

***PROGRAMMABLE TIMER* UNTUK PERCOBAAN TUMBUKAN**

Laporan tugas akhir ini diajukan sebagai salah satu syarat memperoleh gelar Ahli

Madya

Pada Program Studi Diploma 3 Instrumentasi Industri

Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia



PUTRADHIPA KUSUMA

2305221222

PROGRAM D3 INSTRUMENTASI ELEKTRONIKA DAN INDUSTRI

DEPARTEMEN FISIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

DEPOK

2008

LEMBAR PENGESAHAN

Nama : Putradhipa Kusuma
NPM : 2305221222
Jurusan : D3 Fisika Instrumentasi Industri
Tanggal Sidang : 11 Juli 2008
Judul Tugas Akhir : *Programmable Timer* untuk Percobaan Tumbukan

Laporan Tugas Akhir Penelitian ini telah diperiksa dan disahkan oleh :

Pembimbing

(Arief Sudarmaji, M.T)

Penguji 1

Penguji 2

Penguji 3

(Djati Handoko, M.Si)

(Adhi Harmoko, M.Kom)

(Dr. Supriyanto)

ABSTRAK

Telah dibuat suatu Programmable Timer untuk percobaan tumbukan. Programmable timer yang dibuat terdiri dari 3 mode. Mode pertama dengan menggunakan 2 buah sensor, mode kedua dan ketiga dengan menggunakan 1 buah sensor. Programmable timer dapat menampilkan waktu dalam 4 buah seven segment dan 100 buah LED. Waktu maksimal yang dapat ditampilkan yaitu 10000 detik, dengan nilai waktu yang paling kecil mencapai 10 ms. Nilai waktu yang diperoleh dari programmable timer jika dibandingkan dengan pewaktu seperti stopwatch maka akan didapat nilai yang mendekati sama, untuk waktu 100 detik perbedaan yang didapat rata-rata hanya 0,058 detik, untuk waktu 500 detik diperoleh perbedaan dengan rata-rata sekitar 6,089 detik, untuk waktu 1000 detik perbedaan yang didapat rata-rata sebesar 8,068 detik, dan untuk waktu 9999 detik perbedaan yang didapat rata-rata sebesar 16,694 detik.

Kata Kunci : mode, sensor, LED, seven segment, timer.

KATA PENGANTAR

Puji syukur penulis ucapkan atas Kehadirat Allah SWT, karena Rahmat dan Ridho-Nya penulis mampu menyelesaikan penelitian ini beserta laporan akhir tepat pada waktunya. Serta dengan karunia-Nya penulis juga mendapat begitu besar limpahan ilmu dan semangat untuk terus berusaha dan berdoa dalam menjalani proses penelitian yang panjang. Sehingga Nikmat yang begitu besar atas penyelesaian tugas akhir ini penulis dapatkan.

Akhir dari kerja keras penulis, diwujudkan dalam penyusunan dan penulisan laporan tugas akhir penelitian dengan judul **Programmable Timer untuk Percobaan Tumbukan**. Laporan tugas akhir ini menjadi sebuah pengembangan analisa dan pengkajian secara teknis terhadap objek penelitian.

Sebagai bentuk ungkapan terima kasih, penulis memberikan sanjungan serta penghargaan kepada seluruh pihak atas andilnya terhadap penyusunan laporan tugas akhir ini. Pujian penulis secara tulus diberikan dan diucapkan kepada:

1. Allah SWT, Tuhan Yang Majha Esa, yang hingga detik ini senantiasa melimpahkan kasih sayang, rahmat dan nikmat-Nya
2. Bapak dan Ibu tercinta untuk doa dan semangatnya tidak pernah berkurang, dan untuk adik-adikku tersayang yang selalu menghilangkan beban dan penat. Juga untuk semua saudara dan kerabat.
3. Dr. Prawito, selaku Ketua Jurusan Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia, yang dedikasi pengabdian dan jasanya terus tercurah dalam membimbing dan mendidik penulis.
4. Bapak Surya Dharma, M.Si selaku koordinator Tugas Akhir Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.
5. Arief Sudarmaji, M.T, selaku dosen dan pembimbing penulis pada Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia..

Terima kasih atas bimbingan dan bantuannya selama proses penelitian sehingga penelitian berjalan baik dan menyenangkan.

6. Seluruh dosen dan staff pengajar Program Diploma 3 Instrumentasi Elektronika dan Industri Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.
7. Pak Katman dan Pak Parno yang selalu membantu dan memudahkan dalam peminjaman alat selama penelitian di Laboratorium Elektronika dan Bengkel Mekanik.
8. Krisna, partner setia yang selalu berbagi dan bekerja keras dalam menghadapi segala rintangan selama penyelesaian penelitian dengan tawa dan senyum. Terima kasih juga untuk segala bantuan dan semangatnya.
9. Franky, thank to kostannya. Yang telah diperbolehkan untuk diacak-acak menjadi kapal pecah. Dan maaf bayarnya ngaret trus.
10. E'ce, tempat ku mengenyangkan perutku dengan duit 4000 perak.
11. Rekan-rekan seperjuangan ku Instrument05, Bam's, Indra JB, Mr. Jo, Bos Hendry, Argi, and smuanya yang telah membuat diriku terasa berarti.
12. Dan seluruh pihak yang tidak bisa saya sebutkan satu persatu

Penulis menyadari bahwa laporan ini banyak kekurangannya dan penulis terbuka untuk menerima kritik dan saran yang bersifat membangun untuk penyusunan laporan agar tercipta peningkatan mutu dan kualitas penulisan laporan tugas akhir yang lebih baik.

Diharapkan Laporan Tugas Akhir ini bermanfaat bagi pembaca untuk menjadi panduan di kalangan para akademisi dan seluruh civitas akademika Universitas Indonesia, dan bagi penulis pada khususnya.

Universitas Indonesia

Depok, Juli 2008

Penulis

DAFTAR ISI

Halaman Judul	
Lembar Pengesahan	ii
Abstrak.....	iii
Kata Pengantar	iv
Daftar Isi	vi
Daftar Gambar	viii
Daftar Tabel	ix
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Tujuan Penelitian	2
1.2 Deskripsi Singkat.....	2
1.4 Pembatasan Masalah	3
1.5 Metodologi Penelitian	4
1.6 Sistematika Penulisan	5
BAB 2 TEORI DASAR	6
2.1 Microcontroller	6
2.1.1 Perkembangan Microcontroller.....	7
2.1.2 Sekilas tentang Microcontroller AVR.....	9
2.1.3 Perlengkapan Dasar Microcontroller	11
2.2 Minimum Sistem	12
2.2.1 Konfigurasi Pin AVR ATmega16	12
2.2.2 Struktur Memori ATmega16	15
2.3 Sensor Infrared	18
2.4 Seven Segment	20
2.5 LED	21
2.6 Timer	21

2.7 Interrupt	23
2.8 PWM	25
2.9 Bascom AVR	26
BAB 3 PERANCANGAN DAN PEMBUATAN PROGRAM	28
3.1 Konfigurasi Sistem	28
3.2 Sistem Pewaktu	29
3.3 Display LED dan Seven Segment	31
3.4 I/O Hardware	33
3.5 Mode Timer	34
3.5.1 Mode 1	35
3.5.2 Mode 2	36
3.5.3 Mode 3	37
BAB 4 PENGUJIAN DAN ANALISA	39
4.1 Pengujian dan Analisa Timer	39
4.2 Pengujian dan Analisa Data Mode	42
BAB 5 PENUTUP	46
5.1 Kesimpulan	46
5.2 Saran	46
Daftar Pustaka	48

DAFTAR GAMBAR

Gambar 1.1. Benda Jatuh Bebas	2
Gambar 1.2. Blok Diagram Pada Sistem	3
Gambar 2.1. Pin ATmega 16	13
Gambar 2.2. Peta Memori Flash	15
Gambar 2.3. Peta Memori SRAM.....	16
Gambar 2.4. Register alamat EEPROM Bit 15...8	17
Gambar 2.5. Register data EEPROM Bit Bit 7...0 ⁶	17
Gambar 2.6. <i>Register kontrol EEPROM Bit Bit 7...0</i>	18
Gambar 2.7. Pin IC 555	19
Gambar 2.8. Seven Segment.....	20
Gambar 2.9. Rangkaian Dasar Seven Segment	20
Gambar 2.10. Skema Interrupt	23
Gambar 2.11. Sinyal PWM dengan <i>duty cycle</i> 50%.....	25
Gambar 2.12. Sinyal PWM dengan <i>duty cycle</i> 10%.....	26
Gambar 3.1. Sistem Pewaktu	30
Gambar 3.2. Display LED Seven Segment.....	32
Gambar 3.3. Push button dan solenoid	33
Gambar 3.4. Flowchart Utama.....	35
Gambar 3.5. Flowchart Mode 1	36
Gambar 3.6. Flowchart Mode 2	37
Gambar 3.7. Flowchart Mode 3	38

DAFTAR TABEL

Tabel 1.1. Benda Jatuh Bebas	2
Tabel 1.2. Blok Diagram Pada Sistem	3
Tabel 2.1. Perbedaan seri AVR berdasarkan jumlah memori.....	10
Tabel 2.2. Interrupt vektor	24
Tabel 4.1. Pengujian untuk 100 detik	39
Tabel 4.2. Pengujian untuk 500 detik	40
Tabel 4.3. Pengujian untuk 1000 detik	41
Tabel 4.4. Pengujian untuk 9999 detik	41
Tabel 4.5. Pengujian mode 1	43
Tabel 4.6. Pengujian mode 2.....	43
Tabel 4.7. Pengujian mode 3.....	44

BAB 1

PENDAHULUAN

Pada Bab I akan dibahas mengenai Pendahuluan yaitu tentang latar belakang dibuatnya Tugas Akhir ini. Selain itu penulis juga akan membahas tujuan penulisan, pembatasan masalah, metode penelitian yang disertai skema penelitian, dan sistematika penulisan yaitu digunakan dalam penulisan ini dengan membagi menjadi beberapa bab dan masing-masing terdiri dari sub bab.

1.1 Latar Belakang

Dalam Perkembangan ilmu fisika modern, kemajuan teknologi telah membuat banyak kemudahan disegala bidang. Banyak teori – teori fisika yang telah diterapkan didalam segala aspek kehidupan, salah satunya adalah teori mengenai tumbukan yang mengacu pada hukum kekekalan momentum.

Dalam pengertiannya tumbukan adalah interaksi antara sebuah benda terhadap benda yang lain sehingga ada perubahan kecepatan dan perubahan momentum benda setelah terjadi tumbukan.

Setiap benda yang mengalami tumbukan dengan benda lain yang diam ataupun bergerak akan mengalami perubahan kecepatan pada benda tersebut. Kecepatan tersebut dipengaruhi oleh waktu tempuh dan juga jarak tempuh, serta massa dari benda tersebut. Kecepatan juga dipengaruhi oleh gesekan antara permukaan benda dengan permukaan lantai. Maka jika diketahui waktu tempuh dari benda yang bertumbukan maka akan bisa dicari nilai kecepatannya, dengan begitu juga dapat dicari nilai gesekannya.

Sehingga sangat diperlukan pencatat waktu yang sangat akurat dalam melakukan percobaan tumbukan. Sehingga akan didapatkan nilai kecepatan dan juga nilai gesekan yang mempunyai tingkat error yang sangat kecil. Oleh karena itu penulis membuat Tugas Akhir dengan judul "Programable Timer untuk Percobaan Tumbukan".

1.2 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah agar dalam melakukan percobaan tumbukan didapatkan waktu yang lebih akurat dan juga mempermudah dalam percobaan. Dengan selesainya proyek akhir ini diharapkan pewaktu terprogram ini dapat membantu dalam memahami sebuah aplikasi fisika dan dapat bermanfaat bagi dunia pendidikan.

1.3 Deskripsi Singkat

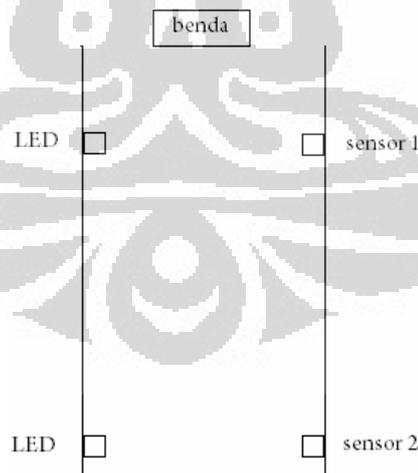
Pewaktu terprogram yang digunakan pada percobaan tumbukan ini diibaratkan seperti waktu yang dibutuhkan sebuah benda yang jatuh bebas(lihat Gambar 1.2). Pewaktu untuk percobaan tumbukan ini ada dua tipe, yaitu :

1. Dengan menggunakan 1 sensor

Pada tipe pertama ini, waktu mulai dihitung saat benda mulai jatuh, waktu akan berhenti saat benda melewati sensor 2.

