)



Gambar 4. 16 PWM

4.1.12 Serial komunikasi

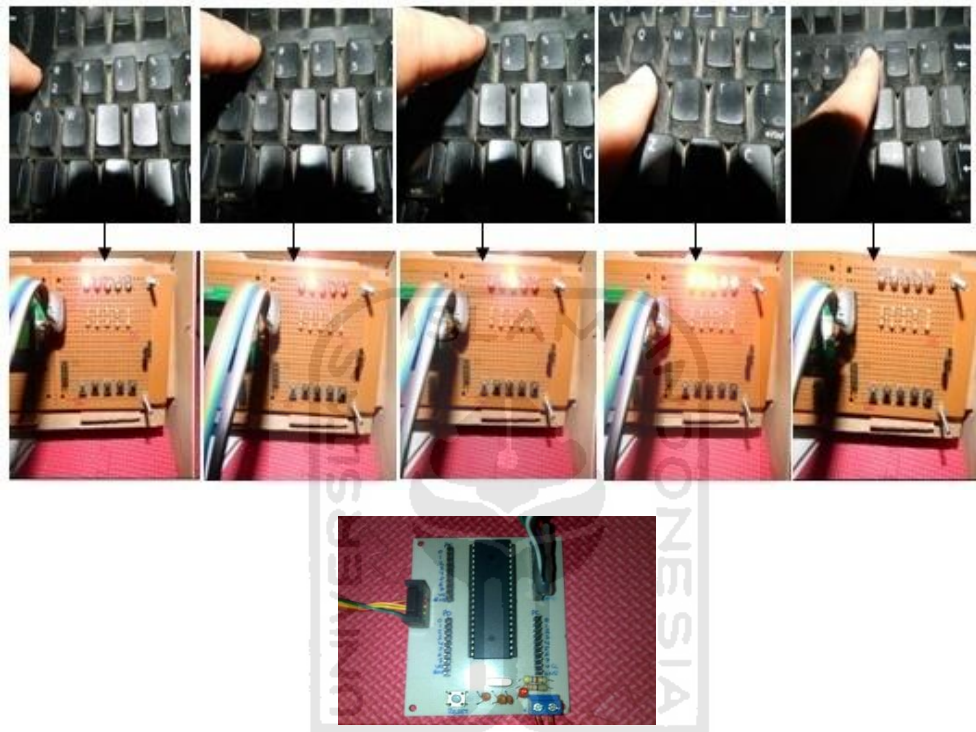
Cara menghubungkan serial komunikasi mikrokontroler melalui hyperterminal cukup mudah. *Klik start pada PC -> allprogram -> accessories -> communication -> hyperterminal -> isikan name koneksinya->tekan ok ->isikan baudrate 9600 ->isikan hardware none kemudian ok.* Hyperterminal siap digunakan.

Pada gambar 4. 6 dapat kita lihat ketika keyboard ditekan maka LED akan *on*, lebih jelasnya dapat kita lihat pada table 3.2 di bawah :

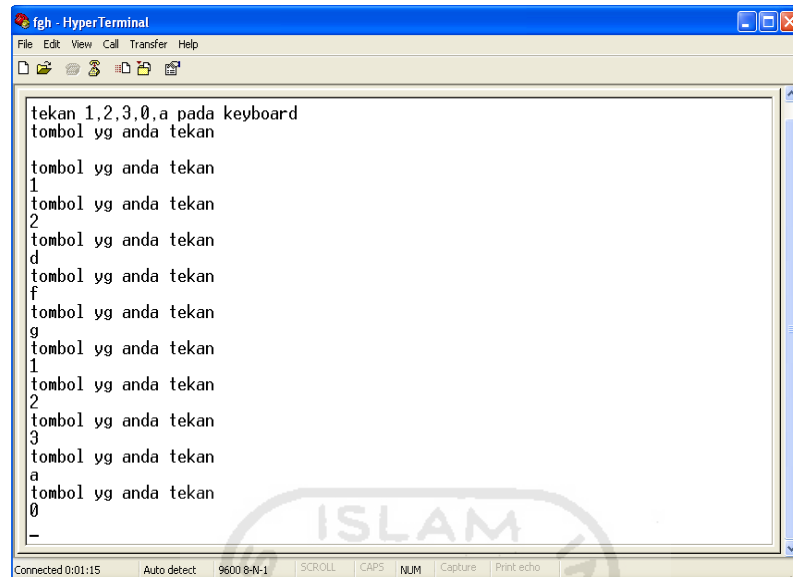
Tabel 4.3 Serial Komuknikasi

No	Shortcut keyboard	LED on
1	1	1
2	2	2
3	3	3
4	4	1,2,3,4,5
5	5	-

Pada serial komunikasi port yang digunakan hanya 1 *port* karena *port* input menggunakan keyboard pada PC, yaitu portA (lihat gambar 4.2) yang dihubungkan j1 pada rangkaian LED (lihat gambar 3.2). (*Script* program lihat lampiran hal.22)



Gambar 4. 17 Serial Komunikasi



Gambar 4. 18 Hyperterminal

Gambar 4. 7 adalah hasil *screenshot* dari hyperterminal dari PC ketika keyboard ditekan.

