

# MIKROKONTROLER DAN INTERFACE

**MIKROKONTROLER** adalah bidang ilmu keteknikan yang mempelajari tentang pengontrolan alat elektronika yang mengkombinasikan hardware (rangkaiian elektronika) dengan software (pemrograman).

**INTERFACE** adalah model pengaplikasian mikrokontroler dengan perangkat lain (Perangkat Antarmuka).

**Mata Kuliah MIKROKONTROLER DAN INTERFACE** adalah mata kuliah yang memberikan ilmu pengontrolan berbasis program yang dapat dirubah setiap saat untuk mengontrol bermacam-macam perangkat lewat berbagai macam media komunikasi antarmuka.

## BAHASAN MELIPUTI:

- Pengenalan dan Sejarah Mikrokontroler
- Hardware dan Software Mikrokontroler AVR dan Arduino
- Interface Data Digital
- Interface Dengan LCD
- Interface Input Analog (ADC)
- Interface Output PWM
- Interface Serial USART
- Interface Serial SPI
- Interface Serial I2C



BUKU AJAR  
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI MAKASSAR  
<http://www.unm.ac.id/>



MIKROKONTROLER DAN INTERFACE

ARDUINO



# MIKROKONTROLER DAN INTERFACE

Sutarsi Suhaeb, S.T., M.Pd.  
Yasser Abd Djawad, S.T., M.Sc., Ph.D.  
Dr. Hendra Jaya, S.Pd., M.T.  
Ridwansyah, S.T., M.T.  
Drs. Sabran, M.Pd.  
Ahmad Risal, A.Md.

**BUKU AJAR**  
**MIKROKONTROLER DAN INTERACE**

Sutarsi Suhaeb, S.T., M.Pd.  
Yasser Abd Djawad, S.T., M.Sc., Ph.D.  
Dr. Hendra Jaya, S.Pd., M.T.  
Ridwansyah, S.T., M.T.  
Drs. Sabran, M.Pd.  
Ahmad Risal, A.Md.



UNM

## MIKROKONTROLER DAN INTERFACE

Universitas Negeri Makassar  
Fakultas Teknik  
Pendidikan Teknik Elektronika

Penulis: Ahmad Risal

Desain Sampul: Ahmad Risal

Pembimbing:

1. Sutarsi Suhaeb, S.T., M.Pd.
2. Yasser Abd Djawad, S.T., M.Sc., Ph.D.

Penguji:

1. Dr. Hendra Jaya, S.Pd., M.T.
2. Ridwansyah, S.T., M.T.

Validator Konten/Materi: Drs. Sabran, M.Pd.

Validator Desain/Media: Dr. Muh. Ma'ruf Idris, S.T., M.T.

@Desember2017

# Kata Pengantar

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT, yang telah memberikan rahmat dan karuniaNya, sehingga Buku Ajar *Mikrokontroler dan Interface* ini dapat diselesaikan dengan baik. Pembahasan materi pada buku ajar ini dilakukan dengan cara memaparkan landasan teori elektronika dan instrumentasi digital khususnya tentang mikrokontroler.

Mikrokontroler adalah bidang ilmu keteknikan yang mempelajari tentang pengontrolan alat elektronika yang mengkombinasikan hardware (rangkai-an elektronika) dengan software (pemrograman). Interface adalah model pengaplikasian mikrokontroler dengan perangkat lain (Perangkat Antarmuka). Mata Kuliah Mikrokontroler dan Interface adalah mata kuliah yang memberikan ilmu pengontrolan berbasis program yang dapat dirubah setiap saat untuk mengontrol bermacam-macam perangkat lewat berbagai macam media komunikasi.

Isi buku ajar ini mencakup materi pokok mikrokontroler dan interface yang mencakup: Sejarah dan Pengenalan Mikrokontroler, Pemrograman Mikrokontroler AVR dan Mikrokontroler Arduino, Interface Data Digital, Interface Dengan LCD, Interface Input Analog (ADC), Interface Output PWM, Interface Serial USART, Interface Serial SPI, Interface Serial I2C.

Pada kesempatan ini penyusun menyampaikan terima kasih kepada semua pihak yang telah membantu penyusun dalam menyelesaikan buku ajar ini. Mudah-mudahan buku ajar ini dapat memberikan sedikit manfaat bagi para mahasiswa pada umumnya yang mengambil mata kuliah mikrokontroler dan interface.

*Makkassar, November 2017*  
*Penulis,.*



# Daftar Isi

<b>Kata Pengantar</b>	<b>iii</b>
<b>Daftar Isi</b>	<b>iv</b>
<b>1 Pengenalan &amp; Sejarah Mikrokontroler</b>	<b>1</b>
Tujuan Pembelajaran . . . . .	1
1.1 Pengenalan Mikrokontroller . . . . .	2
1.2 Sejarah Mikrokontroler . . . . .	3
1.3 Mikroprosesor, Mikro-Komputer dan Mikrokontroler . . . . .	5
1.4 Arsitektur Mikrokontroler . . . . .	6
1.5 Jenis-Jenis Mikrokontroler . . . . .	8
1.6 Produsen Mikrokontroler . . . . .	9
1.6.1 Motorola . . . . .	9
1.6.2 Mitsubishi . . . . .	9
1.6.3 NEC . . . . .	10
1.6.4 Hitachi . . . . .	10
1.6.5 Philips . . . . .	10
1.6.6 Intel . . . . .	11
1.6.7 Toshiba . . . . .	11
1.6.8 National Semiconductor . . . . .	12
1.6.9 Zilog . . . . .	12
1.6.10 Texas Instruments . . . . .	13
1.6.11 Siemens . . . . .	13
1.6.12 Sharp . . . . .	13
Rangkuman . . . . .	14
Soal Formatif . . . . .	14

<b>2</b>	<b>Hardware dan Software Mikrokontroler AVR dan Arduino</b>	<b>15</b>
	Tujuan Pembelajaran . . . . .	15
2.1	Mikrokontroler AVR . . . . .	16
	2.1.1 Arsitektur . . . . .	16
	2.1.2 Memori . . . . .	17
2.2	Arduino Board . . . . .	18
2.3	Algoritma Pemrograman . . . . .	22
	2.3.1 Definisi Algoritma . . . . .	23
	2.3.2 Program . . . . .	23
	2.3.3 Perbedaan interpreter dan compiler . . . . .	24
	2.3.4 Penulisan Algoritma . . . . .	25
2.4	Bahasa Pemrograman . . . . .	27
	2.4.1 Bahasa Mesin . . . . .	27
	2.4.2 Bahasa Assembly . . . . .	27
	2.4.3 Bahasa Tingkat Tinggi . . . . .	28
2.5	Bahasa Pemrograman C . . . . .	29
	2.5.1 Penulisan Bahasa C AVR . . . . .	29
	2.5.2 Penulisan Bahasa C Arduino . . . . .	31
	2.5.3 Tipe Data . . . . .	32
	2.5.4 Operator . . . . .	32
	Rangkuman . . . . .	34
	Soal Formatif . . . . .	34
<b>3</b>	<b>Interface Data Digital</b>	<b>35</b>
	Tujuan Pembelajaran . . . . .	35
3.1	Data Digital . . . . .	36
3.2	Sistem Digital dan Sistem Analog . . . . .	37
3.3	Aplikasi Interface Data Digital . . . . .	38
	3.3.1 Led Emiting Diode . . . . .	38
	3.3.2 Seven Segment . . . . .	39
	3.3.3 Led Matrix . . . . .	40
	3.3.4 Motor . . . . .	42
	3.3.5 Buzzer . . . . .	43
	3.3.6 Push Button . . . . .	43
3.4	Data Digital Mikrokontroler AVR . . . . .	45
	3.4.1 Aplikasi Kontrol Output LED . . . . .	49
	3.4.2 Aplikasi Kontrol Output Seven Segement . . . . .	50
	3.4.3 Aplikasi Kontrol Output Motor DC . . . . .	52
	3.4.4 Aplikasi Input dengan Push Button . . . . .	55
3.5	Data Digital Arduino . . . . .	56