2. Dengan menggunakan 2 sensor

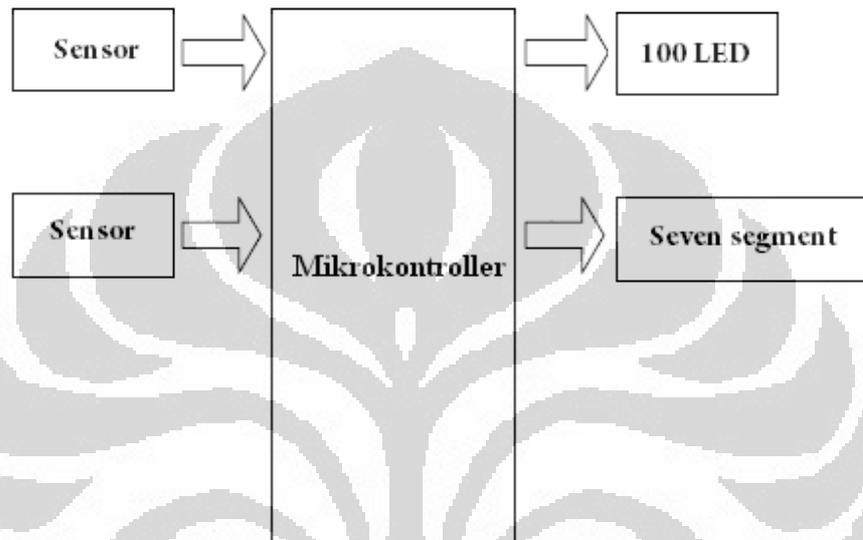
Untuk tipe yang kedua ini, waktu akan mulai dihitung jika benda telah melalui sensor 1, dan waktu akan berhenti jika benda melalui sensor 2.



Gambar 1.1. Benda Jatuh Bebas

Waktu yang diperlukan oleh benda yang jatuh bebas tersebut akan diwakilkan oleh 100 buah LED, dimana setiap 1 buah LED yang menyala akan bernilai 10

milisecond(ms). LED akan menyala secara berputar satu arah, yang banyak putarannya sesuai dengan waktu jatuh benda. Jika LED sudah berputar sebanyak satu putaran(samadengan 100 LED) maka seven segment akan memunculkan angka 1 yang berarti 1 detik untuk setiap satu putaran penuh. Ketika waktu jatuh benda sudah selesai maka total waktu yang terukur akan ditampilkan lewat seven segment.



Gambar 1.2. Blok Diagram Pada Sistem

1.4 Batasan Masalah

Dalam proses perancangan dan pengerjaannya Proyek Tugas Akhir ini dibagi dalam dua buah bagian, yaitu :

- Bagian pertama adalah mengenai perancangan dan pembuatan rancang bangun dari sistem mekanik pada percobaan tumbukan, dimana pada bagian ini akan difokuskan dalam pembuatan mekanik dari proyek tugas akhir tersebut. Mulai dari rancangan alat, pencarian bahan, hingga proses pembuatan dan penyelesaiannya.
- Bagian kedua adalah mengenai pewaktu terprogram untuk percobaan tumbukan, dimana dibagian kedua ini tentang program pewaktunya.

Dalam proposal tugas akhir ini penulis hanya akan membahas bagian yang kedua, yaitu pembuatan program untuk pewaktu terprogram yang nantinya akan digunakan pada percobaan tumbukan. Pada pewaktu terprogram ini terdapat dua tipe pilihan, yaitu dengan 1 sensor, dan dengan 2 sensor.

1.5 Metode

1. Study Literatur

Penulis menggunakan metode ini untuk memperoleh informasi dan data yang berkaitan dengan penelitian yang penulis lakukan. Study literatur ini mengacu pada buku-buku pegangan, data sheet dari berbagai macam komponen yang dipergunakan, data yang didapat dari internet, dan makalah-makalah.

2. Perancangan dan Pembuatan Sistem

Berisi tentang proses perencanaan sistem baik hardware maupun software. Pada bagian hardware akan membahas desain dan cara kerjanya. Pada bagian software akan dibahas program yang digunakan.

3. Pembuatan Program

Pembuatan program dilakukan dengan menggunakan *Software Basic Compiler (BASCUM)*, dengan menggunakan Software ini memungkinkan kita untuk memanipulasi kinerja alat sesuai dengan yang diinginkan.

4. Uji Sistem

Dari sistem yang dibuat maka dilakukan pengujian secara menyeluruh, dengan tujuan untuk mengetahui kinerjanya apakah sudah sesuai dengan apa yang diharapkan atau belum.

5. Pengambilan Data

Pada bab ini akan diuraikan tentang kinerja dari masing – masing blok dengan harapan dalam pengujian tidak terdapat kesalahan yang fatal.

6. Penulisan Penelitian

Dari hasil pengujian dan pengambilan data kemudian dilakukan suatu analisa sehingga dapat diambil suatu kesimpulan. Dengan adanya beberapa saran juga dapat kita ajukan sebagai bahan perbaikan untuk penelitian lebih lanjut

1.6 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari beberapa bab. Agar lebih mempermudah pemahaman dan pembacaan, maka laporan tugas akhir ini disusun menjadi seperti di bawah ini :

BAB 1. PENDAHULUAN

Pada bab ini berisi tentang latar belakang, tujuan penelitian pembatasan masalah, metode penelitian dan sistematika penulisan.

BAB 2. TEORI DASAR

Pada Bab ini berisi tentang konsep dasar dari microcontroller ATmega16 dan semua sistem yang mendukung kinerja dari proyek tugas akhir ini .

BAB 3. PERANCANGAN DAN PEMBUATAN PROGRAM

Merupakan penjelasan pembuatan rancangan sistem program untuk programmable timer, baik dalam membuat timer maupun mode-modenya.

BAB 4. PENGUJIAN DAN ANALISA

Program yang telah dibuat kemudian diuji dengan parameter-parameter yang terkait. Pengujian ini meliputi pengujian ketepatan waktu yang dihasilkan oleh timer dan juga aktifnya mode-mode yang ada. Di samping pengujian, proses pengambilan data kerja sistem ini juga dituliskan di bab ini untuk memastikan kemampuan sistem secara keseluruhan. Dari hasil ini dapat dilakukan analisa terhadap kerja sistem, sehingga dapat diketahui apa yang menjadi penyebab dari kendala atau kegagalan bila selama kegiatan penelitian ditemui hal-hal tersebut.

BAB 5. PENUTUP

Pada Bab ini berisi tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan yang berkaitan dengan sistem kerja program dari proses pengujian timer yang dilakukan pada proyek tugas akhir ini.

BAB 2

TEORI DASAR

Pada perancangan Programmable Timer diperlukan beberapa pemahaman dasar. Beberapa pemahaman dasar tersebut antara lain: pin ATmega 16, PWM (*Pulse Width Modulation*), Sensor Infrared, Seven Segment, Interrupt, Timer, dan pemrograman Bascom.

2.1. Microcontroller

Microcontroller sebagai sebuah “*one chip solution*” pada dasarnya adalah rangkaian terintegrasi (*Integrated Circuit-IC*) yang telah mengandung secara lengkap berbagai komponen pembentuk sebuah komputer. Berbeda dengan penggunaan *microprocessor* yang masih memerlukan komponen luar tambahan seperti RAM, ROM, Timer, dan sebagainya--untuk sistem *microcontroller*. Hal ini disebabkan semua komponen penting tersebut telah ditanam bersama dengan sistem prosesor ke dalam IC tunggal *microcontroller* bersangkutan. Dengan alasan itu sistem *microcontroller* dikenal juga dengan istilah populer *the real Computer On a Chip*-komputer utuh dalam keping tunggal, sedangkan sistem *microprocessor* dikenal dengan istilah yang lebih terbatas yaitu *Computer On a Chip*-komputer dalam keping tunggal.

Berdasarkan fungsinya, *microcontroller* secara umum digunakan untuk menjalankan program yang bersifat permanen pada sebuah aplikasi yang spesifik (misal aplikasi yang berkaitan dengan pengontrolan dan monitoring). Sedangkan program aplikasi yang dijalankan pada sistem *microprosesor* biasanya bersifat sementara dan berorientasi pada pengolahan data. Perbedaan fungsi kedua sistem diatas secara praktis mengakibatkan kebutuhan minimal yang harus dipenuhi juga akan berbeda (misal ditinjau dari kecepatan detak operasi, jumlah RAM, panjang register, dan lain sebagainya). Hampir tidak dapat disangkal, dewasa ini akan sukar

dijumpai seseorang yang masih menggunakan komputer dengan microprocessor berbasis 8 atau 16 bit (misal microprocessor 8088 dan 8086 produk perusahaan Intel). Mengapa demikian?, salah satu alasannya— Perangkat lunak komputer yang beredar saat ini umumnya mensyaratkan kecepatan CPU yang sangat tinggi (dalam orde Mega bahkan GigaHz) serta memori dengan kapasitas sangat besar (dalam orde MegaByte) yang mana hal tersebut tidak mungkin dapat dipenuhi oleh sistem microprocessor lama tersebut. Sedangkan untuk sistem microcontroller, program yang dijalankan biasanya tidak memerlukan sumber daya sebanyak dan sebesar itu. Untuk aplikasi kontrol sederhana dan tingkat menengah, microcontroller yang digunakan cukup berbasis 4 sampai 8 bit. Microcontroller dengan ukuran lebih besar (misal 16 dan 32 bit) umumnya hanya digunakan untuk aplikasi-aplikasi khusus pada bidang pengolahan citra atau bidang kontrol yang memerlukan kepresisian tinggi.

2.1.1. Perkembangan Mikrokontroler

Pada awal perkembangannya (yaitu sekitar tahun 1970-an), Sumber daya perangkat keras serta perangkat lunak microcontroller yang beredar masih sangat terbatas. Saat itu, sistem microcontroller hanya dapat diprogram secara khusus dengan perangkat yang dinamakan EPROM programmer. Sedangkan perangkat lunak yang digunakan umumnya berbasis bahasa assembler yang relatif sulit dipelajari.

Seiring dengan perkembangan teknologi *solid state* dan perangkat lunak komputer secara umum, saat ini pemrograman sistem microcontroller dirasakan relatif mudah dilakukan, terutama dengan digunakannya metode pemrograman ISP (*In system Programming*). Dengan menggunakan metode ini kita dapat memprogram sistem microcontroller sekaligus mengujinya pada sistem minimum atau papan pengembang (*development board*) secara langsung tanpa perlu lagi perangkat “pembakar“ program atau emulator secara terpisah. Selain itu, ditinjau dari aspek perangkat lunak pemrogramannya, dewasa ini banyak alternatif bahasa aras tinggi dari pihak ke-tiga, baik gratis maupun komersil yang dapat digunakan. Penggunaan bahasa aras tinggi ini (seperti Pascal, C, basic dan sebagainya) selain akan

menghemat waktu pengembangan, kode program yang disusun juga akan bersifat lebih modular dan terstruktur.

Bagi para pemula yang berminat memperdalam microcontroller baik sekedar untuk tujuan penyaluran hobi atau kelak untuk tujuan yang lebih profesional, dewasa ini banyak microcontroller dari berbagai vendor yang dapat dijadikan sarana untuk berlatih (misal microcontroller PIC produk Microchip, COP-8 produk National, AT89S51/52 dan AVR produk Atmel, HC11 produk Motorola dan lain sebagainya). Untuk memutuskan microcontroller mana yang akan dijadikan sarana berlatih dan akan diperdalam secara serius, ada baiknya hal-hal berikut ini dijadikan bahan pertimbangan :

- Apakah microcontroller tersebut mudah dijumpai dipasaran
- Apakah banyak dukungan pihak ketiga dalam penggunaan microcontroller tersebut, (misalnya perusahaan-perusahaan pembuat papan pengembang, pemasok compiler serta *debugger* untuk pemrogramannya, dan sebagainya)
- Apakah banyak referensi dan contoh-contoh program untuk panduan anda berlatih (misalnya dari sumber-sumber internet dan buku)
- Apakah banyak forum-forum diskusi (terutama di Internet) tempat anda dan programer lain berbagi pengalaman.