4.1.13 ADC dengan sensor suhu LM35

Fitur ADC pada ATmega8535 dapat membaca sinyal analog kemudian dikonversikan menjadi sinyal *DIGITAL*. Ada 2 mode konversi pada ADC *single conversion* dan *free running*. *Single conversion* -> dalam setiap konversi di perlukan satu kali intruksi, *free running* -> ADC cukup diberi perintah sekali saja, maka secara otomatis akan bekerja terus sampai ada perintah yang menghentikannya.

Gambar 4.16 merupakan tampilan sensor suhu yang di tampilkan pada LCD. Suhu pada LCD dapat berubah-ubah tergantung ruangan dimana sensor tersebut diaktifkan.

Pada ADC ini port yang digunakan adalah portA (lihat gambar 4.2). V.in pada pada lm35 dihubungkan pada portA pin.1, dan vcc, GND pada lm35 dihubungkan pada portA pada pin.9 dan pin.10 (lihat gambar 3.7). (*Script* program lihat lampiran hal.22)



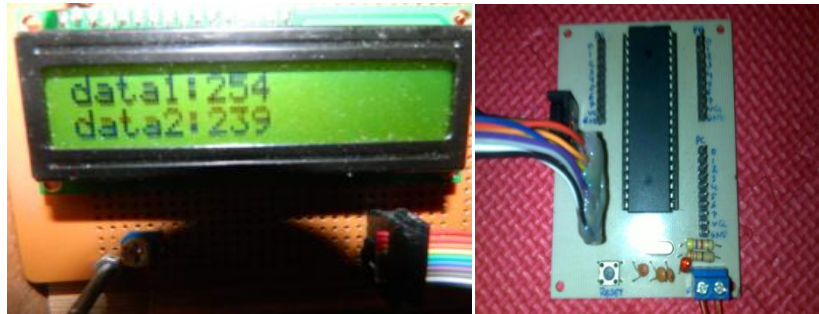
Gambar 4. 19 ADC dengan Sensor Suhu LM35

4.1.14 EEPROM

Fungsi dari EEPROM untuk menyimpan data secara permanen, artinya jika kita menyimpan data dalam SRAM, maka ketika power dimatikan data akan hilang. Beda ceritanya jika disimpan dalam EEPROM, data akan tetap tersimpan.

Gambar 4.17 menunjukkan LCD menampilkan data yang sudah di simpan pada EEPROM.

Pada EEPROM disini akan ditampilkan pada LCD sehingga port yang digunakan adalah portD (lihat gambar 4.2) yang dihubungkan pada j1 pada LCD (lihat gambar 3.4). (*Script* program lihat lampiran hal.24)



Gambar 4. 20 EEPROM

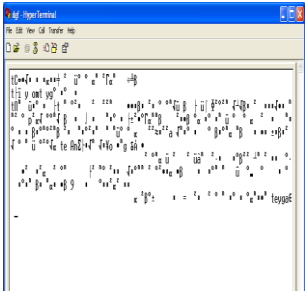
4.2 Troubleshooting

Dalam *troubleshooting* ini akan membahas tentang *error-error* yang terjadi selama dalam pengerjaan aplikasi mikrokontroler ini. Baik dari *software*, maupun *hardware*.

Tabel 4.4 Troubleshooting

No	Error yang terjadi	Penyelesaian
1	LED Mati	<ul style="list-style-type: none"> • Periksa anoda dan katoda (-/+) pada LED menggunakan <i>multimeter</i>, bisa saja LED rusak / mati. • Periksa kabel pelangi menggunakan <i>mulimeter</i>, di mungkinkan kabel ada yang putus/ rusak. • Periksa <i>listing program</i>, di mungkinkan port yang digunakan belum diberi logika 0 (<i>low</i>).