3.5.1	Aplikasi Kontrol Output LED . . . . .	57
3.5.2	Aplikasi Kontrol Output Seven Segment . . . . .	58
3.5.3	Aplikasi Kontrol Output Motor DC . . . . .	60
3.5.4	Aplikasi Input dengan Push Button . . . . .	63
	Rangkuman . . . . .	64
	Soal Formatif . . . . .	64
	Tugas Latihan . . . . .	64
<b>4</b>	<b>Interface Dengan LCD</b>	<b>65</b>
	Tujuan Pembelajaran . . . . .	65
4.1	Pengenalan LCD . . . . .	66
4.2	Fungsi Pin-Pin Modul LCD . . . . .	67
4.3	Interface LCD di Mikrokontroler AVR . . . . .	68
4.3.1	Rangkaian Hardware . . . . .	68
4.3.2	Pemrograman . . . . .	69
4.4	Interface LCD di Arduino . . . . .	71
4.4.1	Rangkaian Hardware . . . . .	71
4.4.2	Pemrograman Software . . . . .	72
	Rangkuman . . . . .	74
	Soal Formatif . . . . .	74
	Tugas Latihan . . . . .	74
<b>5</b>	<b>Interface Input Analog (ADC)</b>	<b>77</b>
	Tujuan Pembelajaran . . . . .	77
5.1	Data Analog . . . . .	78
5.2	Sinyal Analog . . . . .	79
5.3	Analog to Digital Converter . . . . .	81
5.3.1	Pengertian ADC . . . . .	81
5.3.2	Prinsip Kerja ADC . . . . .	82
5.3.3	Komparator ADC . . . . .	82
5.3.4	Jenis-Jenis ADC . . . . .	83
5.4	Komponen Antar Muka ADC . . . . .	83
5.4.1	Sensor Cahaya . . . . .	84
5.4.2	Sensor Suhu . . . . .	85
5.4.3	Sensor Tekanan . . . . .	86
5.4.4	Sensor Kecepatan . . . . .	87
5.4.5	Sensor Magnet . . . . .	87
5.4.6	Sensor Penyandi . . . . .	87
5.4.7	Flow Meter Sensor . . . . .	88
5.4.8	Flame Sensor . . . . .	88

5.5	Interface ADC di Mikrokontroler AVR . . . . .	88
5.5.1	Rangkaian . . . . .	89
5.5.2	Pemrograman . . . . .	90
5.6	Interface ADC di Arduino . . . . .	92
5.6.1	Rangkaian . . . . .	92
5.6.2	pemrograman . . . . .	92
	Rangkuman . . . . .	94
	Soal Formatif . . . . .	94
	Tugas Latihan . . . . .	95
<b>6</b>	<b>Interface Output PWM</b>	<b>97</b>
	Tujuan Pembelajaran . . . . .	97
6.1	Timer/Counter . . . . .	98
6.1.1	Timer / Counter 8 Bit . . . . .	98
6.1.2	Timer / Counter 16 Bit . . . . .	100
6.2	PWM (Pulse Width Modulation) . . . . .	101
6.3	Komponen dengan kontrol output PWM . . . . .	102
6.3.1	LED . . . . .	103
6.3.2	Motor DC . . . . .	104
6.3.3	Motor Servo . . . . .	105
6.4	Interface PWM di mikrokontroler AVR . . . . .	107
6.4.1	Interface PWM menggunakan Motor DC . . . . .	107
6.4.2	Interface PWM menggunakan Motor Servo . . . . .	109
6.5	Interface PWM di Arduino . . . . .	111
6.5.1	Interface PWM dengan Motor DC . . . . .	112
6.5.2	Interface PWM dengan Motor Servo . . . . .	113
	Rangkuman . . . . .	115
	Soal Formatif . . . . .	115
	Tugas Latihan . . . . .	115
<b>7</b>	<b>Interface Serial USART</b>	<b>117</b>
	Tujuan Pembelajaran . . . . .	117
7.1	Komunikasi Serial . . . . .	118
7.2	Antarmuka Serial dan USART . . . . .	118
7.3	Register USART . . . . .	120
7.3.1	Register UCSRA . . . . .	121
7.3.2	Register UCSRB . . . . .	121
7.3.3	Register UCSRC . . . . .	122
7.4	Fitur pada Serial USART . . . . .	122
7.5	Komponen dengan antarmuka USART . . . . .	123

7.5.1	Bluetooth . . . . .	123
7.5.2	Modul GSM . . . . .	124
7.6	Interface Serial USART pada Mikrokontroler AVR . . . . .	124
7.6.1	Rangkaian USART . . . . .	125
7.6.2	Pemrograman USART . . . . .	125
7.7	Interface Serial pada Arduino . . . . .	127
7.7.1	Rangkaian USART . . . . .	128
7.7.2	Pemrograman USART . . . . .	128
	Rangkuman . . . . .	130
	Soal Formatif . . . . .	130
	Tugas Latihan . . . . .	130
<b>8</b>	<b>Interface Serial SPI</b>	<b>133</b>
	Tujuan Pembelajaran . . . . .	133
8.1	Antarmuka Serial dan SPI . . . . .	134
8.2	Komponen dengan antarmuka SPI . . . . .	137
8.2.1	Downloader ASP/ISP . . . . .	137
8.2.2	Joystik PS2 . . . . .	138
8.3	Interface Serial pada Mikrokontroler AVR . . . . .	140
8.3.1	Rangkaian ISP . . . . .	140
8.3.2	Pemrograman SPI . . . . .	140
8.4	Interface Serial pada Arduino . . . . .	146
8.4.1	Rangkaian SPI pada JoyStick . . . . .	146
8.4.2	Pemrograman SPI pada JoyStick . . . . .	147
	Rangkuman . . . . .	151
	Soal Formatif . . . . .	151
<b>9</b>	<b>Interface Serial I2C</b>	<b>153</b>
	Tujuan Pembelajaran . . . . .	153
9.1	Pengenalan I2C . . . . .	154
9.2	RTC sebagai Interface I2C . . . . .	155
9.3	Interface I2C pada Mikrokontroler AVR . . . . .	157
9.3.1	Rangkaian . . . . .	157
9.3.2	Pemrograman . . . . .	158
9.4	Interface I2C pada Arduino . . . . .	160
9.4.1	Rangkaian . . . . .	160
9.4.2	Pemrograman . . . . .	160
	Rangkuman . . . . .	163
	Soal Formatif . . . . .	163
	Tugas Latihan . . . . .	164

<b>Daftar Pustaka</b>	<b>165</b>
<b>Kunci Jawaban Soal Formatif</b>	<b>166</b>
<b>Glosarium</b>	<b>175</b>
<b>Kumpulan Data Seet Mikrokontroler</b>	<b>177</b>

## Bab 1

# Pengenalan & Sejarah Mikrokontroler

### Tujuan Pembelajaran

---

1. Peserta Didik mampu menjelaskan sejarah dan perkembangan mikrokontroler.
2. Peserta Didik mampu membedakan mikroprocessor, mikrokontroler dan mikro-komputer.
3. Peserta Didik mampu mengklasifikasikan arsitektur mikrokontroler.
4. Peserta Didik mampu menguraikan jenis-jenis mikrokontroler.

## 1.1 Pengenalan Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil (*"special purpose computers"*) di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan paralel, Port input/output, ADC. Mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program.



Gambar 1.1: IC Mikrokontroler

Mikrokontroler ini adalah ilmu terapan yang pengaplikasiannya dapat kita temui di kehidupan sehari-hari seperti jam digital, televisi, sistem keamanan rumah, dll. Mikrokontroler juga sangat banyak digunakan dalam penelitian dan pengembangan yang dilakukan oleh peneliti, dosen, guru, bahkan sekarang banyak mahasiswa yang mengangkat judul tesis/sekripsi/tugas akhir dengan berbasiskan mikrokontroler.

Mikrokontroler adalah komponen yang sangat umum dalam sistem elektronika modern. Penggunaannya sangat luas, dalam kehidupan kita sehari-hari baik di rumah, kantor, rumah sakit, bank, sekolah, industri, dll. Mikrokontroler digunakan dalam sejumlah besar sistem elektronika seperti : sistem manajemen mesin mobil, keyboard komputer, alat ukur elektronik ( multimeter digital, synthesizer frekuensi, dan osiloskop ), televisi, radio, telepon digital, mobile phone, microwave oven, printer, scanner, kulkas, pendingin ruangan, CD/DVD player, kamera, mesin cuci, PLC (programmable logic controller ), robot, sistem otomasi, sistem akuisisi data, sistem keamanan, sistem EDC ( Electronic Data Capture ), mesin ATM, modem, router, dll.

Mikrokontroler dapat kita gunakan untuk berbagai aplikasi misalnya untuk pengendalian, otomasi industri, akuisisi data, telekomunikasi, dan lain-lain. Keuntungan menggunakan mikrokontroler yaitu harganya murah, dapat diprogram berulang kali, dan dapat kita program sesuai dengan

keinginan kita. Saat ini keluarga mikrokontroler yang ada dipasaran yaitu intel 8048 dan 8051 ( MCS51 ), Motorola 68HC11, microchip PI, hitachi H8, dan atmel AVR.

## 1.2 Sejarah Mikrokontroler

Sejarah mikrokontroler tidak terlepas dari sejarah mikroprosesor dan komputer. Diawali dengan ditemukannya mikroprosesor, kemudian ditemukan komputer, setelah itu ditemukan mikrokontroler. Berikut ini sejarah mikrokontroler:

1. Tahun 1617, John Napier menemukan sistem untuk melakukan perkalian dan pembagian berdasarkan logaritma.
2. Tahun 1694, Gottfried Wilhelm Leibniz membuat mesin mekanik yang dapat melakukan operasi  $+$ ,  $-$ ,  $*$ ,  $/$  dan akar kuadrat.
3. Tahun 1835, Charles Babbage mengusulkan komputer digital (Digital Computer) pertama di dunia menggunakan punched card untuk data dan instruksi, serta program control (looping and branching) dengan unit aritmatika dan unik penyimpanan.