Dengan berbagai macam kelebihan yang dimiliki serta hal-hal yang menjadi bahan pertimbangan diatas, dewasa ini microcontroller AVR 8 bit produk perusahaan Atmel adalah salah satu microcontroller yang banyak merebut minat kalangan profesional dan juga cocok dijadikan sarana berlatih bagi para pemula. Hal ini selain karena ragam fitur yang ditawarkan, juga disebabkan kemudahan untuk memperoleh microcontroller tersebut (berikut papan pengembangnya) di pasaran dengan harga yang relatif murah. Selain itu berkaitan dengan rancangan arsitekturnya, microcontroller AVR ini juga cocok diprogram dengan menggunakan bahasa pemrograman aras tinggi (terutama bahasa C).

2.1.2. Sekilas Tentang Mikrokontroler Keluarga AVR

Secara histories microcontroller seri AVR pertama kali diperkenalkan ke pasaran sekitar tahun 1997 oleh perusahaan Atmel, yaitu sebuah perusahaan yang sangat terkenal dengan produk microcontroller seri AT89S51/52-nya yang sampai sekarang masih banyak digunakan di lapangan. Tidak seperti microcontroller seri AT89S51/52 yang masih mempertahankan arsitektur dan set instruksi dasar microcontroller 8031 dari perusahaan INTEL. Microcontroller AVR ini diklaim memiliki arsitektur dan set instruksi yang benar-benar baru dan berbeda dengan arsitektur microcontroller sebelumnya yang diproduksi oleh perusahaan tersebut. Tetapi walaupun demikian, bagi para programmer yang sebelumnya telah terbiasa dengan microcontroller seri AT89S51/52, dan berencana untuk beralih ke microcontroller AVR, maka secara teknis tidak akan banyak kesulitan yang berarti, hal ini dikarenakan selain konsep dan istilah-istilah dasarnya hampir sama, pemrograman level assembler-nya pun relative tidak jauh berbeda.

Berdasarkan arsitekturnya, AVR merupakan microcontroller RISC (*Reduce Instruction Set Computer*) dengan lebar bus data 8 bit. Berbeda dengan sistem AT89S51/52 yang memiliki frekuensi kerja seperduabelas kali frekuensi oscilator, frekuensi kerja microcontroller AVR ini pada dasarnya sama dengan frekuensi oscilator, sehingga hal tersebut menyebabkan kecepatan kerja AVR untuk frekuensi oscilator yang sama, akan dua belas kali lebih cepat dibandingkan dengan microcontroller keluarga AT89S51/52.

Dengan instruksi yang sangat variatif (mirip dengan sistem CISC-*Complex Instruction Set Computer*) serta jumlah register serbaguna (*general Purpose Register*) sebanyak 32 buah yang semuanya terhubung secara langsung ke ALU (*Arithmetic Logic Unit*), kecepatan operasi microcontroller AVR ini dapat mencapai 16 MIPS (enam belas juta instruksi per detik) —sebuah kecepatan yang sangat tinggi untuk ukuran microcontroller 8 bit yang ada di pasaran sampai saat ini.

Untuk memenuhi kebutuhan dan aplikasi industri yang sangat beragam, microcontroller keluarga AVR ini muncul di pasaran dengan tiga seri utama:

tinyAVR, ClasicAVR (AVR), megaAVR. Berikut ini beberapa seri yang dapat anda jumpai di pasaran:

- ATtiny13 AT90S2313 ATmega103
- ATtiny22 AT90S2323 ATmega128
- ATtiny22L AT90S2333 ATmega16
- ATtiny2313 AT90S4414 ATmega162
- ATtiny2313V AT90S4433 ATmega168
- ATtiny26 AT90S8515 ATmega8535

Keseluruhan seri AVR ini pada dasarnya memiliki organisasi memori dan set instruksi yang sama (sehingga dengan demikian jika kita telah mahir menggunakan salah satu seri AVR, untuk beralih ke seri yang lain akan relative mudah). Perbedaan antara tinyAVR, AVR dan megaAVR pada kenyataannya hanya merefleksikan tambahan-tambahan fitur yang ditawarkannya saja (misal adanya tambahan ADC internal pada seri AVR tertentu, jumlah Port I/O serta memori yang berbeda, dan sebagainya). Diantara ketiganya, megaAVR umumnya memiliki fitur yang paling lengkap, disusul oleh AVR, dan terakhir tinyAVR.

Untuk memberi gambaran yang lebih jelas, tabel 2.1. berikut memperlihatkan perbedaan ketiga seri AVR ditinjau dari jumlah memori yang dimilikinya.

Table 2.1. Perbedaan seri AVR berdasarkan jumlah memori

Microcontroller AVR		Memori (byte)		
Jenis	Paket IC	Flash	EEPROM	SRAM
TinyAVR	8–32 pin	1 – 2K	64 – 128	0 – 128
AVR (classic AVR)	20–44 pin	1 – 8K	128 – 512	0 – 1 K
MegaAVR	32–64 pin	8 – 128 K	512 – 4 K	512 – 4 K

2.1.3. Perlengkapan Dasar *Microcontroller*

- **CPU**

Unit pengolah pusat (CPU) terdiri atas dua bagian yaitu unit pengendali (CU) serta unit aritmatika dan logika (ALU). Fungsi utama unit pengendali adalah untuk mengambil, mengkode, dan melaksanakan urutan instruksi sebuah program yang tersimpan dalam memori. Sedangkan unit aritmatika dan perhitungan bertugas untuk menangani operasi perhitungan maupun boolean dalam program.

- **Alamat**

Pada *microprocesor / microcontroller*, apabila suatu alat dihubungkan dengan *microcontroller* maka harus ditetapkan terlebih dahulu alamat (address) dari alat tersebut. Untuk menghindari terjadinya dua alat bekerja secara bersamaan yang mungkin akan menyebabkan kerusakan.

- **Data**

Microcontroller ATmega16 mempunyai lebar bus data 8 bit. Merupakan *microcontroller* CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan.

- **Pengendali**

Selain bus alamat dan bus data *microprocesor/ microcontroller* dilengkapi juga dengan bus pengendali (control bus), yang fungsinya untuk menyerempakkan operasi *microprocesor / microcontroller* dengan operasi rangkaian luar.

- **Memori**

Microprocesor / microcontroller memerlukan memori untuk menyimpan program/data. Ada beberapa tingkatan memori, 30 diantaranya register internal, memori utama, dan memori massal. Sesuai dengan urutan tersebut waktu aksesnya dari yang lebih cepat ke yang lebih lambat.

- **RAM**

RAM (Random Acces Memory) adalah memori yang dapat dibaca atau ditulisi. Data dalam RAM akan terhapus bila catu daya dihilangkan. Oleh karena itu program *microcontroller* tidak disimpan dalam RAM. Ada dua teknologi yang dipakai untuk membuat RAM, yaitu RAM static dan RAM dynamic.

- **ROM**

ROM (Read Only Memory) merupakan memori yang hanya dapat dibaca. Data dalam ROM tidak akan terhapus meskipun catu daya dimatikan. Oleh karena itu ROM dapat digunakan untuk menyimpan program. Ada beberapa jenis ROM antara lain ROM murni, PROM, EPROM, EEPROM. ROM adalah memori yang sudah diprogram oleh pabrik, PROM dapat diprogram oleh pemakai sekali saja. Sedangkan EPROM merupakan PROM yang dapat diprogram ulang.

- **Input/ Output**

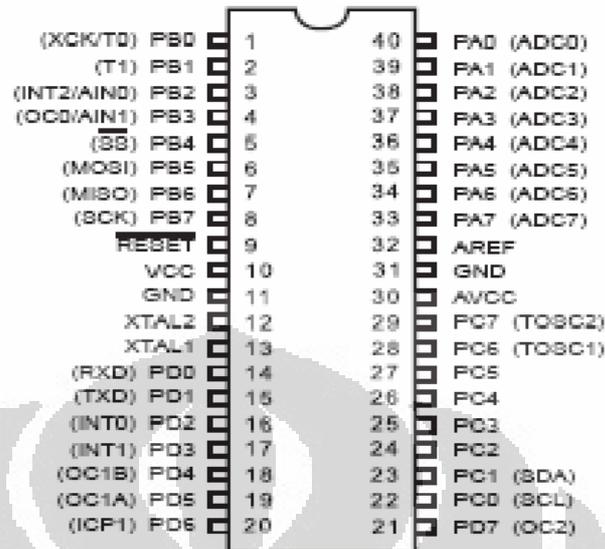
I/O dibutuhkan untuk melakukan hubungan dengan piranti di luar sistem. I/O dapat menerima data dari alat lain dan dapat pula mengirim data ke alat lain. Ada dua perantara I/O yang dipakai, yaitu piranti untuk hubungan serial (UART) dan piranti untuk hubungan paralel (PIO).

2.2. Minimum System

Rangkaian minimum System atau minsys merupakan komponen hardware yang sangat guna memamsukan perintah ke dalam sistem kerja dari perangkat mekanik. Rangkain minsys juga merupakan rangkaian utama yang digunakan sebagai induk hardware yang memiliki peran besar dalam proses pengontrolan. Ranagkain minsys menggunakan tipe – tipe IC yang berbeda, pada proyek tugas akhir ini digunakan tipe IC ATmega 16.

2.2.1 Konfigurasi Pin AVR ATmega 16

Pin-pin pada ATmega16 dengan kemasan 40-pin DIP (dual inline package) ditunjukkan oleh gambar 2.1. Kemasan pin tersebut terdiri dari 4 Port yaitu Port A, Port B, Port C, Port D yang masing masing Port terdiri dari 8 buah pin. Selain itu juga terdapat RESET, VCC, GND 2 buah, VCC, AVCC, XTAL1, XTAL2 dan AREF.



Gambar 2.1. Pin ATmega 16

Deskripsi dari pin-pin ATmega 16 adalah sebagai berikut :

1. **VCC** : Supply tegangan digital.
2. **GND** : Ground
3. **Port A** : Port A sebagai input analog ke A/D konverter. Port A juga sebagai 8-bit bi-directional port I/O, jika A/D konverter tidak digunakan. Pin-pin port dapat menyediakan resistor-resistor internal pull-up. Ketika port A digunakan sebagai input dan pull eksternal yang rendah akan menjadi sumber arus jika resistor- resistor pull-up diaktifkan. Pin-pin port A adalah tri- state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif.
4. **Port B** : Port B adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port B mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port B yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor- resistor pull-up diaktifkan. Pin-pin port B adalah tri- state ketika kondisi reset menjadi aktif sekalipun clock tidak aktif.

5. **Port C** : Port C adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port C mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port C yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor- resistor pull-up diaktifkan. Pin-pin port C adalah tri- state ketika kondisi reset menjadi aktif seklipun clock tidak aktif. Jika antarmuka JTAG enable, resistor- resistor pull-up pada pin-pin PC5(TDI), PC3(TMS), PC2(TCK) akan diktifkan sekalipun terjadi reset.
6. **Port D** : Port D adalah port I/O 8-bit bi-directional dengan resistor-resistor internal pull-up. Buffer output port D mempunyai karakteristik drive yang simetris dengan kemampuan keduanya sink dan source yang tinggi. Sebagai input, port D yang mempunyai pull eksternal yang rendah akan menjadi sumber arus jika resistor- resistor pull-up diaktifkan. Pin-pin port D adalah tri- state ketika kondisi reset menjadi aktif seklipun clock tidak aktif.
7. **Reset** : Sebuah low level pulsa yang lebih lama daripada lebar pulsa minimum pada pin ini akan menghasilkan reset meskipun clock tidak berjalan
8. **XTAL1** : Input inverting penguat Oscilator dan input intenal clock operasi rangkaian.
9. **XTAL2** : Output dari inverting penguat Oscilator.
10. **AVCC** : Pin supply tegangan untuk PortA dan A/D converter . Sebaiknya eksternalnya dihubungkan ke VCC meskipun ADC tidak digunakan. Jika ADCdigunakan seharusnya dihubungkan ke VCC melalui low pas filter.
11. **AREF** : Pin referensi analog untuk A/D konverter

2.2.2. Struktur Memori AVR ATmega 16

Untuk memaksimalkan performa dan paralelisme, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Instruksi pada memori program dieksekusi dengan pipelining single level. Selagi sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program.