2	LED Mati Pada Seven Segment dan Dotmatrik	Sama dengan halnya pada LED mati diatas, tetapi pada <i>sevent segment</i> dan <i>dotmatrik</i> dapat juga terjadi kerusakan pada transistor dan dapat pula di periksa dengan <i>multimeter</i> .
3	Kabel Pelangi	Pada saat menghubungkan pin antara mikrokontroler dengan <i>hardware output</i> , harus sama tipe pinnya. Misalkan pin.0 pada mikrokontroler harus bertemu pin.0 pada pin output <i>hardware</i> nya, VCC dengan VCC, GND dengan GND. Jika tidak terkoneksi dengan benar maka output pada <i>hardware</i> pun akan tidak sesuai dengan <i>script</i> programnya.
4	Crystal	Dalam aplikasi mikrokontroler ini menggunakan <i>crystal eksternal</i> 11MHz. Jadi, pada saat downloader terkoneksi dengan <i>software downloader</i> harus dilakukan <i>setting fuse bits</i> terlebih dahulu pada <i>software downloader</i> . Dengan cara <i>fuse bits -> read</i> .
5	Downloader k-125R pada AVR OSP II 1) Error opening COM port 3	1) Jika ini yang terjadi berarti K-125 tidak terdeteksi oleh komputer walau pada <i>device manager</i> sudah

	<p>2) Checking programmer tipe... Time during COM-port read operation!..... Supported programmer not found on Com Port 3</p> <p>3) Checking programmer type... Found AVR ISP..... Entering programming mode... Signature = 0xFF 0xFF 0xFF... Leaving programming mode...</p>	<p>terdeteksi sebagai <i>Com. Port.</i> yang perlu diperhatikan dalam hal ini adalah penginstalan <i>driver</i> yang belum tepat.</p> <p>2) Jika hal ini yang terjadi coba perhatikan switch biru pada K-125 pada keadaan difungsikan sebagai <i>Downloader switch</i> biru harus tergeser pada posisi Program tidak pada posisi <i>serial</i>.</p> <p>3) Jika hal ini yang terjadi maka cobalah lihat koneksi MOSI, MISO, SCK, dan RESET pada koneksi K-125 dengan <i>board</i>. Pastikan kabel terkoneksi dengan benar, ceklah menggunakan <i>multimeter</i>.</p>
6	<p>Komunikasi serial</p> <p>Jika Hyperterminal error seperti gambar di bawah.</p> 	<p>Gambar disamping ini menunjukkan <i>error</i> yang terjadi ketika komunikasi <i>serial</i> dijalankan pada hyperterminal. Solusinya kita bisa cek pin RX dan pin TX antara mikrokontroler dengan downloader k-125R apakah terbalik atau tidak. Jika pin sudah benar, maka tekan tombol <i>reset</i> pada</p>

		mikrokontroler.
7	LCD LCD bergeser ke kiri tetapi tidak sebanyak 16 baris. Dan hasil pada LCD pun akan tidak beraturan.	Terjadi <i>human error</i> seperti <i>script</i> di bawah: <pre>For A = 1 To -16 Shiftlcd Right</pre> Script yang benar: <pre>For A = 1 To 16 Shiftlcd Right</pre> Script diatas di maksudkan LCD akan bergeser kekanan sebanyak 16 baris.

4.3 Analisis Kelebihan dan Kekurangan Sistem

Pada bagian ini akan dibahas mengenai keunggulan serta kelemahan dari sistem aplikasi yang telah dibuat.

4.3.1 Kelebihan

Kelebihan dari aplikasi mikrokontroler ATmega8535 yang dibuat adalah :

1. Mudah di pelajari bagi para pemula yang ingin belajar mikrokontroler dengan bahasa *basic*.
2. PCB yang digunakan adalah PCB *matriks* atau lebih dikenal PCB 'berlubang'. PCB ini cocok digunakan untuk pemula karena tidak perlu menggambar *layout* pada PCB.
3. Menggunakan *downloader serial k-125R*, sehingga aplikasi dapat dijalankan pada laptop dan PC.
4. Bentuknya yang minimalis sehingga mudah di bawa dan di bongkar pasang.