Gambar 1.2: IC Mikrokontroler

4. Tahun 1850, George Boole mengembangkan symbolic logic termasuk operasi binary (AND, OR, dll).
5. Tahun 1946, Von Neumann menyarankan bahwa instruksi menjadi kode numerik yang disimpan pada memori. Komputer dan semua mikrokontroler didasarkan pada komputer Von Neumann.

6. Tahun 1948, ditemukannya transistor, dengan dikembangkannya konsep software, pada tahun 1948 mulai adanya perkembangan hardware penting seperti transistor.
7. Tahun 1959, pertama kali dibuatnya IC ( Integrated Circuit).
8. Tahun 1971, intel membuat mikroprosesor intel 4004. mikroprosesor ini merupakan mikroprosesor pertama yang dikembangkan oleh intel ( Integrated Electronics ). Mikroprosesor ini terdiri dari 2250 transistor. Intel 4004 merupakan mikroprosesor 4 bit. Kemudian pada tahun 1974, intel membuat mikroprosesor generasi kedua ( intel 8008), intel 8008 merupakan mikroprosesor 8 bit. Semakin besar ukuran bit berarti mikroprosesor dapat memproses lebih banyak data. IC mikroprosesor intel 4004 dan intel 8008 ini dikemas dalam bentuk DIP ( Dual Inline Package ) seperti pada gambar dibawah ini.
9. Tahun 1972, Mikrokontroler yang dibuat adalah TMS 1000. TMS 1000 merupakan mikrokontroler 4-bit buatan Texas Instrument (TI). Mikrokontroler TMS 1000 dibuat oleh Gary Boone dari Texas Instrument. Boone merancang IC yang dapat menampung hampir semua komponen yang membentuk kalkulator, hanya layar dan keypad yang tidak dimasukkan. TI menawarkan mikrokontroler ini untuk dijual kepada industri elektronik pada tahun 1983, sebanyak 100 juta IC mikrokontroler TMS 1000 telah dijual.
10. Tahun 1974, beberapa pabrikan IC menawarkan mikroprosesor dan pengendali menggunakan mikroprosesor. Mikroprosesor yang ditawarkan pada saat itu yaitu Intel 8080, 8085, Motorola 6800, Signetics 6502, Zilog Z80, Texas Instrument 9900 (16 bit)
11. Tahun 1975, mikrokontroler PIC dikembangkan dan dibuat pertama kali di Universitas Harvard. PIC mulai diperkenalkan kepada publik oleh Microchip pada tahun 1985. PIC merupakan kependekan dari Peripheral Interface Controller atau bisa juga kependekan dari Programmable Intelligent Computer.
12. Tahun 1976, dibuat Intel 8048, yang merupakan mikrokontroler intel pertama.
13. Tahun 1978, mikroprosesor 16 bit menjadi lebih umum digunakan yaitu Intel 8086, Motorola 68000 dan Zilog Z8000. Sejak saat itu pabrikan mikroprosesor terus mengembangkan mikroprosesor dengan berba-

gai keistimewaan dan arsitektur. Mikroprosesor yang dikembangkan termasuk mikroprosesor 32 bit seperti Intel Pentium, Motorola DragonBall, dan beberapa mikrokontroler yang menggunakan ARM (Advanced RISC Machine Ltd) core. ARM hanya menjual desain arsitektur mikrokontroler/mikroprosesor. Saat ini sedang dipromosikan penggunaan mikrokontroler 32 bit yang berbasis prosesor ARM dari keluarga seri Cortex M (ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M3, ARM Cortex-M4, ARM Cortex-M7, ARM Cortex-R4, dan ARM Cortex A5). Perusahaan yang menggunakan lisensi ARM prosesor meliputi : Advanced Micro Devices, Inc., Broadcom, Samsung, Toshiba, Alcatel-Lucent, Apple Inc, Atmel, Intel, LG, Nuvoton, STMicroelectronics, Texas Instruments, Infineon, dan masih banyak lagi yang lainnya.

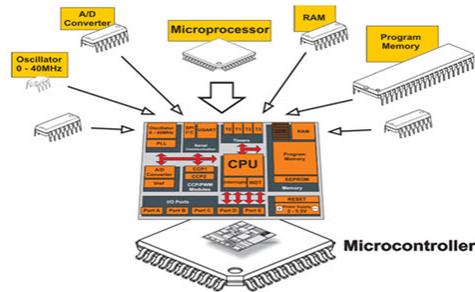
14. Tahun 1980, Intel 8051 atau lebih dikenal dengan keluarga mikrokontroler yang paling populer. Vendor lain yang mengadopsi mikrokontroler Intel 8051 yaitu : Philips, Siemens, Atmel ATMEL juga membuat Mikrokontroler MCS 51 yaitu mikrokontroler Atmel seri AT89xxx, misalnya : AT89S51 dan AT89S52.
15. Tahun 1996, Atmel AVR adalah salah satu keluarga mikrokontroler pertama yang menggunakan on-chip flash memory untuk penyimpanan program.

### 1.3 Mikroprosesor, Mikro-Komputer dan Mikrokontroler

Mikroprosesor adalah *Central Processing Unit (CPU)* didalam single chip. Komponen CPU : *Arithmetic and Logic Unit (ALU)*, *instruction decoder*, *register*, *bus control circuit*, dll. Mikro-Komputer adalah mikroprosesor yang dihubungkan dengan rangkaian pendukung.

komponen I/O dan memori (program & data) ditempatkan bersama untuk membentuk komputer kecil khususnya untuk akuisisi data dan aplikasi kontrol.

Jika komponen yang menyusun sebuah mikro-komputer diletakkan bersama didalam single chip silicon maka disebut mikrokontroler berisi CPU, memori, timer, port serial dan paralel, port input / output, ADC.