1. Flash Memori

ATmega16 memiliki 16K byte flash memori dengan lebar 16 atau 32 bit. Kapasitas memori itu sendiri terbagi menjadi dua bagian yaitu *bagian boot program* dan *bagian aplikasi program*. Flash memori memiliki kemampuan mencapai 10.000 *write dan erase*.

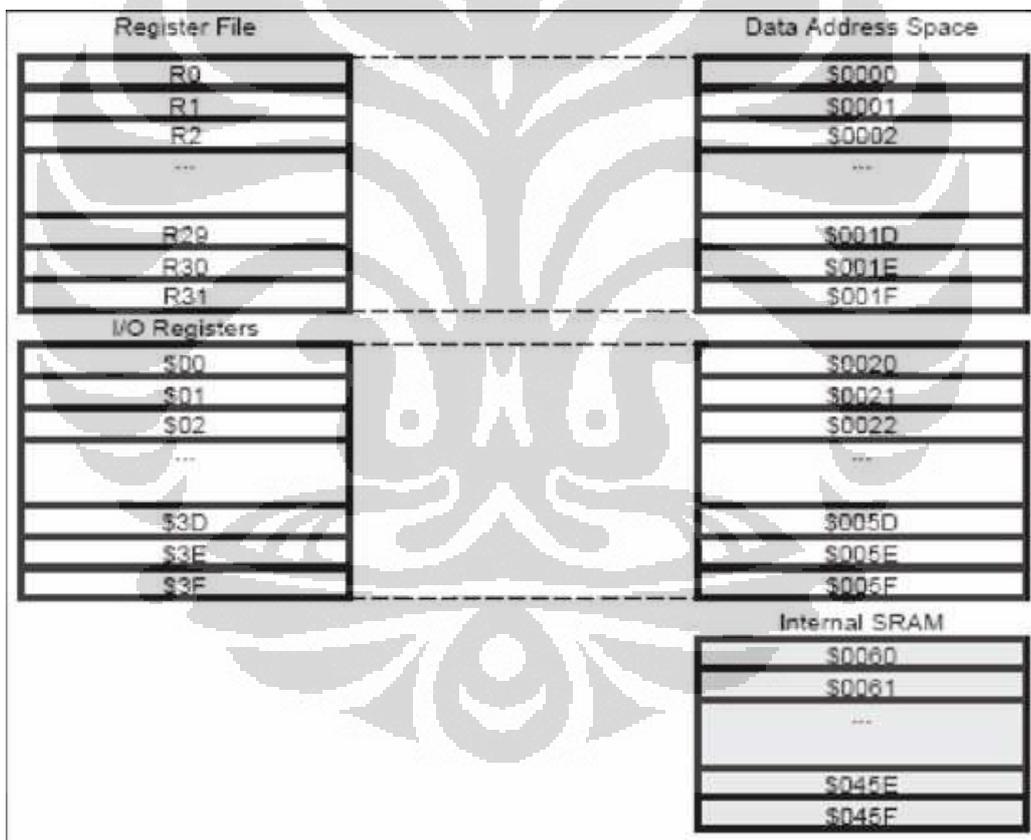


Gambar 2.2. Peta Memori Flash

2. Memori SRAM

Penempatan memori data yang lebih rendah dari 1120 menunjukkan register, I/O memori, dan data internal SRAM. 96 alamat memori pertama untuk file register dan memori I/O, dan 1024 alamat memori berikutnya untuk data internal SRAM.

Lima mode pengalamatan yang berbeda pada data memori yaitu direct, indirect, indirect displacement, indirect pre-decrement dan indirect post-increment. Pada file register, mode indirect mulai dari register R26-R31. Pengalamatan mode direct mencapai keseluruhan kapasitas data. Pengalamatan mode indirect displacement mencapai 63 alamat memori dari register X atau Y. Ketika menggunakan mode pengalamatan indirect dengan pre-decrement dan post increment register X, Y, dan Z akan di-decrement-kan atau di-increment-kan. Pada ATmega16 memiliki 32 register, 64 register I/O dan 1024 data internal SRAM yang dapat mengakses semua mode-mode pengalamatan.



Gambar 2.3 Peta Memori SRAM

3. Memori EEPROM

Pada EEPROM ATmega16 memiliki memori sebesar 512 byte engan daya tahan 100.000 siklus write/read.

Register-register pada memori EEPROM :

- **Bit 15...9** – Res:reserved bits Bit ini sebagai bit-bit bank pada ATmega16 dan akan selalu membaca

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W								
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

Gambar 2.4. Register alamat EEPROM Bit 15...8

- **Bit 8..0** – EEAR8..0:EEPROM address
Bit-bit ini sebagai alamat EEPROM.
- **Bit 7..0** – EEDR7..0:EEPROM
data Bit-bit ini sebagai data EEPROM.

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.5. Register data EEPROM Bit Bit 7...0⁶

- **Bit 7..4** – Res:reserved bits

Bit-bit ini terdapat pada register kontrol.

Bit ini sebagai *Enable Interupt Ready* pada EEPROM.

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

Gambar 2.6. Register kontrol EEPROM Bit Bit 7...0

- **Bit 2**
Bit ini sebagai *Enable Interupt Master* pada EEPROM.
- **Bit 1**
Bit ini sebagai *write enable* pada EEPROM.
- **Bit 0**
Bit ini sebagai *read enable* pada EEPROM.

2.3. Sensor Inframerah

Sensor infrared banyak digunakan sebagai sistem komunikasi seperti halnya dalam proses pengiriman data. Sensor infrared juga biasa digunakan untuk mendeteksi suatu benda.

Sifat-sifat cahaya infrared yaitu :

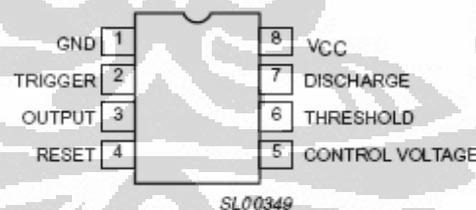
- Tidak tampak manusia
- Tidak dapat menembus materi yang tak tembus pandang
- Dapat ditimbulkan oleh komponen yang menghasilkan panas

Berbeda dengan LED biasa, LED Infrared pada penggunaannya dapat diaktifkan dengan Tegangan DC Untuk transmisi/sensor jarak dekat dan Tegangan AC (30 – 40 KHz) Untuk transmisi/sensor jarak jauh. Infrared sebagai Komunikasi Data dengan tegangan AC (30 – 40 KHz) berfungsi sebagai carrier dan data dimodulasikan dalam tegangan AC tersebut. Komponen-komponen elektronik yang

menggunakan infrared yaitu seperti Photo transistor, Photo diode. Opto Coupler, Opto Diac, dan juga Infrared Module.

Cahaya infra merah adalah suatu pilihan yang cukup baik sebagai sensor karena cahaya spektrum cahaya ini berada di luar spektrum cahaya matahari sehingga tidak mengganggu kinerja sensor. Sensor infra merah ini difungsikan sebagai transmitter dimana digunakan phototransistor sebagai receiver. Rangkaian timer infrared akan menghasilkan keluaran frekuensi sebesar 38 KHz yang nantinya akan diterima sensor phototransistor secara kontinu.

Signal keluaran pada sensor masih berupa signal analog, kemudian dirubah menjadi signal digital, dalam blok sensor terdapat rangkaian transmitter dan receiver yang mempunyai fungsi untuk memberikan logika 1 dan 0 pada rangkaian keluaran, logika- logika itulah yang dibutuhkan oleh komputer untuk dapat diolah. Rangkaian transmitter sendiri terdiri dari beberapa komponen antara lain seperti kapasitor, led dan juga IC 555. Alat ini bekerja berdasarkan kerja IC 555 yang memberikan keluaran berupa tegangan, frekuensi yang beraturan. Keluaran ini digunakan untuk menghasilkan data yang terpancarkan oleh led infra merah. Kemudian pada penggunaannya, cahaya tersebut diterima oleh photo transistor yang ada pada rangkaian receiver, sehingga dapat digunakan untuk memicu rangkaian receiver



Gambar 2.7. Pin IC 555

Penggunaan receiver sebagai kelanjutan dari rangkaian transmitter. Rangkaian ini terdiri dari beberapa komponen utama, seperti halnya phototransistor. Setelah sinar infra merah dipancarkan oleh rangkaian transmitter, sinar tersebut diterima oleh photo transistor sebagai masukan. Adapun sinar tersebut berpengaruh pada perubahan arus dan tegangan yang terhubung dengan photo transistor. Perubahan yang terjadi

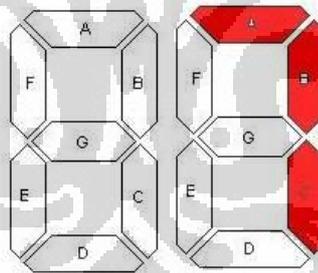
membuat tegangan menjadi berkurang sehingga berpengaruh pada kerja sistem rangkaian.

Keadaan data yang dalam keadaan normal adalah berupa nilai tegangan dengan keluaran berbentuk logika 0. Dan pada saat itu transmitter sendiri dapat memberikan signal kepada reciever tanpa terhalang. Keadaan ini berubah ketika reciever tidak lagi mendapatkan signal dari transmitter. Hal ini menyebabkan rangkaian bekerja dan menghasilkan logika 1.

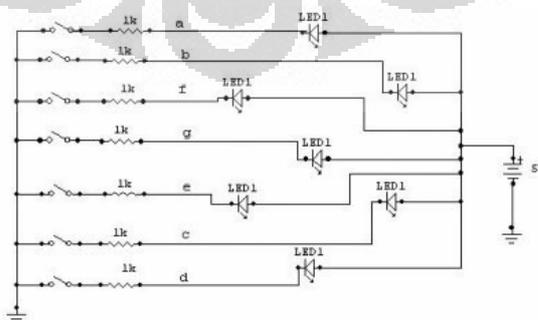
2.4. Seven Segment

Seven Segmen merupakan peraga angka yang dapat memperagakan angka dari 0 sampai 9. Seven Segment ini dibentuk oleh 7 buah segmen yang tersusun sedemikian rupa sehingga jika ada beberapa segmen tertentu yang aktif maka akan membentuk sebuah angka.

Jika segmen A, B, dan C hidup, maka angka yang akan ditampilkan adalah 7. demikian seretusnya.



Gambar 2.8. Seven Segment



Gambar 2.9. Rangkaian Dasar Seven Segment

Jika saklar di a ditutup, maka arus mengalir dari Ground melalui tahanan pembatas hingga LED segmen-a dan keluar melalui hubungan anoda biasa ke catu daya. Dengan demikian hanya segmen a saja yang menyala. Jika kita ingin menampilkan angka 8 desimal pada peraga, saklar a, b, c, d, e, f, dan g ditutup, sehingga pada rangkaian akan menyala decimal 8. Dalam hal ini tegangan ground (LOW) mengaktifkan segmen LED.

2.5. LED

Light Emmiting Dioda atau lebih dikenal dengan sebutan **LED** (*light-emitting diode*) adalah suatu semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju.

Sebuah LED adalah sejenis dioda semikonduktor istimewa. Seperti sebuah dioda normal, dia terdiri dari sebuah chip bahan semikonduktor yang diisi penuh, atau di-dop, dengan ketidakmurnian untuk menciptakan sebuah struktur yang disebut p-n junction. Pembawa-muatan elektron dan lubang mengalir ke junction dari elektroda dengan voltase berbeda. Ketika elektron bertemu dengan lubang, dia jatuh ke tingkat energi yang lebih rendah, dan melepas energi dalam bentuk photon.

2.6. Timer

Timer/Counter merupakan fitur di setiap *microcontroller* yang sangat banyak peranannya dalam perancangan sebuah sistem. Dengan sebuah timer kita bisa menunda suatu eksekusi dari suatu perintah, sehingga sistem *microcontroller* yang berjalan cepat (mungkin dalam orde nano atau mikro detik), dapat sinkron dengan operator (manusia). Dengan timer pula kita dapat menghitung lebar pulsa/frekuensi dari suatu sinyal, menghitung kecepatan jatuh suatu benda, atau kecepatan suatu kendaraan, dengan timer/counter kita dapat menghitung jumlah mobil yang lewat dalam jalan tol misalnya, dan ada lebih banyak lagi fungsi timer/counter dalam sebuah *microcontroller*. *Microcontroller* AVR keluaran Atmel memiliki beberapa timer(ATmega16 dalam hal ini, tipe lain yang lebih tinggi sama, kecuali penambahan jumlah timernya), diantaranya timer0(8 bit), timer1(16 bit), dan timer2(8 bit).