4.3.2 Kekurangan

Kekurangan dari aplikasi mikrokontroler ATmega8535 yang dibuat adalah :

1. Pada PCB *matriks* sulit mengatur system pengkabelan yang menghubungkan antara komponen satu dengan komponen lainnya sehingga menyebabkan kabel-kabel yang dihubungkan saling menyilang. Kesulitan lainnya juga pada penyolderan kaki-kaki komponen dengan 2 kabel penghubung atau lebih, pada titik solder yang sama.
2. Fitur PWM hanya disimulasikan pada LED. Seharusnya lebih bagus disimulasikan dengan motor servo.
3. Butuh lebih banyak perangkat keras agar dalam pembuatan *script program* dapat ber-*variatif*.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melalui berbagai tahapan pembuatan program dan hardware, maka penulis dapat mengambil beberapa kesimpulan sebagai berikut:

1. Untuk mempelajari AVR ATMEGA8535 tidaklah cukup dengan membaca teori dan datasheetnya, tetapi dengan mempraktekannya. Namun, untuk melaksanakan hal itu kita harus mempunyai alat pendukung piranti elektronika.
2. *Script* bahasa pemrograman yang sudah benar, belum tentu benar hasil outputnya pada hardware. Maka dari itu harus dilakukan pengecekan terus-menerus.

5.2 Saran

Setelah melihat hasil yang dicapai dalam Tugas Akhir ini, maka ada beberapa saran yang perlu disampaikan sebagai berikut:

1. Pilih kualitas kabel yang bagus dan sesuai standar, karena kabel yang kurang bagus lebih mudah patah dan aliran datanya kurang cepat.
2. Perhatikan pesan-pesan error yang terdapat pada software, baik pada BASCOM-AVR maupun AVR OSPII.
3. Aplikasi mikrokontroler ini masih besar kemungkinannya dapat dikembangkan lebih *variatif* lagi tergantung user yang mengembangkannya.
4. User dapat menambahkan hardware pendukung lainnya yang lebih *variatif* pula.
5. Kreatifitas sangat dibutuhkan pada pemrograman yang mempunyai output hardware.

DAFTAR PUSTAKA

- Atmel Corporation, 2006, *ATMega8535 Data Sheet*, <http://atmel.com>.
- Anonym, 2006, Data Sheet, <http://datasheetarchive.com>.
- Anonym, <http://id.wikipedia.org>. (Akses Terakhir 25/07/ 2011)
- Agfiantio, & Dhani nugraha, 2010, *Tutorial Pemograman AVR dengan WinAVR GCC (ATMega 16/32/8535)*, <http://www.scribd.com/doc/49577248/5/Diagram-Pin-dan-Diagram-Blok-Mikrokontroler-AVR-ATMega16>.
- Chandra, 2010, *Rangkaian sensor suhu LM35*, <http://telinks.wordpress.com/2010/04/09/rangkaian-sensor-suhu-lm35/>.
- Elektro-control team, 2011, *timer dan counter AVR*, <http://electrocontrol.wordpress.com/2011/04/22/timer-dan-counter-avr/>.
- Zygan, *all about controler*, <http://www.avrku.com/>.
- Fahmizal, <http://fahmizaleeits.wordpress.com/>.
- Iswanto, 2008. *Design dan implementaasi sistem embedded mikrokontroler ATMega8535 dengan bahasa basic*, Yogyakarta : penerbit gava media.
- Inkubator team, 2008. *Pemrograman mikrokontroler AVR ATMega8535 dengan BASCOM AVR*, Yogyakarta : Inkubator teknologi.
- Wardana, & Lingga, 2006, *belajar sendiri Mikrokontroler AVR Seri ATMega8535*, Yogyakarta: penerbit andi.
- Creative Vision, 2010, *Buku Manual K-125R.pdf*, Jakarta : penerbit CreativeVision.

LAMPIRAN

1. program LED Berjalan Berurutan

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Portc = Output
```

```
Do
```

```
Portc = &B11111110
```

```
Wait 1
```

```
Portc = &B11111101
```

```
Wait 1
```

```
Portc = &B11111011
```

```
Wait 1
```

```
Portc = &B11110111
```

```
Wait 1
```

```
Portc = &B11101111
```

```
Wait 1
```

```
Loop
```



Ketika portc.0 *on* dibutuhkan 1 detik kemudian portc.1 akan *on*. Ketika portc.1 *on* dibutuhkan 1 detik kemudian portc.2 *on* demikian seterusnya sampai portc.5 dan akan diulang kembali ke portc.0.

2. program Traffilight

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Ddrc = &B1111111
```

```
Ddra = &B1111111
```

```
Do
```

```
Portc = &B11011110
```

```
Porta = &B11110011
```

```
Wait 3
```

```
Portc = &B11011101
```

```
Porta = &B11110011
```

```
Wait 1
```

```
Portc = &B11011011
```

```
Porta = &B11110110
```

```
Wait 3
```

```
Portc = &B11011011
```