Gambar 1.3: Perbedaan Mikroprocessor dan Mikrokontroler

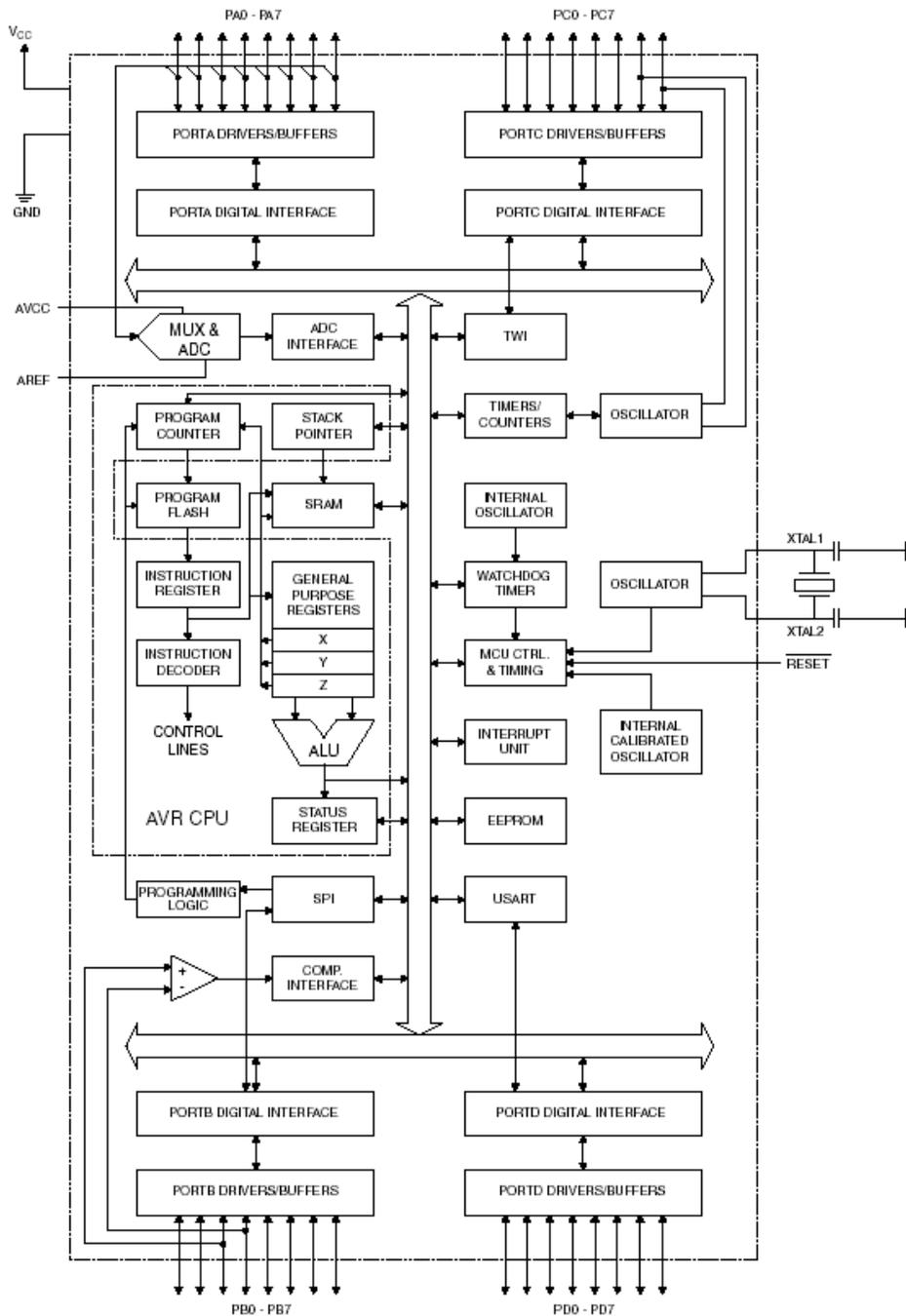
## 1.4 Arsitektur Mikrokontroler

Berdasarkan arsitekturnya, mikrokontroler dibagi menjadi : CISC dan RISC

- CISC (*Complex Instruction Set Computer*) Mikroprosesor CISC merupakan jenis mikrokontroler yang memiliki jumlah instruksi yang kompleks dan lengkap. Contoh: Mikrokontroler CISC: Intel 80C51 (MCS51) dan Motorola 68HC11 mengikuti arsitektur CISC.
- RISC (*Reduce Instruction Set Computer*) Mikroprosesor RISC merupakan jenis mikrokontroler yang memiliki jumlah instruksi yang terbatas dan sedikit. Pada arsitektur RISC jumlah instruksi lebih sedikit, tetapi memiliki banyak register dibandingkan dengan CISC. Selain itu pada arsitektur RISC kebanyakan instruksi dieksekusi hanya dalam satu clock cycle dan mode addressing memory yang sederhana. Contoh mikrokontroler RISC: ATMELAVR, Microchip PIC2/16CXX dan National Semiconductor COP8.

Program assembly dengan prosesor RISC menjadi lebih kompleks dibandingkan dengan program assembly prosesor CISC. Hal ini disebabkan hampir semua instruksi prosesor RISC adalah instruksi dasar, instruksi-instruksi ini umumnya hanya memerlukan 1 siklus mesin untuk menjalankannya. Sebagai contoh misalnya karena tidak ada instruksi untuk perkalian pada arsitektur RISC sehingga harus dibuat program perkalian dengan menggunakan instruksi-instruksi dasar seperti instruksi penjumlahan, dan lain-lain. Namun pada arsitektur RISC tidak diperlukan hardware yang kompleks, prosesor yang tidak rumit akan cepat dan andal.

Untuk merealisasikan instruksi dasar yang jumlahnya tidak banyak ini, mikroprosesor RISC tidak memerlukan gerbang logik yang banyak. Karena



Gambar 1.4: Arsitektur ATmega8535

itu dimensi IC dan konsumsi daya prosesor RISC umumnya lebih kecil dibanding prosesor CISC. Akan tetapi, program assembly pada prosesor CSIC menjadi lebih sederhana karena sudah ada instruksi yang kompleks. Untuk membuat instruksi yang kompleks seperti instruksi perkalian, pembagian, dan instruksi lain yang rumit pada prosesor CSIC, diperlukan hardware yang kompleks juga. Dibutuhkan ribuan gerbang logik (*logic gates*) transistor untuk membuat prosesor CSIC. Instruksi yang kompleks juga membutuhkan jumlah siklus mesin (*machine cycle*) yang lebih panjang untuk dapat menyelesaikan eksekusinya.

Salah satu mikrokontroler yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroler RISC (reduce instruction set compute) 8 bit berdasarkan arsitektur Harvard, yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced versatile RISC* atau *Alf and Vegards Risc* processor yang berasal dari nama dua mahasiswa Norwegian institute of technology (NTH), yaitu Alf-Egil Bogen dan Vegard Wollan.

AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan mikrokontroler AVR yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 siklus clock, lebih cepat dibandingkan dengan mikrokontroler MCS51 yang memiliki arsitektur CISC (complex instruction set computer) di mana mikrokontroler MCS51 membutuhkan 12 siklus clock untuk mengeksekusi 1 instruksi. Selain itu, mikrokontroler AVR memiliki fitur yang lengkap (ADC internal, EEPROM internal, Timer/counter, Watchdog Timer, PWM, Port I/O, Komunikasi serial, komparator, I2C, dll.), sehingga dengan fasilitas yang lengkap ini, programmer dan desainer dapat menggunakan untuk berbagai aplikasi sistem elektronika seperti robot, otomatis instruksi, peralatan telekomunikasi, dan berbagai keperluan lain. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi 5 kelompok yaitu keluarga Attiny, AT90SXX, Atmega, AVRXMega, dan AVR32 UC3.

## 1.5 Jenis-Jenis Mikrokontroler

1. Mikrokontroler TinyAVR (ATTiny) adalah mikrokontroler 8 bit. ATTiny merupakan mikrokontroler AVR kecil dan memiliki peripheral yang terbatas.
2. Mikrokontroler AT90S adalah mikrokontroler 8 bit jenis lama, merupakan mikrokontroler AVR klasik.

3. Mikrokontroler Atmega adalah mikrokontroler 8 bit. Atmega memiliki peripheral lebih banyak dibandingkan dengan seri ATTiny.
4. Mikrokontroler Xmega adalah mikrokontroler 8/16 bit. Xmega memiliki peripheral baru dan canggih dengan untuk kerja, sistem monitoring event dan DMA yang ditingkatkan,serta merupakan pengembangan keluarga AVR untuk pasar low power dan high performance. Dengan adanya fitur DMA(direct memory access) dapat mengurangi kemungkinan terjadinya kemacetan pada saat transfer data. Xmega mendukung kriptografi AES dan DES.
5. Mikrokontroler AVR32 adalah mikrokontroler 32 bit, mikrokontroler ini pertama kali dibuat oleh atmel pada tahun 2006. AVR32 menggunakan arsitektur RISC 32 bit, mikrokontroler ini ditujukan untuk bersaing dengan mikrokontroler yang berbasis prosesor ARM mikrokontroler AVR32 tidak memiliki EEPROM internal, sebagai pengganti EEPROM , AVR32 dapat menggunakan SD Card dan MMC.

## 1.6 Produsen Mikrokontroler

Dibawah ini adalah daftar produsen besar dari mikrokontroler.

### 1.6.1 Motorola

Motorola merupakan sebuah perusahaan multinasional yang menghasilkan berbagai macam produk teknologi. Didirikan pada tahun 1928 dengan nama Ludwig van Beethoven and Johannes Brahms. Bermarkas di Schaumburg, Illinois. Perusahaan ini menghasilkan berbagai macam produk-produk teknologi. Mempekerjakan 66.000 pekerja pada tahun 2006. Setelah kehilangan \$4,3 miliar dari tahun 2007 hingga 2009, perusahaan ini dibagi menjadi dua perusahaan publik yang independen, Motorola Mobility dan Motorola Solutions pada tanggal 4 Januari 2011.[6] Motorola Solutions umumnya dianggap menjadi penerus langsung ke Motorola, Inc, karena reorganisasi itu terstruktur dengan Motorola Mobility yang dipisahkan.

### 1.6.2 Mitsubishi

Mitsubishi Companies adalah sebuah perusahaan Jepang yang menaungi berbagai perusahaan yang berbagi merek dagang dan bagian perusahaan Mitsubishi. Perusahaan ini didirikan pada tahun 1870 sebagai perusahaan pelayaran oleh Yataro Iwasaki.

### **1.6.3 NEC**

NEC Corporation (dahulu merupakan singkatan dari Nippon Electric Company) adalah sebuah perusahaan teknologi informasi multi-nasional yang berkantor pusat di Minato, Tokyo, Jepang. NEC menyediakan informasi teknologi dan solusi jaringan kepada bisnis, penyedia jasa komunikasi, dan pemerintah. NEC merupakan pembuat dari Earth Simulator, salah satu superkomputer tercepat di dunia. NEC merupakan bagian dari Sumitomo Group.

Bisnis mereka terbagi dalam tiga segmen utama: Solusi TI, Solusi jaringan, dan peralatan elektronik. Solusi TI melayani solusi penghitungan kepada bisnis, pemerintah, dan individual. Solusi jaringan menawarkan sistem jaringan, peralatan genggam, penyiaran, dan sistem lainnya. Divisi peralatan elektroniknya termasuk semikonduktor, tampilan, dan lain-lain.

Tahun-tahun belakangan ini NEC merupakan salah satu perusahaan teratas dalam jumlah paten A.S. yang dikeluarkan, dengan rata-rata paten disetujui 1.764 per tahun.

### **1.6.4 Hitachi**

Hitachi, adalah perusahaan multinasional yang berkantor pusat di Marunouchi, Chiyoda, Tokyo, Jepang. Perusahaan induk bagi Hitachi Group. Hitachi berada di peringkat 371 dalam Forbes Global 2000 tahun 2007. Didirikan pada tahun 1910. Perusahaan ini mempekerjakan 384.444 pekerja pada tahun 2007.

### **1.6.5 Philips**

Koninklijke Philips Electronics N.V. (Royal Dutch Philips Electronics Ltd.), biasa dikenal sebagai Philips, ialah salah satu produsen elektronik konsumen terbesar di dunia. Pada 2004, penjualannya sekitar 30,3 juta dan mereka mempekerjakan 161.586 orang di lebih dari 60 negara. Philips diorganisasi dalam sejumlah divisi: Philips Consumer Electronics, Philips Semiconductors, Philips Lighting, Philips Medical Systems dan Philips Domestic Appliances and Personal Care.