Timer0 dan timer2 (8 bit) menghitung maksimal 255 hitungan. Dimana periode setiap hitungan(detaknya) tergantung dari setting prescaler-nya. Untuk mengatur timer mode operasi apa yang digunakan dan mengatur prescaler digunakan register TCCR_x(x=0,2). TCCR_x adalah register 8 bit sebagai berikut:

FOC0 WGM00 COM01 COM00 WGM01 CS2 CS1 CS0
7 6 5 4 3 2 1 0

CS0,CS1,CS2 digunakan untuk pengaturan prescaler, misalnya

0 0 0 : timer dihentikan jika diset nilai ini

0 0 1 : sekali detaknya sama dengan detak oscillator crystal yang digunakan

0 1 0 : sekali detaknya sama dengan oscillator/8

Mode-Mode kerja Timer

1. mode normal, timer digunakan untuk menghitung saja, membuat delay, menghitung selang waktu.
2. mode PWM, phase correct
3. CTC, dalam mode ini, nilai timer yang ada pada TCNT_x akan di nol-kan lagi jika TCT_x sudah *match* dengan nilai yang ada pada register OCR_x. Sebelumnya OCR diset dulu, karena timer0 dan timer2 maksimumnya 255, maka range OCR 0-255.
4. Fast PWM

Jadi hal utama jika akan menggunakan timer adalah setting TCCR_x-nya, digunakan untuk mode apa (secara default, adalah sebagai mode normal) dan ketika saat timer jalan, jumlah hitungannya dapat dilihat di register TCNT_x.

Interrupt Timer

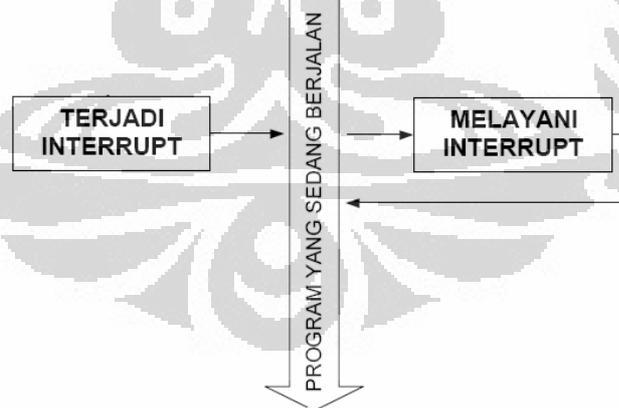
1. Overflow Interrupt, akan terjadi interrupt jika TCNT_x mencapai 255 untuk timer0, dan 255 untuk timer2.
2. Compare Match interrupt terjadi jika nilai OCR sama dengan TCNT_x

Contoh Aplikasi: Jam digital. Digunakan Compare Match Interrupt, menggunakan crystal 12 MHz, digunakan timer 0. Dengan prescaler Osc/8: $12\text{mhz}/8=1,5\text{ mhz}$ periode $1/1,5\text{mhz}$ ==sekitar 100 us, jadi agar setara 1 detik harus dikali 10000.

2.7. Interrupt

Interrupt adalah suatu kejadian atau peristiwa yang menyebabkan mikrokontroler berhenti sejenak untuk melayani interrupt tersebut. Program yang dijalankan pada saat melayani interrupt disebut **Interrupt Service Routine**. Analoginya adalah sebagai berikut, seseorang sedang mengetik laporan, mendadak telephone berdering dan menginterruptsi orang tersebut sehingga menghentikan pekerjaan mengetik dan mengangkat telephone. Setelah pembicaraan telephone yang dalam hal ini adalah merupakan analogi dari Interrupt Service Routine selesai maka orang tersebut kembali meneruskan pekerjaannya mengetik.

Demikian pula pada sistem *microcontroller* yang sedang menjalankan programnya, saat terjadi interrupt, program akan berhenti sesaat, melayani interrupt tersebut dengan menjalankan program yang berada pada alamat yang ditunjuk oleh vektor dari interrupt yang terjadi hingga selesai dan kembali meneruskan program yang terhenti oleh interrupt tadi. Seperti yang terlihat Gambar 2.10, sebuah program yang seharusnya berjalan terus lurus, tiba-tiba terjadi interrupt dan harus melayani interrupt tersebut terlebih dahulu hingga selesai sebelum ia kembali meneruskan pekerjaannya.



Gambar 2.10. Skema Interrupt

Proses yang terjadi saat *microcontroller* melayani interrupt adalah sebagai berikut:

- Instruksi terakhir yang sedang dijalankan diselesaikan terlebih dahulu
- Program Counter (alamat dari instruksi yang sedang berjalan) disimpan ke stack
- Interrupt Status disimpan secara internal
- Interrupt dilayani sesuai peringkat dari interrupt (lihat Interrupt Priority)
- Program Counter terisi dengan alamat dari vector interrupt (lihat Interrupt Vector) sehingga *microcontroller* langsung menjalankan program yang terletak pada vector interrupt

Program pada vector interrupt biasanya diakhiri dengan instruksi RETI di mana pada saat ini proses yang terjadi pada *microcontroller* adalah sebagai berikut:

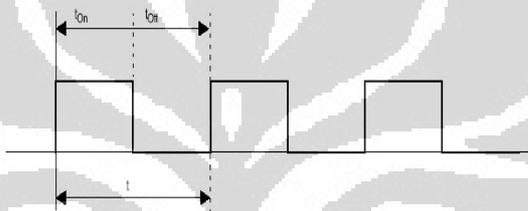
- Program Counter diisi dengan alamat yang tersimpan dalam stack pada saat interrupt terjadi sehingga *microcontroller* kembali meneruskan program di lokasi saat interrupt terjadi
- Interrupt Status dikembalikan ke kondisi terakhir sebelum terjadi interrupt

Tabel 2.2. Interrupt vektor

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset.
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$008	TIMER2 OVF	Timer/Counter2 Overflow
6	\$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$010	TIMER1 OVF	Timer/Counter1 Overflow
10	\$012	TIMER0 OVF	Timer/Counter0 Overflow
11	\$014	SPI, STC	Serial Transfer Complete
12	\$016	USART, RXC	USART, Rx Complete
13	\$018	USART, UDRE	USART Data Register Empty
14	\$01A	USART, TXC	USART, Tx Complete
15	\$01C	ADC	ADC Conversion Complete
16	\$01E	EE_RDY	EEPROM Ready
17	\$020	ANA_COMP	Analog Comparator
18	\$022	TWI	Two-wire Serial Interface
19	\$024	INT2	External Interrupt Request 2
20	\$026	TIMER0 COMP	Timer/Counter0 Compare Match
21	\$028	SPM_RDY	Store Program Memory Ready

2.8. PWM (*Pulse Width Modulation*)

Suatu teknik yang digunakan untuk mengontrol kerja dari suatu alat atau menghasilkan suatu tegangan DC yang variabel adalah *PWM (Pulse width Modulation)*. Rangkaian *PWM* adalah rangkaian yang lebar pulsa tegangan keluarannya dapat diatur atau dimodulasi oleh sebuah sinyal tegangan modulasi. Disamping itu kita dapat menghasilkan suatu sinyal *PWM* dengan menentukan frekuensi dan waktu dari variabel ON dan OFF. Pemodulasian sinyal yang beragam dapat menghasilkan *duty cycle* yang diinginkan. Gambar dibawah memperlihatkan sinyal kotak dengan *duty cycle* 50%.



Gambar 2.11. Sinyal PWM dengan *duty cycle* 50%

$$D = \frac{t_1}{t_1 + t_2} \dots\dots\dots(2.1)$$

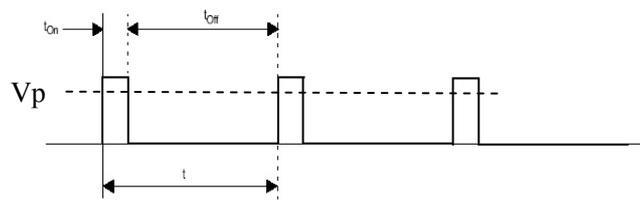
Dimana :

D = Duty Cycle

T₁ = T_{on} (Waktu ON)

T₂ = T_{off} (Waktu OFF)

Dengan *duty cycle* yang bermacam-macam, rata-rata output dari tegangan dc dapat dikontrol. Seperti pada gambar 3.3 memperlihatkan sinyal kotak dengan *duty cycle* 10%.



Gambar 2.12. Sinyal PWM dengan *duty cycle* 10%.

Dari gambar diatas maka kita dapat mengetahui nilai RMS nya dimana :

$$V(t) \begin{cases} 0 \rightarrow t \\ t \rightarrow T \end{cases} \quad (2.2)$$

$$V_{RMS} = \sqrt{\frac{\int_0^{T_1} V(t)^2 dt}{T}} = \sqrt{\frac{\int_0^t V_p^2 dt + \int_t^T 0 dt}{T}} \quad (2.3)$$

$$V_{RMS} = \sqrt{\frac{\int_0^t V_p^2 dt}{T}} = \sqrt{\frac{V_p^2 t}{T}} \quad (2.4)$$

$$V_{RMS} = \sqrt{\frac{V_p^2 t}{T}} = V_p \sqrt{\frac{t}{T}} \quad (2.5)$$

$$P = \frac{V_{rms}^2}{R} \quad (2.6)$$

$$P = \frac{V_p^2}{R} \cdot \frac{t}{T} \quad (2.7)$$

2.9. BASCOM AVR

Pada aplikasi ini digunakan *microcontroller* jenis AVR produksi ATMEL.

Keunggulan dari jenis *microcontroller* ini adalah memiliki:

1. Kecepatan

Untuk mengeksekusi 1 buah instruksi *microcontroller* AVR hanya memerlukan 1 clock sedangkan MCS51 dalam hal ini AT89S51 memerlukan 12 clock. Jadi jelas AVR lebih cepat dari MCS51

2. Bahasa Pemrograman

Bahasa pemrograman yang digunakan oleh AVR adalah bahasa C (*salah satunya, red*). Sehingga lebih mudah dipahami daripada bahasa assembly. Memang bahasa assembly lebih dekat ke bahasa mesin sehingga lebih cepat eksekusinya. Tapi, pihak Atmel telah mengoptimasi bahasa C yang dipakai untuk AVR sehingga lebih cepat dan efisien.

3. Memory yang lebih besar

Untuk keluarga *microcontroller* AVR sudah banyak yang memiliki memory internal yang relatif besar. Misal untuk seri ATmega16 : 16 Kb (Flash memory), 512 Bytes (eeprom), 1 Kb (RAM). sedangkan untuk AT89S51 memiliki 4 Kb (Flash memory), 128 bytes (RAM). Dari sini terlihat kalau kapasitas memory AVR lebih besar daripada AT89S51.

4. Efisiensi Hardware

Dengan memakai AVR kita bisa meminimalisasi penggunaan komponen pendukung seperti EEPROM eksternal (*baca poin 3*), sehingga kita tidak perlu lagi EEPROM eksternal. Untuk sumber clock AVR telah menyediakannya secara terintegrasi. Karena didalam AVR sudah ada XTAL yang bisa diaktifkan sehingga kita tidak perlu memakai XTAL lagi. Untuk nilainya bisa kita kalibrasi sendiri apakah 4 Mhz, 1Mhz atau lainnya.

BASCOM AVR merupakan singkatan dari *Basic Compiler AVR*. BASCOM AVR termasuk dalam program *microcontroller* buatan MCS Electronics yang mengadaptasi bahasa tingkat tinggi yang sering digunakan oleh awam (Bahasa Basic). Dengan menggunakan bahasa pemrograman tingkat tinggi, maka pemrogram mendapatkan banyak kemudahan dalam mengatur sistem kerja dari *microcontroller* (terutama apabila tipe *microcontroller* yang digunakan masih baru dan tidak terlalu dikenal). BASCOM AVR memiliki program sendiri untuk memasukkan program yang telah dikompilasi ke dalam AVR. Untuk ATmega16 yang menggunakan kabel ISP, konfigurasi dari *programmer* harus diubah menjadi STK300/STK200.