Porta = &B11110101

Wait 1

Portc = &B10011011

Porta = &B11111011

Wait 3

Portc = &B01011011

Porta = &B11111011

Wait 1

Portc = &B11110011

Porta = &B11110011

Wait 3

Portc = &B11101011

Porta = &B11110011

Wait 1

Loop



No port	PORTA	PORTC
0	Hijau 2	Hijau 1
1	Kuning 2	Kuning 1
2	Merah 2	Merah 1
3	Merah 3	Hijau 4

4	-	Kuning 4
5	-	Merah 4
6	-	Hijau 3
7	-	Kuning 3
8	vcc	-
9	-	-

Portc.0, portc.5, porta.2, dan port.3 *on* (lampu dalam keadaan hijau,merah,merah,merah), dibutuhkan 3 detik agar portc.1, portc.5, porta. 2 dan porta.3 *on* (lampu dalam keadaan kuning, merah, merah, merah). Kemudian dibutuhkan 1 detik ke port selanjutnya agar keadaan lampu menjadi hijau,merah ,merah, merah.

3. program LCD

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Lcdpin = Pin , Rs = Portd.0 , E = Portd.2 , Db4 = Portd.4
```

```
Config Lcdpin = Pin , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7
```

```
Config Lcd = 16 * 2
```

```
Deflcdchar 0 , 32 , 2 , 20 , 14 , 5 , 8 , 32 , 32
```

```
Deflcdchar 1 , 32 , 8 , 5 , 14 , 20 , 2 , 32 , 32
```

```
Deflcdchar 2 , 32 , 4 , 4 , 31 , 4 , 4 , 32 , 32
```

```
Deflcdchar 3 , 32 , 14 , 27 , 28 , 31 , 14 , 32 , 32
```

Cursor Off

Do

Cls

Dim A As Integer

Lcd "*UiiInformatika*"

Wait 3

Locate 2 , 3

Lcd "khalidun 2005"

Waitms 300

Do

Locate 2 , 1

Lcd Chr(0) ;

Waitms 500

Locate 2 , 1

Lcd Chr(1) ;

Waitms 500

Locate 2 , 1

Lcd Chr(2) ;

Waitms 500



```
Locate 2 , 1  
Lcd Chr(0) ;  
Waitms 500  
Locate 2 , 1  
Lcd Chr(1) ;  
Waitms 500  
Locate 2 , 1  
Lcd Chr(2) ;  
Waitms 500  
Locate 2 , 11  
Lcd Chr(3);  
Waitms 500  
Locate 2 , 11  
Lcd Chr(4);  
Waitms 500  
For A = 1 To 16  
Shiftlcd Right  
Waitms 300  
Next
```



```

        Wait 1

        For A = 1 To 16

            Shiftlcd Left

            Waitms 300

            Next

        Loop

Loop
End

```

Script program locate akan menentukan kolom,baris. Dimisalkan locate 2,1 maka posisi pada lampilan LCD dimulai dari kolom 2 dan baris 1. Lcd chr() pada *script* diatas digunakan agar icon terlihat dapat bergerak. *Shiflcd right* diartikan LCD akan bergeser ke kanan, dah *shiflcd left* diartikan LCD akan bergeser ke kiri.

4. program Dotmatrik

```

$regfile = "m8535.dat"

$crystal = 11059200

Config Portc = Output

Config Porta = Output

Do

    Portc = &B11100000

```

Porta = &B11111110

Waitms 500

Portc = &B11100000

Porta = &B11111101

Waitms 500

Portc = &B11100000

Porta = &B11111011

Waitms 500

Portc = &B11100000

Porta = &B11110111

Waitms 500

Portc = &B11100000

Porta = &B11101111

Waitms 500

Portc = &B11100000

Porta = &B11011111

Waitms 500

Portc = &B11100000

Porta = &B10111111



Waitms 500

Portc = &HFE

Porta = &H80

Waitms 500

Portc = &HFD

Porta = &H80

Waitms 500

Portc = &HFB

Porta = &H80

Waitms 500

Portc = &HF7

Porta = &H80

Waitms 500

Portc = &HEF

Porta = &H80

Waitms 500

Loop



PORTC sebagai kolom, dan PORTA sebagai baris. PORTC diaktifkan dan porta.0 diaktifkan maka LED pada baris 1 akan menyala kemudian butuh waktu 500ms untuk porta.1 diaktifkan dan baris 2 akan menyala demikian seterusnya.

5. program Seven segment

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Portb = Output
```

```
Config Porta = Output
```

```
Dim I As Byte
```

```
Do
```

```
  For I = 0 To 1
```

```
    Porta = Lookup(i , Digit)
```

```
    Portb = Lookup(i , Angka)
```

```
    Waitms 5
```

```
  Next
```

```
Loop
```