Perusahaan ini didirikan pada 1891 oleh 2 bersaudara Gerard dan Anton Philips (1874-1951) di Eindhoven, Belanda. Produk pertamanya ialah bola lampu 'dan alat elektronik lainnya'. Pabrik pertamanya kini menjadi museum. Pada 1920-an, perusahaan ini mulai memproduksi produk lain, dan pada 1939 pisau cukur listrik pertamanya, Philishave, diperkenalkan. Philips memasarkan alat cukurnya di AS dengan nama Norelco. Philips

memperkenalkan tape compact audio cassette, yang dengan ramai sukses, melalui percobaan mereka mengatur standar untuk VCR, V2000, gagal di muka persaingan dari Betamax dan khususnya standar VHS.

Pada 9 Mei 1940, direktur Philips diberitahu mengenai penyerbuan Jerman ke Belanda pada 10 Mei. Mereka memutuskan meninggalkan negerinya dan lari ke Amerika Serikat. Mereka menerima banyak modal perusahaan dengannya. Beroperasi dari AS, mereka mengurus untuk menjalankan perusahaan sepanjang perang. Di saat yang sama, perusahaan itu sendiri pindah ke Antillen Belanda untuk menjaganya dari tangan Jerman. Setelah perang kembali ke Belanda, dengan markasnya di Eindhoven (dan pada 1997 keputusan dibuat untuk pindah ke Amsterdam perpindahan ini selesai pada 2001). Banyak fasilitas penelitian rahasia dikunci dan berhasil disembunyikan dari penyerang, yang memungkinkan perusahaan berlari cepat lagi setelah perang. Juga dipercaya bahwa Philips -sebelum dan selama perang- memasok banyak peralatan listrik kepada angkatan pendudukan Jerman, yang membuat beberapa orang berpikir bahwa perusahaan itu berkolaborasi dengan Nazi, seperti banyak perusahaan lain saat itu.

Bagaimanapun, tiada fakta yang mendukung bahwa Philips sendiri atau manajemennya pernah setuju dengan Nazi atau pahamnya. Secara jelas, ada sedikit Philips sudah bisa berbuat mencegah Jerman dari penyalahgunaan fasilitas produksinya dan memaksa pekerjanya melakukan kerja paksa selama pendudukan. Fasilitas produksi di Eindhoven merupakan satu-satunya sasaran Belanda yang dengan sengaja dibom angkatan sekutu selama perang.

### 1.6.6 Intel

Intel Corporation (NASDAQ: INTC; didirikan 1968) adalah sebuah perusahaan multinasional yang berpusat di Amerika Serikat dan terkenal dengan rancangan dan produksi mikroprosesor dan mengkhususkan dalam sirkuit terpadu. Intel juga membuat kartu jaringan, chipset papan induk, komponen, dan alat lainnya. Intel memiliki projek riset yang maju dalam seluruh aspek produksi semikonduktor, termasuk MEMS. Intel mengganti logo dan slogannya pada 2 Januari 2006. Slogan lamanya "Intel inside" diganti dengan "Leap ahead".

### 1.6.7 Toshiba

Toshiba adalah perusahaan pemroduksi elektronik teknologi tinggi yang bemarkas di Tokyo, Jepang. Toshiba adalah perusahaan elektronik terbesar

di dunia. Toshiba saat ini kebanyakan buatan RRC.

Semikonduktor buatan Toshiba termasuk ke dalam jajaran 20 Semikonduktor dengan Penjualan Terbesar. Tahun 2009, Toshiba merupakan perusahaan komputer terbesar kelima di dunia, di bawah Hewlett-Packard dari AS, Dell dari AS, Acer dari Taiwan, dan Apple dari AS.

### **1.6.8 National Semiconductor**

National Semiconductor (NYSE: NSM) adalah sebuah perusahaan pembuat semikonduktor Amerika Serikat yang berkecimpung di bidang peralatan analog dan subsistem yang berpusat di Santa Clara, California, Amerika Serikat. Produk-produk National Semiconductor meliputi sirkuit pengelolaan tenaga, driver tampilan, amplifier operasional dan audio, produk antarmuka komunikasi dan solusi konversi data. Pasar utama National meliputi peralatan nirkabel, layar tampilan dan berbagai pasar elektronik luas, termasuk di bidang medis, otomotif, industri, dan persyaratan uji dan penilaian.

### **1.6.9 Zilog**

Zilog, Inc, sebelumnya dikenal sebagai Zilog (yang merupakan singkatan dari "Z (kata terakhir dalam) logika terpadu"), adalah produsen Amerika dari mikrokontroler 8-bit dan 16-bit, dan yang paling terkenal dengan Intel 8080-kompatibel Z80 series.

Zilog didirikan di California pada tahun 1974 oleh Federico Faggin, yang meninggalkan Intel setelah bekerja pada 4004 dan 8080 mikroprosesor. Perusahaan menjadi anak perusahaan dari Exxon pada tahun 1980, namun manajemen dan karyawan membelinya kembali pada tahun 1989 dipimpin oleh Dr Edgar Sack.

Z80 (i) adalah sebuah implementasi penyempurnaan dari arsitektur Intel 8080, yang lebih cepat, lebih mampu, dan jauh lebih murah; bersama 6502 itu adalah salah satu prosesor 8-bit yang paling populer untuk mikrokomputer tujuan umum dan aplikasi lainnya. Itu digunakan dalam Nintendo Game Boy, Sinclair ZX80, ZX81, ZX Spectrum dan rumah komputer Amstrad BPK serta arsitektur MSX dan Microbee dan Tandy TRS-80 seri-di antara banyak lainnya. Lebih dari sekadar memicu perbaikan di bidang tunas komputasi rumah dan game, Z-80 juga memicu revolusi dalam musik elektronik, sebagai synthesizer pertama benar-benar diprogram polifonik (serta periferal mereka) sangat bergantung pada implementasi dari CPU ini.

### **1.6.10 Texas Instruments**

Texas Instruments Inc. (NYSE: TXN), dikenal luas sebagai TI, adalah sebuah perusahaan Amerika Serikat yang berpusat di Dallas, Texas, Amerika Serikat, yang mengembangkan dan memasarkan semikonduktor dan teknologi komputer. TI adalah produsen semikonduktor terbesar ketiga di dunia setelah Intel dan Samsung, pemasok cip untuk peralatan seluler terbesar kedua setelah Qualcomm, dan produsen semikonduktor prosesor sinyal digital (DSP) dan analog terbesar, di antara produk-produk semikonduktor lainnya. Setelah pengambilan alih National Semiconductor tahun 2011, perusahaan ini memiliki portofolio gabungan sebanyak 45.000 produk analog dan peralatan desain pelanggan. Pada tahun 2011, Texas Instruments menempati peringkat 175 dalam Fortune 500.

### **1.6.11 Siemens**

Siemens AG adalah salah satu perusahaan industri besar dunia. Kantor pusat internasionalnya terletak di Berlin dan Mnchen, Jerman. Siemens AG terdaftar di Bursa Saham Frankfurt dan juga di Bursa Saham New York sejak 12 Maret 2001.

Di seluruh dunia, Siemens dan anak perusahaannya mempekerjakan 461.000 orang (2005) di 190 negara dan melaporkan penjualan global sebesar 85 miliar dalam tahun fiskal 2005.

### **1.6.12 Sharp**

Sharp Corporation merupakan sebuah perusahaan multinasional yang menghasilkan berbagai macam produk elektronik. Didirikan pada tahun 1912. Bermarkas di Osaka, Jepang. Perusahaan ini mempekerjakan 56.000 pekerjanya pada tahun 2008.

## Rangkuman

---

1. Mikrokontroler adalah sebuah komputer kecil ("special purpose computers") di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan paralel, Port input/output, ADC. Mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program.
2. Sejarah mikrokontroler diawali pada tahun 1617, John Napier menemukan sistem untuk melakukan perkalian dan pembagian berdasarkan logaritma dan terus berkembang sampai yang populer sekarang adalah mikrokontroler jenis AVR.
3. Mikroprosesor adalah Central Processing Unit (CPU) didalam single chip, mikro-Komputer adalah mikroprosesor yang dihubungkan rangkaian pendukung, jika komponen yang menyusun sebuah mikro-komputer diletakkan bersama didalam single chip silicon maka disebut dengan mikrokontroler
4. Arsitektur Mikrokontroler terbagi atas dua jenis yaitu Arsitektur CISC (Complex Instruction Set Computer) dan Arsitektur RISC (Reduce Instruction Set Computer)
5. Jenis Mikrokontroler yang ada pada kalangan umum sekarang adalah TinyAVR, AT90S, ATmega, Xmega, dan AVR32.

## Soal Formatif

---

1. Jelaskan pengertian mikrokontroler?
2. Uraikan dengan singkat sejarah perkembangan mikrokontroler?
3. Jelaskan Perbedaan Mikrokontroler, Mikroprocessor, dan Mikro-Komputer?
4. Jelaskan Perbedaan Arsitektur mikrokontroler CISC dan RISC?
5. Uraikan Jenis-jenis Mikrokontroler?