BAB 3

PERANCANGAN DAN PEMBUATAN PROGRAM

Pada pembuatan "Programmable Timer untuk Percobaan Tumbukan" untuk proyek akhir ini secara umum terbagi menjadi dua bagian pokok, yaitu pembuatan perangkat keras dan perangkat lunak. Pembuatan perangkat lunak untuk percobaan tumbukan ini dirancang dengan menggunakan BasCom AVR. Untuk pembuatan perangkat lunak yang meliputi pembuatan program untuk timer, seven segment, LED, dan solenoid akan dibahas pada bab ini yang merupakan pokok bahasan utama dalam proyek akhir ini.

Sebagai interface dalam sistem ini digunakan minimum sistem ATmega16 yang akan mengontrol perangkat keras melalui port-port yang tersedia. Port A dan port C dari *microcontroller* digunakan untuk menampilkan data ke 4 buah seven segment, dimana data tersebut berupa nilai dari timer. Pada port B dihubungkan dengan display untuk LED dan juga untuk menggerakkan solenoid, pembagiannya adalah dari port B.0 sampai port B.6 untuk LED, dan port B.7 untuk penggerak solenoid. Port D untuk input dan juga output, yaitu port D.0 dan port D.1 sebagai output untuk member logic pada rangkaian pemilihan, port D.2 dan port D.3 untuk input sebagai interrupt, port D.4 dan port D.5 sebagai output untuk PWM yang mengatur kecerahan seven segment, port D.6 dan port D.7 sebagai input dari 2 buah pushbutton.

3.1. Konfigurasi Sistem

Secara umum konfigurasi sistem dari alat ini terdiri dari input, controller dan output. Dari bagian besar tersebut didalamnya terdapat perangkat keras (hardware) dan lunak (software). Dari sisi masukan (*input*) terdiri 2 buah sensor yaitu sensor infrared type TSOP38. Untuk controller, akan digunakan sebuah mikrocontroller AVR ATmega16. Dari sisi keluaran (*output*) terdapat driver motor. Pada proyek akhir ini akan dipakai 2 buah solenoid sebagai penggerak kereta. Sedangkan untuk

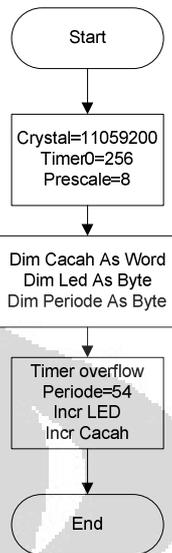
pengaturan kecerahan seven segment dilakukan dengan menggunakan metode PWM (*Pulse Width Modulation*). PWM (*Pulse Width Modulation*) yaitu merupakan sebuah teknik pemodulasian lebar pulsa yang dapat dimanfaatkan pada pengaturan kecepatan motor servo dan dapat juga digunakan untuk mengatur kecerahan seven segment. Pada sinyal PWM yang dikontrol adalah duty cycle dari sinyal, dimana frekwensi dari sinyal adalah tetap. Blok diagram sistem dapat dilihat pada gambar 1.2

Dari gambar diatas terlihat jika interrupt yang berasal dari 1 ataupun 2 buah sensor akan masuk ke *microcontroller* sebagai interrupt lalu akan menjalankan timer ataupun juga menghentikan timer. Data timer yang dihasilkan akan ditampilkan lewat LED dan seven segment. Dimana setiap 1 buah LED bernilai 10 ms dan setiap 100 LED menyala maka nilai tampilan seven segment akan bertambah.

3.2. Sistem Pewaktu

Dalam ATmega16 terdapat 3 macam timer, yaitu timer0, timer1, dan timer2. Nilai dari timer0 dan timer2 yaitu 8 bit, sedangkan timer1 bernilai 16 byte. Timer yang akan digunakan sebagai pewaktu adalah timer yang bernilai 8 byte. Sehingga timer yang digunakan adalah timer0.

Pada pembuatan sistem pewaktu ini ingin didapatkan 1 data setiap 10 ms, dari data yang didapat tersebut akan dikirim ke LED. Untuk mendapatkan 1 data setiap 10 ms maka dipergunakan crystal 11059200, lalu menggunakan timer0 karena timer ini bernilai 8 byte atau sama saja dengan 256, dan dengan mengatur nilai prescale dari timer0 menjadi 8. Setelah itu semua dilakukan maka akan didapat nilai 5400. Timer0 overflow juga diperlukan dalam pembuatan sistem pewaktuan ini. Timer overflow adalah fungsi interrupt internal dari sebuah timer. Pada sistem ini overflow dari timer0 akan diberi nilai sebesar 54, sehingga jika sudah terjadi 54 kali hitungan maka nilai overflow akan kembali ke awal. Maka jika nilai 5400 dibagi dengan nilai 54 dari overflow maka akan didapat 100. Hal ini menunjukkan jika dalam 1 detik(1000 ms) akan terjadi 100 kali pengiriman data, maka 1 kali pengiriman data memerlukan waktu selama 10 ms Sehingga jika data dari overflow dikirim ke LED maka setiap 10 ms LED akan menyala secara bergantian sampai LED ke 100 yang berarti 1 detik.



Gambar 3.1. Sistem Pewaktu

Program untuk sistem pewaktu

```

Time_sampling_isr:
  Periode = Periode + 1
  If Periode = 54 Then
    Periode = 0
    Led = Led + 1
    If Led = 100 Then
      Led = 0
      Cacah = Cacah + 1
      If Cacah = 10000 Then Cacah = 0
    End If
  End If
End If
Return
  
```

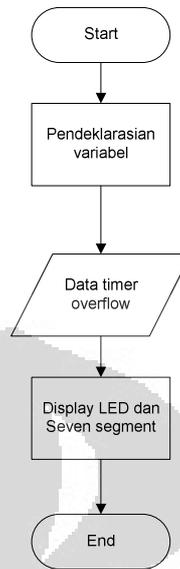
3.3. Display LED dan Seven Segment

LED yang digunakan sebanyak 100 buah. Fungsi dari LED tersebut sebagai display dari seperseratus detik, untuk membuat 2 angka dibelakang koma. Setiap LED bernilai 10 ms dan akan menyala secara bergantian. Port pada Minsys yang digunakan untuk LED adalah portB.0 sampai portB.6.

Cara menghidupkan keseratus LED tersebut secara bergantian dengan mengirimkan data dari timer ke port LED. Maka jika data yang didapat dari timer berubah maka LED yang menyala pun akan ikut berubah. 1 LED bernilai 10 ms maka jika 100 LED akan bernilai 1000 ms atau 1 detik.

Seven segment pada MinSys menggunakan 2 buah port yaitu portA dan portC. Setiap port diisi oleh 2 buah seven segment. Seven segment yang digunakan bersifat common Anoda. Keempat seven segment yang digunakan mempunyai fungsi sebagai penunjuk detik yang berbeda. Yaitu untuk menampilkan satuan, puluhan, ratusan lalu ribuan.

Cara menghidupkan nilai tampilan dari seven segment sesuai dengan nilai timer yang ada, yaitu dengan mengirim nilai timer ke port dari seven segment dengan terlebih dahulu menggunakan fungsi **makeBCD** dari BASCOM, atau dengan menggunakan nilai dari yang di Mod kan lalu dikali 16. Hal ini dikarenakan sebelum data sampai ke seven segment terlebih dahulu data melewati IC decoder. IC tersebut menerima data BCD lalu mengirimkannya ke seven segment.



Gambar 3.2. Display LED Seven Segment

Program untuk display LED dan seven segment

```

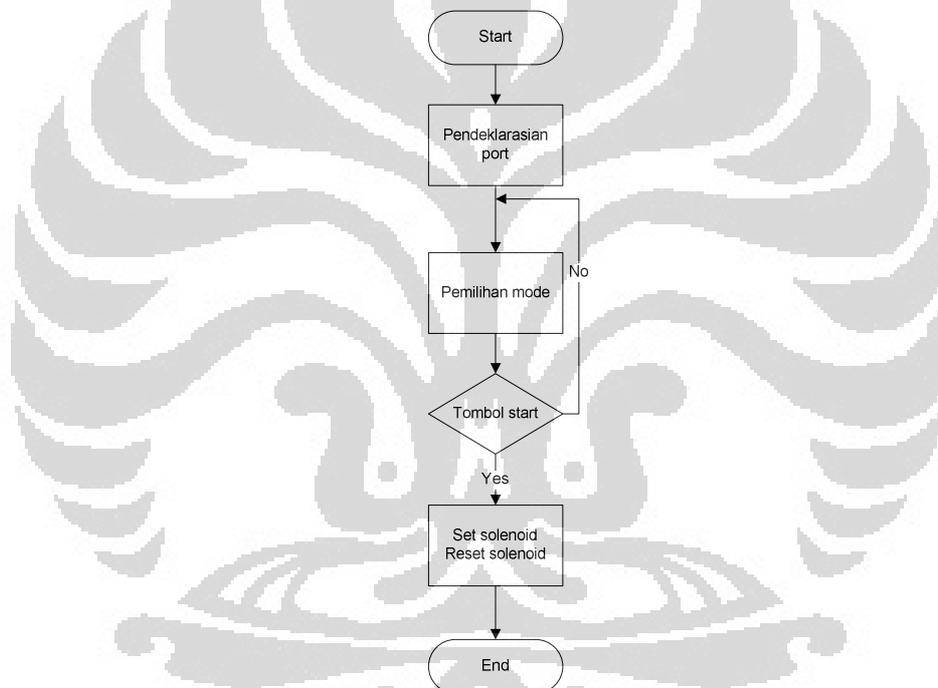
Led_display = Led + Punch
Satuan = Cacah Mod 10
Sisa_satuan = Cacah - Satuan
Sisa_satuan = Sisa_satuan / 10
Puluhan = Sisa_satuan Mod 10
Sisa_puluhan = Sisa_satuan - Puluhan
Sisa_puluhan = Sisa_puluhan / 10
Ratusan = Sisa_puluhan Mod 10
Sisa_ratusan = Sisa_puluhan - Ratusan
Sisa_ratusan = Sisa_ratusan / 10
Ribuan = Sisa_ratusan Mod 10
Digit12 = 16 * Puluhan
Digit12 = Digit12 + Satuan
Digit34 = 16 * Ribuan
Digit34 = Digit34 + Ratusan
Portc = Digit12
  
```

Porta = Digit34

Portb = Led_display

3.4. I/O Hardware

I/O hardware yang dimaksud disini adalah input dari 2 buah push button dan output untuk menggerakkan solenoid. Push button digunakan sebagai pemilihan mode dan sebagai tombol start, port D.6 sebagai pemilihan(mode) dan port D.7 sebagai start. Port B.7 dihubungkan dengan driver motor yang nantinya digunakan untuk menghidupkan solenoid.



Gambar 3.3. Push button dan solenoid

Programnya sebagai berikut :

Do

If Pind.6 = 0 Then Mode_timer = Mode_timer + 1

Waitms 100

Mode_timer_bcd = Makebcd(mode_timer)

Portc = Mode_timer_bcd

Loop Until Pind.7 = 0

Portb = 128

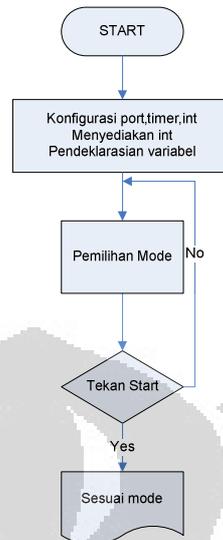
Portb = 0

3.5. Mode Timer

Mode disini adalah pilihan dalam penggunaan sistem waktu yang akan digunakan. Sistem waktu berhubungan dengan penggunaan sensor, baik dari jumlah sensor maupun fungsi dari sensor tersebut. Jumlah penggunaan sensor pada tiap paling banyak 2 buah. Sensor-sensor tersebut berfungsi sebagai trigger untuk mengaktifkan maupun menonaktifkan timer.

Sensor yang digunakan adalah sensor infrared. Sensor infrared dihubungkan dengan kaki interrupt pada MinSys. Sensor tersebut mempunyai sifat akan high atau menghasilkan logika 1 jika ada sesuatu yang menghalanginya menerima pancaran dari infrared, jika tidak ada yang menghalanginya mendapat pancaran dari infrared maka sensor akan menghasilkan output low atau menghasilkan logika 0. Tapi karena input yang dibutuhkan oleh kaki interrupt pada MinSys akan aktif ketika low, maka keluaran dari sensor terlebih dahulu dilalui ke IC pembalik.