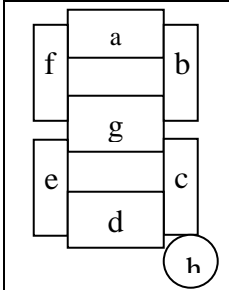
```
Digit:
```

```
  Data &HFE , &HFD ,
```

```
Angka:
```



Data &H80 , &H80



angka	h	g	f	e	d	c	b	a	heksa
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	0	82
7	1	1	1	1	1	0	0	0	F8
8	1	0	0	0	0	0	0	0	80
9	1	0	0	1	0	0	0	0	90

Tabel diatas dapat dilihat bagaimana cara membuat angka pada *seven segment*. *Seven segment* ini pun dapat dibuat bentuk beberapa huruf, tergantung kebutuhan user.

6. program Switch dan LED

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200  
  
Config Portc = Input  
  
Portc = &HFF  
  
Config Porta = Output  
  
Porta = &HFF  
  
Do  
  
  If Pinc.0 = 0 Then  
    Porta = &B11111110  
  
  ElseIf Pinc.1 = 0 Then  
    Porta = &B11111101  
  
  ElseIf Pinc.2 = 0 Then  
    Porta = &B11111011  
  
  ElseIf Pinc.3 = 0 Then  
    Porta = &B11110111  
  
  ElseIf Pinc.4 = 0 Then  
    Porta = &B11101111  
  
End If  
  
Loop
```



Ketika pinc.o (switch) ditekan maka porta.o *on* (LED *on*), jika tidak pinc.1 ditekan maka porta.1 *on*. Program diatas akan terus berulang sampai PORTC mendapatkan logika *low*(0) dan PORTA sebagai output LED akan *on*.

7. program keypad dan LED

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Ddrc = &B1110000
```

```
Portc = &HFF
```

```
Ddra = &HFF
```

```
Porta = &HFF
```

```
Do
```

```
    Portc = &B1101111
```

```
    If Pinc.0 = 0 Then
```

```
        Porta = &B11111110
```

```
    ElseIf Pinc.1 = 0 Then
```

```
        Porta = &B11110111
```

```
    ElseIf Pinc.2 = 0 Then
```

```
        Porta = &B11111100
```

```
    ElseIf Pinc.3 = 0 Then
```



```
Porta = &B11101010  
  
End If  
  
Portc = &B1011111  
  
If Pinc.0 = 0 Then  
  
    Porta = &B11111101  
  
Elseif Pinc.1 = 0 Then  
  
    Porta = &B11101111  
  
Elseif Pinc.2 = 0 Then  
  
    Porta = &B11111000  
  
Elseif Pinc.3 = 0 Then  
  
    Porta = &B00000000  
  
End If  
  
Portc = &B0111111  
  
If Pinc.0 = 0 Then  
  
    Porta = &B11111011  
  
Elseif Pinc.1 = 0 Then  
  
    Porta = &B11101110  
  
Elseif Pinc.2 = 0 Then  
  
    Porta = &B11100111
```



```
Elseif Pinc.3 = 0 Then
```

```
    Porta = &B11111111
```

```
End If
```

```
Loop
```

Ketika pinc.o (keypad) ditekan maka porta.o *on* (LED 1 *on*), jika tidak, pinc.1 ditekan maka porta.3 *on* (LED 4 *on*). Program diatas akan terus berulang sampai PORTC mendapatkan sinyal *low*(0), dan PORTA sebagai output pada LED akan *on*.