## Bab 2

# Hardware dan Software Mikrokontroler AVR dan Arduino

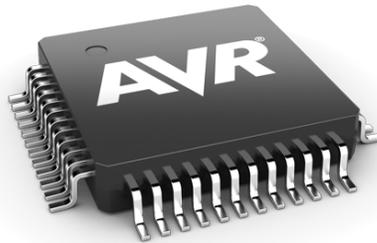
### Tujuan Pembelajaran

---

1. Peserta Didik mampu menjelaskan tentang mikrokontroler AVR.
2. Peserta Didik mampu menjelaskan tentang Arduino.
3. Peserta Didik mampu menguraikan jenis bahasa pemrograman mikrokontroler.
4. Peserta Didik mampu mencontohkan pengaplikasian sintak dasar bahasa pemrograman C.

## 2.1 Mikrokontroler AVR

Mikrokontroler AVR (*Alf and Vegards Risc processor*) dari Atmel ini menggunakan arsitektur RISC (*Reduced Instruction Set Computer*) yang artinya prosesor tersebut memiliki set instruksi program yang lebih sedikit dibandingkan dengan MCS-51 yang menerapkan arsitektur CISC (*Complex Instruction Set Computer*).



Gambar 2.1: Mikrokontroler AVR

Hampir semua instruksi prosesor RISC adalah instruksi dasar (belum tentu sederhana), sehingga instruksi-instruksi ini umumnya hanya memerlukan 1 siklus mesin untuk menjalankannya. Kecuali instruksi percabangan yang membutuhkan 2 siklus mesin. RISC biasanya dibuat dengan arsitektur Harvard, karena arsitektur ini yang memungkinkan untuk membuat eksekusi instruksi selesai dikerjakan dalam satu atau dua siklus mesin, sehingga akan semakin cepat dan handal. Proses downloading programnya relatif lebih mudah karena dapat dilakukan langsung pada sistemnya.

Sekarang ini, AVR dapat dikelompokkan menjadi 6 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, keluarga AT90CAN, keluarga AT90PWM dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya, sedangkan dari segi arsitektur dan instruksi yang digunakan hampir sama.

### 2.1.1 Arsitektur

Mikrokontroler AVR RISC (*Reduced instruction set computing* atau Komputasi set instruksi yang disederhanakan pertama kali digagas oleh John Cocke) adalah perangkat yang di desain untuk berjalan dengan cepat, dengan menggunakan instruksi mesin yang disederhanakan sehingga dapat meningkatkan kinerja dari mikrokontroler. Sebelum ada RISC, namanya CISC (*Complex Instruction Set Computers*). Dengan penggunaan instruksi yang lebih seder-

hana memberikan kontribusi pada kecepatan dengan instruksi mesin yang terbatas. Mikrokontroler AVR RISC dapat berjalan pada single cycle dari prosesor clock, yang berarti Mikrokontroler AVR dengan clock 8 MHz, dapat mengeksekusi sekitar 8 juta instruksi perdetiknya atau 8 MIPS (*million instruction per second*).

### 2.1.2 Memori

Bagian memori dari mikrokontrol Atmel RISC AVR berbasis Harvard Model, yang mana memorinya terbagi sehingga dapat meningkatkan kecepatan akses dan meningkatkan kapasitas. CPU membagi antarmuka untuk bagian kode memori FLASH, bagian memori data, dan memori EEPROM.

**Memori FLASH** memori FLASH merupakan blok dari memori FLASH yang dimulai dari lokasi 0x000 dan ukurannya tergantung dari mikrokontroler yang digunakan. Memori FLASH merupakan memori non-volatile dan digunakan untuk menyimpan kode eksekusi dan konstanta, karena kode-kode tersebut akan digunakan kembali meskipun mikrokontroler tidak terhubung ke catu daya, oh iya.. yang dimaksud non-volatile yaitu kode yang disimpan dalam memori tidak hilang meskipun mikrokontroler tidak dialiri listrik, ruang memori antara 16 bit pada setiap lokasi untuk menangani instruksi mesin yang khususnya single-16 bit word.

**Memori Data** Memori data Atmel AVR khasnya terdiri dari tiga bagian memori baca/tulis terpisah. bagian terendah terdiri dari 32 register kerja umum, yang diikuti oleh 64 register I/O, yang diikuti oleh internal SRAM. Register kerja umum hanya : digunakan untuk menyimpan variabel lokal dan data temporal yang digunakan oleh program saat dieksekusi dan dapat juga digunakan untuk penyimpanan data variabel global, 64 register I/O digunakan sebagai antarmuka untuk perangkat I/O dan peripheral yang berada di papan mikrokontroler. dan internal SRAM digunakan sebagai area penyimpanan variable umum dan juga untuk prosesor stack.

**Register-register**, Register kerja umum menempati 32 sel terendah dalam data memori. register ini kebanyakan digunakan seperti data penyimpanan dalam kalkulator yang mana hanya disimpan sementara. terkadang digunakan untuk menyimpan variable lokal, dan terkadang variable global, dan terkadang sebagai pengarah ke memori yang digunakan oleh prosesor. singkatnya prosesor menggunakan 32 register kerja sebagaimana program dieksekusi.

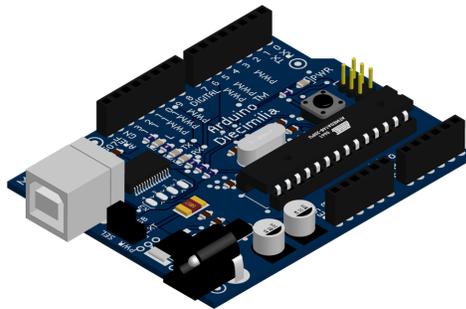
**Register I/O** , Setiap register memberikan akses ke register kontrol atau ke register data I/O peripheral yang berada dalam mikrokontroler. Programmer lebih sering menggunakan I/O register untuk mengantar-muka ke peripheral I/O dari mikrokontroler. ukuran Register I/O tergantung dari perangkat. setiap register I/O memiliki nama, sebuah alamat I/O, dan alamat SRAM.

**SRAM** , Bagian SRAM dari memori digunakan untuk menyimpan variabel yang tidak dapat disimpan kedalam register dan untuk menyimpan prosesor saat ini.

**Memori EEPROM** bagian memori EEPROM adalah area memori baca/tulis yang non volatile. ini biasanya digunakan untuk menyimpan data yang tidak boleh hilang saat catu daya dilepas (mikrokontroler dimatikan) dan dipasang kembali (mikrokontroler dinyalakan). ruang EEPROM dimulai dari 0x000 dan ke nilai maksimum tergantung spesifikasi mikrokontroler yang digunakan.

## 2.2 Arduino Board

Arduino merupakan papan rangkaian sistem minimum mikrokontroler yang memang dirancang untuk bisa digunakan dengan mudah oleh para seniman dan desainer (yang memang bukan orang teknik). Dengan demikian, tanpa mengetahui bahasa pemrograman, Arduino bisa digunakan untuk menghasilkan karya yang canggih.



Gambar 2.2: Arduino Board

Menggunakan Arduino sangatlah membantu dalam membuat suatu prototyping ataupun untuk melakukan pembuatan proyek. Arduino memberikan

I/O yang sudah fix dan bisa digunakan dengan mudah. Arduino dapat digabungkan dengan modul elektro yang lain sehingga proses perakitan jauh lebih efisien. Para desainer hanya tinggal membuat software untuk mendayagunakan rancangan H/D yang ada. Software jauh lebih mudah untuk dimodifikasi tanpa memindahkan kabel.

Saat ini arduino sangat mudah dijumpai dan ada beberapa perusahaan yang mengembangkan sistem H/D *open source* ini. Pengembangan-pengembangan tersebut antara lain:

- Arduino : <http://www.arduino.cc/>
- I-CubeX : <http://www.infusionsystems.com/>
- Arie Robot Project Junior : <http://www.arobotineveryhome.com/>
- Dwengo : <http://www.dwenfo.org/>
- EmbeddedLab : <http://www.embedded.arch.ethz.ch/>
- GP3 : <http://www.awce.com/gp3.htm/>

Di antara pengembang-pengembang yang ada, Arduino merupakan salah satu pengembang yang banyak digunakan. Pada buku ini, kita akan menggunakan produk pengembang dari Arduino. Keistimewaan arduino adalah hardware yang *open source*. Hal ini sangatlah memberi keleluasan bagi pengguna untuk bereksperiment secara bebas dan gratis.

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri.

Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino. Dan seperti Microcontroller yang banyak jenisnya, Arduino lahir dan berkembang, kemudian muncul dengan berbagai jenis. Diantaranya adalah:

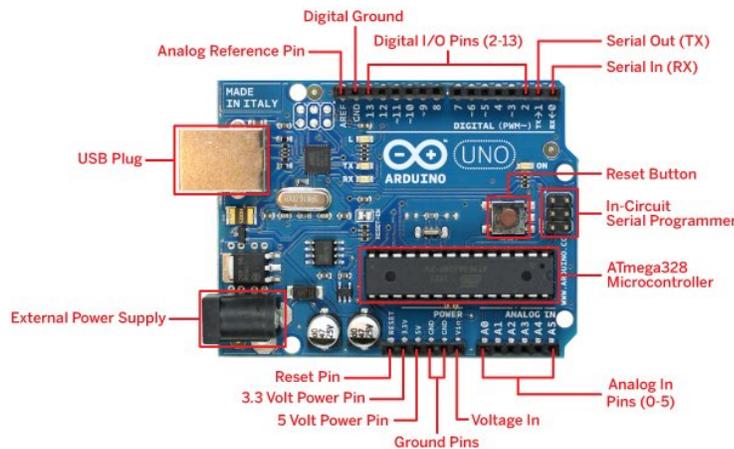
1. Arduino Uno Jenis yang ini adalah yang paling banyak digunakan. Terutama untuk pemula sangat disarankan untuk menggunakan Arduino Uno. Dan banyak sekali referensi yang membahas Arduino Uno.

Versi yang terakhir adalah Arduino Uno R3 (Revisi 3), menggunakan ATMEGA328 sebagai Microcontrollernya, memiliki 14 pin I/O digital dan 6 pin input analog. Untuk pemograman cukup menggunakan koneksi USB type A to To type B. Sama seperti yang digunakan pada USB printer.

2. Arduino Due Berbeda dengan saudaranya, Arduino Due tidak menggunakan ATMEGA, melainkan dengan chip yang lebih tinggi ARM Cortex CPU. Memiliki 54 I/O pin digital dan 12 pin input analog. Untuk pemogramannya menggunakan Micro USB, terdapat pada beberapa handphone.
3. Arduino Mega Arduino Mega Mirip dengan Arduino Uno, sama-sama menggunakan USB type A to B untuk pemogramannya. Tetapi Arduino Mega, menggunakan Chip yang lebih tinggi ATMEGA2560. Dan tentu saja untuk Pin I/O Digital dan pin input Analognya lebih banyak dari Uno.
4. Arduino Leonardo Bisa dibilang Leonardo adalah saudara kembar dari Uno. Dari mulai jumlah pin I/O digital dan pin input Analognya sama. Hanya pada Leonardo menggunakan Micro USB untuk pemogramannya.
5. Arduino Fio Bentuknya lebih unik, terutama untuk socketnya. Walau jumlah pin I/O digital dan input analognya sama dengan uno dan leonardo, tapi Fio memiliki Socket XBee. XBee membuat Fio dapat dipakai untuk keperluan projek yang berhubungan dengan wireless.
6. Arduino Lilypad Bentuknya yang melingkar membuat Lilypad dapat dipakai untuk membuat projek unik. Seperti membuat amor iron man misalkan. Hanya versi lamanya menggunakan ATMEGA168, tapi masih cukup untuk membuat satu projek keren. Dengan 14 pin I/O digital, dan 6 pin input analognya.
7. Arduino Nano Sepertinya namanya, Nano yang berukuran kecil dan sangat sederhana ini, menyimpan banyak fasilitas. Sudah dilengkapi dengan FTDI untuk pemograman lewat Micro USB. 14 Pin I/O Digital, dan 8 Pin input Analog (lebih banyak dari Uno). Dan ada yang menggunakan ATMEGA168, atau ATMEGA328.
8. Arduino Mini Fasilitasnya sama dengan yang dimiliki Nano. Hanya tidak dilengkapi dengan Micro USB untuk pemograman. Dan ukurannya hanya 30 mm x 18 mm saja.

9. Arduino Micro Ukurannya lebih panjang dari Nano dan Mini. Karena memang fasilitasnya lebih banyak yaitu; memiliki 20 pin I/O digital dan 12 pin input analog.
10. Arduino Ethernet Ini arduino yang sudah dilengkapi dengan fasilitas ethernet. Membuat Arduino kamu dapat berhubungan melalui jaringan LAN pada komputer. Untuk fasilitas pada Pin I/O Digital dan Input Analognya sama dengan Uno.
11. Arduino Esplora Rekomendasi bagi kamu yang mau membuat gadget seperti Smartphone, karena sudah dilengkapi dengan Joystick, button, dan sebagainya. Kamu hanya perlu tambahkan LCD, untuk lebih mempercantik Esplora.
12. Arduino Robot Ini adalah paket komplit dari Arduino yang sudah berbentuk robot. Sudah dilengkapi dengan LCD, Speaker, Roda, Sensor Infrared, dan semua yang kamu butuhkan untuk robot sudah ada pada Arduino ini.

Pembahasan pada buku ini menggunakan arduino jenis Arduino UNO. Arduino UNO adalah jenis arduino yang paling populer digunakan dikalangan pelajar maupun kalangan umum. Berikut ini adalah perangkat keras arduino dengan fungsi dari masin-masing bagiannya.



Gambar 2.3: Arduino UNO

## 2.3 Algoritma Pemrograman

Dalam pembuatan sebuah program menurut Saputra (2015) terdapat 7 langkah umum yaitu:

1. Mendefinisikan masalah, Langkah yang pertama dilakukan adalah mendefinisikan permasalahan. langkah ini harus dilakukan untuk menentukan masalah yang ada serta ditentukan pula input dan output program.
2. Mencari solusi, Kemudian ditentukan solusi dari permasalahan yang dihadapi. Bila untuk mendapatkan solusi harus melalui langkah yang terlalu rumit dapat dilakukan pembagian masalah dalam beberapa modul-modul kecil agar mudah untuk dikerjakan. Lalu modul-modul kecil tersebut digabungkan menjadi satu untuk dapat menentukan solusi.
3. Menentukan algoritma, Dalam pemilihan algoritma, pemrogram atau analis harus menggunakan algoritma yang sesuai dan efisien untuk masalah yang dihadapi.
4. Menulis program, Penulisan program bisa dilakukan dengan menggunakan bahasa pemrograman yang dikuasai dan memiliki kompatibilitas dengan perangkat keras yang akan menggunakan program tersebut.
5. Menguji program, Bila program sudah selesai dibuat, pengujian diperlukan untuk mengetahui apakah program yang dibuat sudah layak untuk digunakan.
6. Mendokumentasikan program, Penulisan dokumentasi yang biasanya dilupakan oleh pemrogram menjadi sangat penting saat akan dilakukan perubahan pada program yang dibuat. penulisan program ini dapat dilakukan dengan menulis komentar pada source code tentang kegunaannya (variabel, parameter, prosedur, fungsi).
7. Merawat program, Program yang sudah selesai dibuat juga perlu dirawat dengan pendeteksian bug yang belum diketahui sebelumnya juga penambahan fasilitas baru yang mempermudah pengguna program.

Dari penjelasan diatas, dapat kita peroleh kesimpulan bahwa pembelajaran algoritma pemrograman adalah bagian dari langkah-langkah pembuatan program. Untuk lebih jelasnya tentang algoritma pemrograman akan diulas sebagai berikut :

### 2.3.1 Definisi Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.

Menurut E. Knuth (2011) dalam bukunya yang berjudul *The Art of Computer Programming*, algoritma harus mempunyai lima ciri penting :

1. Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas (berhingga)
2. Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (ambiguous)
3. Algoritma memiliki nol atau lebih masukan (input)
4. Algoritma mempunyai nol atau lebih keluaran (output)
5. Algoritma harus efektif dan efisien.

### 2.3.2 Program

Agar dapat dilaksanakan oleh pemroses, algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program. Jadi program adalah perwujudan atau implementasi teknis algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh pemroses. Kata algoritma dan kata program seringkali dipertukarkan dalam penggunaannya. Algoritma adalah urutan langkah-langkah penyelesaian masalah sedangkan Program adalah realisasi algoritma dalam bahasa pemrograman. Program ditulis dalam salah satu bahasa pemrograman dan kegiatan membuat program disebut pemrograman (programming). Orang yang menulis program disebut programmer. Tiap-tiap langkah di dalam program disebut pernyataan atau instruksi. Jadi, program tersusun atas sederetan instruksi. Bila suatu instruksi dilaksanakan, maka operasi-operasi yang bersesuaian dengan instruksi tersebut dikerjakan oleh pemroses. Secara garis besar perangkat yang membutuhkan alogaritma tersusun atas empat komponen utama, yakni:

1. piranti masukan berfungsi untuk memasukkan data atau program ke dalam memori

2. piranti keluaran berfungsi untuk menampilkan hasil dari eksekusi program
3. unit pemroses utama berfungsi mengerjakan operasi-operasi dasar
4. memori berfungsi untuk menyimpan program dan data atau informasi.

Proses Eksekusi Program, Mekanisme eksekusi sebuah program adalah. Program disimpan di dalam memori melalui piranti masukan. Ketika sebuah program dieksekusi maka setiap instruksi program akan dikirim dari memori ke unit pemroses utama. Unit pemroses utama kemudian akan menjalankan operasi sesuai instruksi-instruksi yang dibaca. Apabila sebuah instruksi membutuhkan data masukan, maka piranti masukan akan membaca data masukan, mengirimkan ke memori kemudian mengirimkan ke unit pemroses utama untuk diproses. Apabila eksekusi program menghasilkan data keluaran, maka data keluaran akan disimpan di dalam memori, kemudian dikirim ke piranti keluaran.