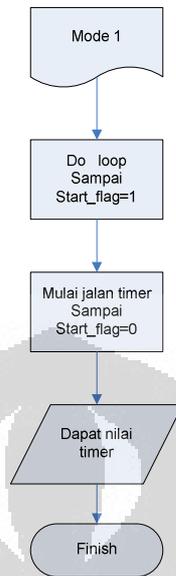
Mode yang telah berhasil dibuat ada 3 buah dimana dari setiap mode mempunyai tipe yang berbeda dalam tiap cara kerjanya. Pemilihan penggunaan mode dilakukan dengan cara menekan push button.



Gambar 3.4. Flowchart Utama

3.5.1 Mode 1

Pada mode 1 ini sensor yang digunakan 2 buah yang dihubungkan pada 2 interrupt. Dimana satu sensor berfungsi untuk menjalankan timer, dan sensor satunya lagi untuk mematikan timer. Saat 1 sensor aktif maka interrupt aktif sehingga akan memulai timer, sampai sensor yang lain belum aktif maka interrupt yang lain juga belum aktif maka timer akan terus berjalanan. Fungsi dari mode ini yaitu untuk mengukur waktu tempuh diantara 2 sensor. Selain itu mode ini juga dapat digunakan sebagai pewaktu untuk melakukan percobaan tumbukan.



Gambar 3.5. Flowchart Mode 1

3.5.2 Mode 2

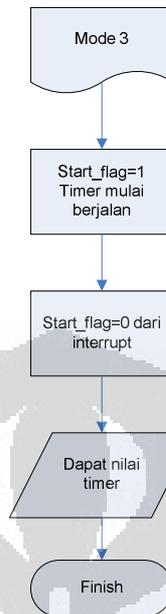
Untuk mode 1 ini sensor yang digunakan hanya 1 buah untuk mengaktifkan satu interrupt. Satu sensor itu berfungsi untuk menjalankan timer dan juga digunakan untuk mematikan timer. Saat sensor aktif pertama kali maka interrupt akan aktif, ketika interrupt aktif ada satu variable yang nilainya bertambah, maka variable tersebut akan bernilai 1 membuat timer akan mulai berjalan. Ketika sensor aktif lagi maka interrupt juga akan aktif lagi dan variable tadi nilainya bertambah juga, nilai variable menjadi 2 menyebabkan timer berhenti berjalan. Mode ini dapat digunakan untuk mengukur waktu tempuh panjang kereta yang digunakan dan dapat digunakan untuk percobaan tumbukan.



Gambar 3.6. Flowchart Mode 2

3.5.3 Mode 3

Pada mode 3 ini sensor yang digunakan juga hanya satu untuk mengaktifkan satu interrupt yang nantinya digunakan untuk menonaktifkan timer. Cara kerja mode 3 sebagai berikut, saat mode 3 dipilih lalu tombol Start ditekan maka selenoid akan bekerja dan secara bersamaan timer akan mulai berjalan. Ketika sensor aktif maka interrupt juga aktif sehingga membuat timer berhenti. Mode ini berfungsi mengukur waktu tempuh dari selenoid sampai sensor.



Gambar 3.7. Flowchart Mode 3

BAB 4

PENGUJIAN DAN ANALISA

Dalam Bab ini akan dibahas tentang pengujian berdasarkan perencanaan dari sistem yang dibuat. Program pengujian disimulasikan di suatu sistem yang sesuai. Pengujian ini dilaksanakan untuk mengetahui kehandalan dari sistem dan untuk mengetahui apakah sudah sesuai dengan perencanaan atau belum.

4.1. Pengujian dan Analisa Data Timer

Pengujian timer dilakukan untuk membuktikan jika sistem pewaktu yang dibuat sesuai dengan *real time* yang ada. Dalam pengujian dilakukan dengan membandingkan waktu yang didapat dari sistem pewaktu dengan stopwatch dari handphone Siemens tipe C60. Handphone tersebut dipilih untuk digunakan sebagai pembanding karena nilai yang didapat bisa mencapai 1/1000 detik, sedangkan sistem pewaktu yang dibuat hanya 1/100 detik. Sehingga diharapkan waktu yang didapat dari stopwatch lebih akurat. Pengujian yang dilakukan yaitu pada waktu 100 detik, 500 detik, 1000 detik dan 9999 detik. Data yang diambil sebanyak 20 kali dari tiap waktu, terkecuali untuk waktu 9999 detik hanya dilakukan sebanyak 3 kali. Pengambilan data dilakukan sebanyak 20 kali agar meminimalisir kesalahan dalam menekan stopwatch. Untuk pengujian pada waktu 100 detik didapat data sebagai berikut :

Tabel 4.1. Pengujian untuk 100 detik

No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)	No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)
1	1000,09	16'48,129	6	500,04	8'26,161
2	1000,88	16'48,983	7	500,06	8'26,137
3	1000,01	16'47,972	8	500,04	8'26,127
4	999,97	16'48,000	9	500,07	8'26,238
5	1000,08	16'48,167	10	500,09	8'26,183

11	1000,15	16'48,215	16	500,12	8'26,183
12	1000,12	16'48,203	17	500,14	8'26,253
13	1000,26	16'48,337	18	500,05	8'26,196
14	1000,07	16'48,147	19	500,08	8'26,156
15	1000,18	16'48,285	20	500,14	8'26,178

Pada pengujian untuk waktu 100 detik didapat nilai dari sistem pewaktu yang bervariasi tetapi tetap mendekati nilai 100 detik. Begitu juga dengan nilai dari stopwatch mendekati nilai 1 menit 40 detik. Nilai tertinggi dari sistem pewaktu yaitu 100,39 detik, dan nilai tertinggi stopwatch yaitu 1'40,481. Nilai terendah dari sistem pewaktu yaitu 99,94 detik, dan nilai terendah stopwatch yaitu 1'40,015. Bila nilai waktu dari sistem pewaktu dirata-ratakan maka akan didapat 100,101 detik. Dan nilai rata-rata dari stopwatch 100,159 detik. Untuk pengujian pada waktu 500 detik didapat data sebagai berikut :

Tabel 4.2. Pengujian untuk 500 detik

No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)	No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)
1	499,87	8'25,778	11	500,04	8'26,161
2	500,05	8'26,238	12	500,06	8'26,137
3	499,95	8'25,887	13	500,04	8'26,127
4	501,18	8'27,355	14	500,07	8'26,238
5	500,04	8'26,076	15	500,09	8'26,183
6	500,61	8'26,875	16	500,12	8'26,183
7	500,22	8'26,289	17	500,14	8'26,253
8	500,12	8'26,175	18	500,05	8'26,196
9	500,02	8'26,127	19	500,08	8'26,156
10	500,08	8'26,146	20	500,14	8'26,178

Pada pengujian untuk waktu 500 detik didapat nilai dari sistem pewaktu yang bervariasi tetapi tetap mendekati nilai 500 detik. Begitu juga dengan nilai dari stopwatch mendekati nilai 8 menit 26 detik. Nilai tertinggi dari sistem pewaktu yaitu 500,18 detik, dan nilai tertinggi stopwatch yaitu 8'27,355. Nilai terendah dari sistem pewaktu yaitu 499,87 detik, dan nilai terendah stopwatch yaitu 8'25,778. Bila nilai

waktu dari sistem pewaktu dirata-ratakan maka akan didapat 500,149 detik. Dan nilai rata-rata dari stopwatch 506,238 detik. Untuk pengujian pada waktu 1000 detik didapat data sebagai berikut :

Tabel 4.3. Pengujian untuk 1000 detik

No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)	No	Sistem Pewaktu(detik)	Stopwatch (menit'detik)
1	1000,09	16'48,129	11	1000,15	16'48,254
2	1000,88	16'48,983	12	1000,24	16'48,225
3	1000,01	16'47,972	13	1000,27	16'48,304
4	999,97	16'48,000	14	1000,14	16'48,213
5	1000,08	16'48,167	15	1000,05	16'48,087
6	1000,15	16'48,215	16	1000,16	16'8,236
7	1000,12	16'48,203	17	1000,07	16'48,110
8	1000,26	16'48,337	18	1000,09	16'48,127
9	1000,07	16'48,147	19	1000,12	16'48,184
10	1000,18	16'48,285	20	1000,05	16'48,092

Pada pengujian untuk waktu 1000 detik didapat nilai dari sistem pewaktu yang bervariasi tetapi tetap mendekati nilai 1000 detik. Begitu juga dengan nilai dari stopwatch mendekati nilai 16 menit 48 detik. Nilai tertinggi dari sistem pewaktu yaitu 1000,88 detik, dan nilai tertinggi stopwatch yaitu 16'48,983. Nilai terendah dari sistem pewaktu yaitu 999,97 detik, dan nilai terendah stopwatch yaitu 16'47,972. Bila nilai waktu dari sistem pewaktu dirata-ratakan maka akan didapat 1000,158 detik. Dan nilai rata-rata dari stopwatch 1008,226 detik. Untuk pengujian pada waktu 9999 detik didapat data sebagai berikut :

Tabel 4.4. Pengujian untuk 9999 detik

No	Sistem Pewaktu(detik)	Stopwatch (jam'menit'detik)
1	9999,08	2'46'54,861
2	9999,15	2'46'57,214
3	9999,12	2'46'55,357

Pada pengujian untuk waktu 9999 detik ini hanya untuk menunjukkan jika sistem pewaktu dapat mencapai nilai maksimalnya yaitu 9999, dan juga untuk mengetahui kesesuaian detik tersebut dengan yang sebenarnya. Dari data diatas didapat nilai rata dari sistem pewaktu adalah 9999,117 detik, dan nilai rata-rata dari stopwatch adalah 10015,811 detik. Dari pengujian sistem pewaktu diatas dalam beberapa waktu maka didapat kesalahan relatif sebagai berikut :

$$\text{Kesalahan Relatif untuk 100 detik} = \frac{100,159 - 100,101}{100} \times 100\% = 0,058\%$$

$$\text{Kesalahan Relatif untuk 500 detik} = \frac{506,238 - 500,149}{500} \times 100\% = 1,218\%$$

$$\text{Kesalahan Relatif untuk 1000 detik} = \frac{1008,226 - 1000,158}{1000} \times 100\% = 0,807\%$$

$$\text{Kesalahan Relatif untuk 9999 detik} = \frac{10015,811 - 9999,117}{9999} \times 100\% = 0,167\%$$

Dapat dilihat dari persentase kesalahan relatif yang didapatkan menunjukkan persentase yang sangat kecil yaitu 0,058% untuk 100 detik, 1,218% untuk 500 detik, 0,807% untuk 1000 detik, dan 0,167% untuk 9999 detik. Sehingga dapat dinyatakan jika sistem pewaktu yang dibuat sesuai dengan real time.

4.2. Pengujian dan Analisa Data Mode

Pada pengujian kali ini dilakukan untuk mode yang telah dibuat. Tujuannya untuk mencoba apakah mode yang telah dibuat dapat berjalan dengan baik setelah dihubungkan dengan sensor yang ada. Dimana fungsi dari sensor ini untuk mengaktifkan interrupt yang nantinya berfungsi untuk menjalankan timer ataupun menghentikan timer. Sesuai dengan pilihan mode yang digunakan. Untuk saat ini hanya terdapat 3 mode saja.