8. program Interupsi Eksternal

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Int0 = Low Level
```

```
Config Int1 = Low Level
```

```
On Int0 Int_ext0
```

```
On Int1 Int_ext1
```

```
Enable Interrupts
```

```
Enable Int0
```

```
Enable Int1
```

```
Ddrc = &HFF
```

```
Ddrd = 0
```



```
Portd = &HFF
```

```
Portc = 0
```

```
Do
```

```
    Portc = &B11111110
```

```
    Wait 1
```

```
    Portc = &B11111101
```

```
    Wait 1
```

```
    Portc = &B11111011
```

```
    Wait 1
```

```
    Portc = &B11110111
```

```
    Wait 1
```

```
    Portc = &B11101111
```

```
    Wait 1
```

```
Loop
```

```
Int_ext0:
```

```
    Portc = &HFF
```

```
Return
```

```
Int_ext1:
```

```
    Portc = &HE0
```



Return

LED akan berjalan berurutan dari 1 sampai 5. Dimisalkan ketika LED 3 *on* dan terjadi int0 maka semua LED akan *off*. Ketika int0 selesai, LED 4 akan *on* dan kembali berjalan berurutan seperti semula.

9. program Timer Counter

Pada ATmega8535 terdapat tiga timer/counter yaitu timer/counter 0, 1, dan 2.

1. timer/counter 0

Mempunyai kapasitas cacahan sampai 8 bit (1byte). Dapat difungsikan sebagai timer, counter, dan PWM.

2. timer/cunter 1

Sama halnya dengan timer/counter0, hanya saja kapasitas cacahannya lebih besar yaitu 16 bit (2byte).

3. timer/couter 2

Kapasitas cacahnya 8 bit, tetapi tidak dapat difungsikan sebagai counter.

a. Timer

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Dim Detik As Byte , Menit As Byte
```

```
Config Lcdpin = Pin , Rs = Portd.0 , E = Portd.2 , Db4 = Portd.4
```

Config Lcdpin = Pin , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7

Config Lcd = 16 * 2

Cursor Off

Cls

Config Timer1 = Timer , Prescale = 256

Do

Tcnt1h = &H57

Tcnt1l = &H3f

Start Timer1

Do

Loop Until Tifr.2 = 1

Stop Timer1

 Tifr.2 = 1

Incr Detik

If Detik = 60 *Then*

 Menit = Menit + 1

 Detik = 0

Elseif Menit = 60 *Then*

 Menit = 0



End If

Locate 1 , 1

Lcd Detik ; "s ";

Locate 2 , 1

Lcd Menit;

Loop

Mengkonfigurasi pin LCD dan timer, menentukan nilai TCNTH dan TCNTL, aktifkan register TIFR.2, incr = +1. Detik akan bertambah +1, jika sudah mencapai 60 maka menit bertambah +1 dan detik kembali 0, dan jika menit sudah mencapai 60 maka menit akan kembali 0. Detik akan di tampilkan pada LCD pada baris 1, dan menit pada baris 2.

Cara mencari TCNT1H dan TCNTL :

Prescale = 256, crystal= 11059200, timer1 16 bit = 65535. Rumus crystal/prescale kemudian kapasitas cacah yang 16 bit dikurangi hasil dari crystal dan prescale.

$$11059200/256 = 43200 ,$$

$$65535 - 43200 = 22335 \rightarrow \text{heksa} = 573F$$

57 di pakai pada *high*, 3F dipakai pada *low*.

b. counter

\$regfile = "m8535.dat"

\$crystal = 11059200

```
Config Lcdpin = Pin , Rs = Portd.0 , E = Portd.2 , Db4 = Portd.4
```

```
Config Lcdpin = Pin , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7
```

```
Config Lcd = 16 * 2
```

```
Cursor On Off
```

```
Cls
```

```
Tcnt0 = 0
```

```
Config Timer0 = Counter , Edge = Falling
```

```
Ddrb.0 = 0
```

```
Portb.0 = 1
```

```
Start Timer0
```

```
Lcd "counter 0="
```

```
Do
```

```
    Locate 1 , 11
```

```
    Lcd Tcnt0 ; " "
```

```
Loop
```

Pertama-tama konfigurasi pin LCD TCNT0 di set pada posisi nol, konfigurasi timer pada posisi counter, *edge* pada posisi *falling* (ketika ditekan). Aktifkan register PORTB.0. PORTB.0 di beri logika *high*, timer diaktifkan dan ditampilkan pada LCD. TCNT0 akan bertambah jika PORTB.0 diberi logika *low* (ditekan).



10. program PWM

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up , Prescale = 1
```

```
Do
```

```
Pwm1a = 20
```

```
Wait 1
```

```
Pwm1a = 40
```

```
Wait 1
```

```
Pwm1a = 60
```

```
Wait 1
```

```
Pwm1a = 255
```

```
Wait 1
```

```
Loop
```



Fitur PWM terdapat pada pin OC1A yaitu pada PORTD.5. Konfigurasi PWM, pwm di pilih 8 bit, nilai TOP pada angka 255. Sinyal PWM di mulai pada 20 tunggu 1 detik sinyal PWM berubah menjadi 40 tunggu 1 detik sinyal berubah menjadi 60 dan tunggu 1 detik sinyal menjadi 255 dan akan berulang terus- menerus.

11. program Serial komunikasi

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
$baud = 9600
```

```
Dim Tombol As Byte
```

```
Config Porta = Output
```

```
Porta = &HFF
```

```
Print "tekan 1,2,3,0,a pada keyboard"
```

```
Do
```

```
Tombol = Waitkey()
```

```
Print "tombol yg anda tekan"
```

```
Print Chr(tombol)
```

```
Select Case Tombol
```

```
Case "1" : Porta = &B11111110
```

```
Case "2" : Porta = &B11111101
```

```
Case "3" : Porta = &B11111011
```

```
Case "a" : Porta = &B11100000
```

```
Case "0" : Porta = &B11111111
```

```
End Select
```



Loop

Pada serial komunikasi ini hasil akhirnya hampir sama dengan keypad dan LED, perbedaannya disini menggunakan keyboard pada PC. Ketika tombol 1 di tekan PORTA.1 *on* ketika tombol 2 ditekan PORTA.2 *on*, ketika tombol 3 ditekan PORTA.3 *on*, ketika tombol a ditekan PORTA.1.2.3.4.5 *on*, ketika tombol 0 ditekan PORTA *off*.

12. program ADC dengan sensor suhu LM35

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Lcdpin = Pin , Rs = Portd.0 , E = Portd.2 , Db4 = Portd.4
```

```
Config Lcdpin = Pin , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7
```

```
Config Lcd = 16 * 2
```

```
Cursor Off
```

```
Deflcdchar 1 , 12 , 18 , 18 , 12 , 32 , 32 , 32 , 32
```

```
Cls
```

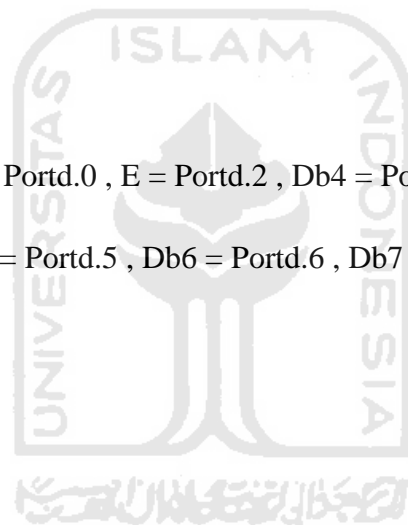
```
Config Adc = Single , Prescaler = Auto , Reference = Avcc
```

```
Dim W As Word , V As Word , T As Single , Suhu As Word
```

```
Lcd "ADC:"
```

```
Locate 2 , 1
```

```
Lcd "teg:mV"
```



Locate 1 , 8

Lcd "suhu:"

Start Adc

Do

W = Getadc(0)

V = W * 5

T = V / 11.6

Suhu = T

Locate 1 , 5

Lcd W

Locate 2 , 5

Lcd V ; " mv "

Locate 1 , 13

Lcd Suhu ; Chr(1) ; "C"

Wait 1

Loop

Mengkonfigurasi pinLCD dan ADC, kemudian menentukan *variable*. ADC diaktifkan. Rumus suhu $T = \text{teg} / \text{skala}$. Jadi untuk menentukan skala harus cari suhu terlebih dahulu, dengan cara ambil thermometer lihat suhunya maka skala akan di dapatkan. Jika suhu yang didapat tidak angka bulat maka masukan *variable suhu* as



word. Pada baris pertama pada LCD akan menampilkan tegangan ADC yang didapat dan suhu, pada baris 2 LCD tegangan dalam mv.

13. progam EEPROM

```
$regfile = "m8535.dat"
```

```
$crystal = 11059200
```

```
Config Lcdpin = Pin , Rs = Portd.0 , E = Portd.2 , Db4 = Portd.4
```

```
Config Lcdpin = Pin , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7
```

```
Config Lcd = 16 * 2
```

```
Cursor Off
```

```
Cls
```

```
Dim Data_eeprom As Byte
```

```
Data_eeprom = &HFE
```

```
Writeeeprom Data_eeprom , 1
```

```
Data_eeprom = &HEF
```

```
Writeeeprom Data_eeprom , 3
```

```
Readeeprom Data_eeprom , 1
```

```
Lcd "data1:" ; Data_eeprom
```

```
Readeeprom Data_eeprom , 3
```

```
Locate 2 , 1
```

```
Lcd "data2:" ; Data_eeprom
```



End

Konfigurasi pin LCD kemudian tentukan nama *variable* EEPROM, kemudian masukan data yang akan disimpan. Setelah itu data akan ditampilkan pada LCD.