Mode pertama, untuk mode ini menggunakan 2 buah sensor. Dimana 1 buah sensor untuk mengaktifkan int0 yang akan menjalankan timer, dan 1 buah sensor lagi

untuk mengaktifkan untuk mengaktifkan int1 yang akan menghentikan timer. Data yang didapat sebagai berikut :

Tabel 4.5. Pengujian Mode 1

No.	Sensor 1	Sensor 2
1	timer on	timer off
2	timer on	timer off
3	timer on	timer off
4	timer on	No Respon
5	timer on	timer off
6	timer on	timer off
7	timer on	No Respon
8	No Respon	
9	timer on	timer off
10	timer on	timer off

Mode kedua, pada mode ini yang digunakan hanya 1 buah sensor. Dimana 1 buah sensor dihubungkan dengan int0, sensor tersebut berfungsi untuk menjalankan timer dan juga untuk menghentikan timer. Caranya yaitu ketika int0 diinterrupt untuk pertama kali maka timer mulai berjalan, dan ketika int0 diinterrupt untuk kedua kalinya maka timer akan berhenti. Data yang didapat sebagai berikut :

Tabel 4.6. Pengujian Mode 2

No.	Interrupt ke 1	Interrupt ke 2
1	timer on	timer off
2	timer on	timer off
3	timer on	No Respon
4	timer on	timer off
5	timer on	timer off
6	timer on	timer off
7	No Respon	
8	timer on	timer off
9	No Respon	
10	timer on	timer off

Mode ketiga, untuk mode ini sama dengan mode 2 hanya menggunakan 1 buah sensor. Tetapi pada mode ketiga ini satu sensor tersebut dihubungkan ke int0, dan hanya berfungsi untuk menghentikan timer. Sedangkan timer mulai berjalan pada saat tombol start ditekan. Data yang didapat sebagai berikut :

Tabel 4.7. Pengujian Mode 3

No.	Start ditekan	Sensor 1
1	timer on	timer off
2	timer on	timer off
3	timer on	timer off
4	timer on	timer off
5	timer on	No Respon
6	timer on	timer off
7	timer on	timer off
8	timer on	No Respon
9	timer on	timer off
10	timer on	timer off

Dari data ketiga data diatas terlihat jika dalam 10 kali uji coba, tidak ke 10 uji coba tersebut dapat berjalan dengan baik. Masih ada beberapa kali kegagalan dalam 10 kali percobaan tersebut. Sehingga dapat dicari nilai persentase keberhasilan dari 10 kali uji coba.

$$\% \text{ Keberhasilan} = \frac{\text{benar} \times 100\%}{10}$$

$$\% \text{ Keberhasilan mode 1} = \frac{7 \times 100\%}{10} = 70\%$$

$$\% \text{ Keberhasilan mode 2} = \frac{7 \times 100\%}{10} = 70\%$$

$$\% \text{ Keberhasilan mode 3} = \frac{8 \times 100\%}{10} = 80\%$$

Persentase keberhasilan yang didapat oleh ketiga mode tersebut diatas 60%, sehingga sistem yang ada dapat dikatakan mampu berjalan dengan baik. Tetapi yang

sebaiknya nilai persentase keberhasilan harus mendekati 100% agar sistem dapat bekerja dengan optimal. Dari persentase keberhasilan yang didapat oleh ketiga mode tersebut terlihat jika sistem penginterrupt yang dilakukan oleh sensor masih belum dapat bekerja dengan optimal. Hal ini mungkin dikarenakan oleh sensor goyah sehingga kedudukan sensor tidak sejajar antara transceiver dan receiver, sensor tidak mendapatkan perubahan tegangan(voltase) sehingga interrupt tidak bekerja, atau perubahan tegangan yang tidak terlalu signifikan sehingga dapat dianggap tidak terjadi perubahan tegangan.



BAB 5

PENUTUP

5.1. Kesimpulan

Timer merupakan salah satu sistem pewaktuan yang dapat dibuat ciptakan dengan menggunakan nilai clock pada suatu *microcontroller*. Walaupun didalam sebuah mikro sudah terdapat sistem timer sendiri. Tetapi dengan menciptakannya sendiri, maka kita dapat membuat sistem pewaktuan yang sesuai dengan keinginan kita. Dari pembuatan programmable timer diatas dapat disimpulkan beberapa hal, antara lain :

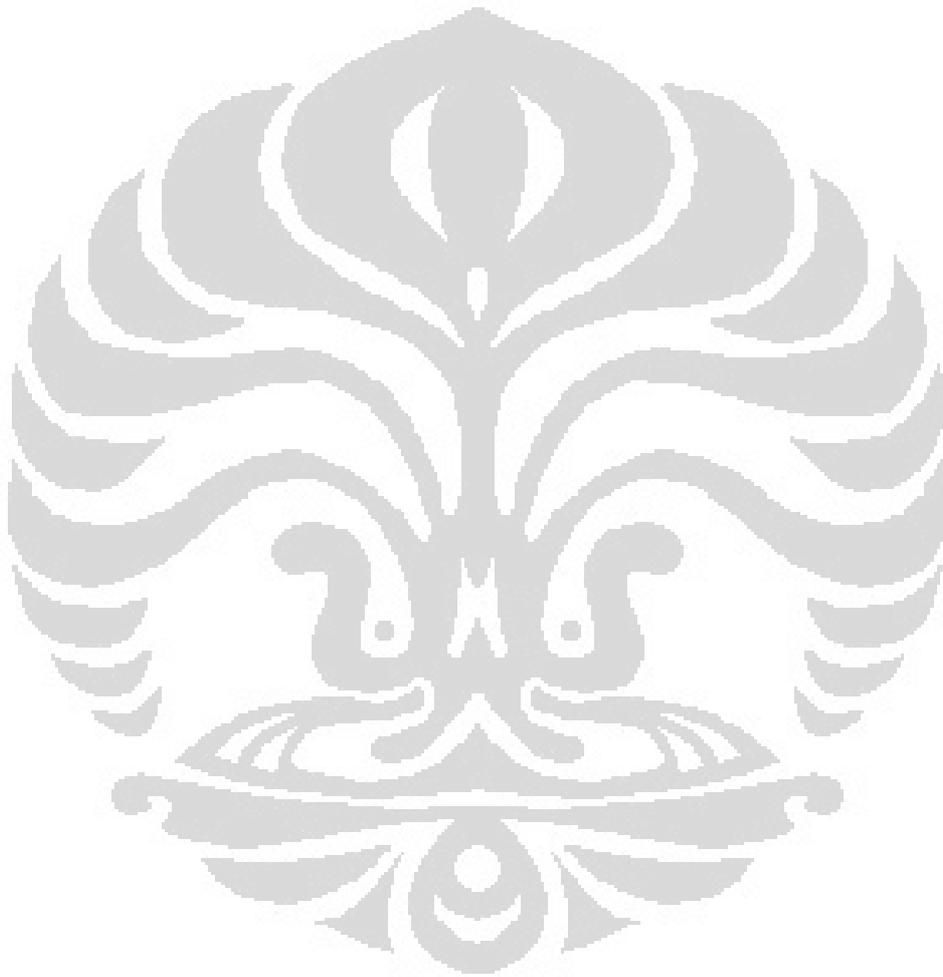
1. Dalam membuat sebuah timer dengan menggunakan timer overflow harus dipertimbangkan beberapa hal, yaitu nilai crystal yang digunakan, besar byte timer yang digunakan, besarnya nilai pewaktuan yang akan dibuat dan prescale dari timer
2. Interrupt pada ATmega 16 digunakan secara aktif low
3. Sensor yang digunakan untuk menghidupkan fungsi interrupt terkadang tidak dapat berfungsi sebagaimana mestinya
4. Data timer yang dihasilkan sudah cukup baik
5. Nilai dari sistem pewaktu sudah hampir mendekati nilai real time, dengan kesalahan relatif yang kecil

5.2. Saran

Penulis menyadari bahwa masih banyak kekurangan yang terdapat pada perangkat lunak sistem keamanan gedung yang telah dirancang. Oleh karena itu penulis menyarankan beberapa hal yang kiranya dapat lebih menyempurnakan perangkat lunak programmable timer yang telah dirancang, yaitu:

1. Merancang rangkaian program timer yang lebih baik lagi.
2. Dapat membuat sistem program tumbukan untuk 2 buah benda.
3. Merancang program sesederhana mungkin.
4. Tampilan timer yang lebih menarik.

5. Menambahkan mode agar lebih bervariasi
6. Dan mencoba untuk lebih kreatif dan inovatif dalam pengembangan yang lebih baik sehingga sempurna.



DAFTAR PUSTAKA

1. Budiharto Widodo, S. Si, M. Kom, *Interfacing Komputer dan Mikrokontroller*, Elex Media Komputindo, Jakarta, 2004.
2. Eugene Lister, *Mesin dan Rangkaian Listrik*, Erlangga, Jakarta, 1993.
3. Putra, Afianto Eko, *Belajar Mikrokontroller*, Gava Media, Jakarta, 2003.
4. Tokheim, Roger L., *Digital Electronics*, Prentice-Hall International, Inc.





LAMPIRAN

```
$regfile = "m16def.dat"  
$crystal = 11059200
```

```
Config Pind.7 = Input  
Config Pind.6 = Input  
Config Pind.2 = Input  
Config Pind.3 = Input  
Config Pind.0 = Output  
Config Pind.1 = Output  
Config Pind.0 = Output  
Config Pind.4 = Output  
Config Pinb.5 = Output  
Config Portc = Output  
Config Portb = Output  
Config Porta = Output  
Config Int1 = Falling  
Config Int0 = Falling  
Config Timer0 = Timer , Prescale = 8  
Config Timer1 = Pwm , Pwm = 10 , Compare A Pwm = Clear Down , Compare B  
Pwm = Clear Down , Prescale = 64
```

```
Set Portd.2  
Set Portd.3
```

```
Set Pind.6  
Set Pind.7
```

```
On Int1 Start_timer1  
On Int0 Start_timer0  
On Ovf0 Time_sampling_isr
```

```
Enable Interrupts  
Enable Int0  
Enable Int1  
Enable Ovf0
```

```
Dim Cacah As Word  
Dim Led As Byte  
Dim Periode As Byte  
Dim Punch As Byte  
Dim Satuan As Byte  
Dim Sisa_satuan As Word  
Dim Puluhan As Byte  
Dim Sisa_puluhan As Word  
Dim Ratusan As Byte
```

```

Dim Sisa_ratusan As Byte
Dim Ribuan As Byte
Dim Digit12 As Byte
Dim Digit34 As Byte
Dim Led_display As Byte
Dim Mode_timer As Byte
Dim Start_flag As Bit
Dim Mode_timer_bcd As Byte
Dim Mulai_timer As Byte
Dim Iseng As Byte
Dim Nilai_timer0 As Byte
Dim Nilai_timer1 As Byte

Mode_timer = 0
Iseng = 60

Do
If Pind.6 = 0 Then Mode_timer = Mode_timer + 1
Waitms 100
Mode_timer_bcd = Makebcd(mode_timer)
Portc = Mode_timer_bcd
Loop Until Pind.7 = 0
Start_flag = 0
Iseng = Makebcd(iseng)
Portc = Iseng
Porta = Iseng
Portb = 128

If Mode_timer = 3 Then 'MODE 3
    Periode = 0
    Led = 0
    Cacah = 0
    Start_flag = 1
End If

Program_utama:
Start Timer0
Start Timer1
Pwm1a = 512
Pwm1b = 768
Portb = 0
Punch = 0

Do
    If Start_flag = 1 Then

```

```

Led_display = Led + Punch
Satuan = Cacah Mod 10           'DIGIT PERTAMA
Sisa_satuan = Cacah - Satuan
Sisa_satuan = Sisa_satuan / 10
Puluhan = Sisa_satuan Mod 10   'DIGIT KEDUA
Sisa_puluhan = Sisa_satuan - Puluhan
Sisa_puluhan = Sisa_puluhan / 10
Ratusan = Sisa_puluhan Mod 10  'DIGIT KETIGA
Sisa_ratusan = Sisa_puluhan - Ratusan
Sisa_ratusan = Sisa_ratusan / 10
Ribuan = Sisa_ratusan Mod 10   'DIGIT KEEMPAT
Digit12 = 16 * Puluhan
Digit12 = Digit12 + Satuan
Digit34 = 16 * Ribuan
Digit34 = Digit34 + Ratusan
Portc = Digit12
Porta = Digit34
Portb = Led_display
End If
Loop

Start_timer0:

If Mode_timer = 1 Then         'MODE 3
  Periode = 0
  Led = 0
  Cacah = 0
  Start_flag = 1
End If

If Mode_timer = 2 Then         'MODE 2
  Enable Int1
  Incr Nilai_timer0
  If Nilai_timer0 = 1 Then
    Periode = 0
    Led = 0
    Cacah = 0
    Start_flag = 1
  End If
  If Nilai_timer0 = 2 Then Start_flag = 0
End If

If Mode_timer = 3 Then Start_flag = 0         'MODE 3

```

Return

Start_timer1:

```
If Mode_timer = 1 Then Start_flag = 0           'MODE 1
Return
```

Time_sampling_isr:

```
Periode = Periode + 1
```

```
If Periode = 54 Then
```

```
    Periode = 0
```

```
    Led = Led + 1
```

```
    If Led = 100 Then
```

```
        Led = 0
```

```
        Cacah = Cacah + 1
```

```
        If Cacah = 10000 Then Cacah = 0
```

```
    End If
```

```
End If
```

```
Return
```