### **2.3.3 Perbedaan interpreter dan compiler**

Interpreter:

1. Menerjemahkan instruksi per instruksi.
2. Source program tidak harus ditulis lengkap.
3. Bila terjadi kesalahan instruksi, dapat langsung diperbaiki secara interaktif.
4. Tidak menghasilkan objek program
5. Pemrosesan program lebih lambat, karena setiap instruksi yang dikerjakan harus diinterpretasi ulang.
6. Source code program terus dipergunakan.

Compiler:

1. Menerjemahkan secara keseluruhan.
2. Source program harus ditulis lengkap.
3. Bila terjadi kesalahan dalam kompilasi, source program harus dibenarkan dan proses kompilasi diulang kembali.

4. Menghasilkan objek program.
5. Pemrosesan program lebih cepat, karena program sudah dalam bahasa mesin.
6. Source code program dipergunakan satu kali saat kompilasi program.

### 2.3.4 Penulisan Algoritma

Pada umumnya terdapat tiga jenis cara penulisan algoritma, yakni :

#### Cara Deskriptif

Dengan notasi ini, deskripsi setiap langkah dijelaskan dengan bahasa yang jelas. Contohnya : Algoritma Bilangan Maksimum Diberikan tiga buah bilangan bulat. Carilah bilangan bulat maksimum di antara ketiga bilangan tersebut.

Deskripsi :

- baca bilangan 1.
- baca bilangan 2.
- bandingkan bilangan 1 dan bilangan 2, kita ambil yang lebih besar, jika kedua bilangan tersebut sama besar, dapat kita ambil bilangan 1, dan sebut bilangan tersebut MAX.
- baca bilangan 3.
- bandingkan MAX dengan bilangan 3, dan pilih yang lebih besar, jika keduanya sama besar, pilih MAX dan sebut bilangan tersebut MAX.
- keluarkan sebagai output MAX

#### Pseudocode

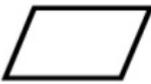
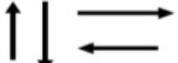
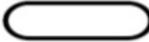
Pseudocode adalah notasi yang menyerupai bahasa pemrograman tingkat tinggi. Keuntungan menggunakan notasi pseudocode adalah memberikan kemudahan bagi programmer untuk menerjemahkan ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap pseudocode dengan notasi bahasa pemrograman.

## Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-rutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti:

- Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
- Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
- Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
- Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
- Setiap langkah dari aktivitas harus berada pada urutan yang benar
- Gunakan simbol-simbol flowchart yang standar.

Simbol-simbol yang sering digunakan dalam flowchart antara lain:

	Input/output; digunakan utk mewakili data i/o		Proses; digunakan utk mewakili suatu proses
	Garis alir; Menunjukkan arus dari proses		Keputusan; digunakan utk suatu seirksi kondisi didlm program
	Penghubung; Menunjukkan penghubung ke hlmn yg sama atau hlmn lain		Proses terdefinisi; menunjukkan suatu operasi yg rinciannya ditunjukkan ditempat lain
	Persiapan; digunakan memberi awal nilai suatu besaran		Terminal; menunjukkan awal & akhir dari suatu proses

Gambar 2.4: Simbol Flowchart dan Kegunaannya

## 2.4 Bahasa Pemrograman

Untuk menjalankan mikrokontroler dibutuhkan sebuah program atau kode inisialisasi dalam bentuk biner 0 dan 1. Dalam menginisialisasi program tersebut dibutuhkan sebuah pengkonversi (compiler) untuk mempermudah manusia dalam menentukan fungsi mikrokontroler tersebut dalam arti lain mengubah bahasa manusia menjadi bahasa kode mikrokontroler yaitu 0/1. Bahasa yang dimaksud dalam pengkonversi tersebut disebut sebagai code program / pemrograman. Dalam pemrograman suatu mikrokontroler terdapat 3 level atau tingkatan berdasarkan tingkat kesukarannya yaitu bahasa mesin, assembly, dan bahasa tingkat tinggi.

### 2.4.1 Bahasa Mesin

Bahasa mesin (machine language/machine code) adalah bahasa yang dipahami oleh komputer. Kode yang terdapat pada bahasa ini berupa kode bilangan biner yang dapat diproses oleh mikrokontroler sehingga sulit dipahami oleh manusia. Bahasa mesin umumnya bersifat spesifik, yaitu tidak lintas jenis mikrokontroler sehingga suatu jenis mikrokontroler memiliki bahasa pemrograman yang berbeda dengan jenis mikrokontroler lainnya. File bahasa mesin memiliki ekstensi .hex, karena untuk kepentingan manusia yang memprogram kode biner dikonversikan menjadi kode bilangan heksadesimal.

### 2.4.2 Bahasa Assembly

Bahasa Assembly terdiri dari instruksi berupa representasi mnemonic dari instruksi berupa kode bilangan biner dari bahasa mesin. Umumnya mnemonic berupa singkatan tiga atau empat huruf dari kata yang mewakili suatu instruksi. Contohnya instruksi Assembly adalah sebagai berikut :

- SUB adalah kode Assembly untuk instruksi subtract, yaitu mengurangi suatu angka dari angka lain.
- CBI adalah kode Assembly untuk instruksi Clear Bit I/O, yaitu memberi logika nol pada suatu pin I/O digital.

Instruksi pada Bahasa Assembly diterjemahkan menjadi bahasa mesin oleh perangkat lunak Assembler. Seperti juga bahasa mesin bahasa Assembly juga spesifik untuk suatu jenis mikrokontroler tertentu.

### 2.4.3 Bahasa Tingkat Tinggi

Bahasa pemrograman tingkat tinggi (high level programming language) memiliki sintaks yang lebih mendekati bahasa manusia. Hal ini menyebabkan bahasa tingkat tinggi lebih mudah untuk dipelajari. Meski demikian bahasa tingkat tinggi umumnya menghasilkan ukuran kode yang lebih besar dibandingkan bahasa Assembly. Bahasa tingkat tinggi memerlukan perangkat lunak kompiler (compiler) untuk menerjemahkan kode menjadi bahasa mesin. Contoh compiler atau bahasa pemrograman yang berada pada tingkat tinggi:

- Bahasa C
- Bahasa Basic

Pada buku ini akan lebih memfokuskan pada penggunaan bahasa tingkat tinggi C. Pemrograman mikrokontroler dilakukan dalam beberapa langkah. Langkah pertama adalah penulisan kode program menggunakan perangkat lunak text editor yang menghasilkan file .asm untuk Assembly, .c untuk C, dan .bas untuk Basic. Langkah berikutnya adalah proses kompilasi program menggunakan perangkat lunak compiler yang menghasilkan file .hex yang berisi kode bahasa mesin. Sebelum dikompilasi file sumber diperiksa sintaksnya terlebih dahulu oleh preangakat lunak debugger. File .hex inilah yang dimasukkan ke mikrokontroler menggunakan perangkat lunak programmer.

Saat proses pemasangan program ini mikrokontroler dipasang pada rangkaian downloader dan dihubungkan dengan PC melalui perangkat keras programmer. Setelah diprogram mikrokontroler menjalankan fungsinya sebagai kontroler/pengendali pada rangkaian sistem minimum (Sismin) yang merupakan rangkaian yang berisi beberapa komponen tambahan pendukung mikrokontroler.

Saat ini tidak diperlukan rangkaian downloader khusus karena AVR memiliki fasilitas ISP (In System Programming) sehingga AVR dapat diprogram langsung pada rangkaian kerjanya menggunakan programmer. Beberapa perangkat lunak pemrograman AVR merupakan IDE (Integrated Development Environment). IDE adalah suatu perangkat lunak pemrograman yang memuat seluruh/sebagian besar fasilitas untuk memprogram mikrokontroler, seperti text editor, debugger, compiler, dan downloader, sehingga tidak diperlukan lagi perangkat lunak lainnya. beberapa IDE bahkan juga memiliki fasilitas simulator dan komunikasi serial.

Berbeda dengan mikrokontroler AVR, proses Upload program pada Arduino sudah tidak menggunakan perangkat keras berupa downloader karena

papan arduino telah tersedia chip bootloader yang memungkinkan upload program dilakukan lewat data serial USART (rx dan tx).

## 2.5 Bahasa Pemrograman C

Menurut Iswanto (2011) bahasa C adalah bahasa pemrograman yang dapat dikatakan berada di bahasa beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin, sedangkan beruas tinggi berorientasi pada manusia. Bahasa beraras rendah, misalnya bahasa assembler. ditulis dengan sandi yang hanya dimengerti oleh mesin sehingga hanya digunakan bagi yang memrogram mikrokontroler. Bahasa beruas rendah merupakan bahasa yang membutuhkan kecermatan tinggi bagi pemrogram karena perintahnya harus rinci, ditambah lagi masing-masing pabrik mempunyai sandi perintah sendiri. Bahasa tinggi relatif mudah digunakan karena ditulis dengan bahasa manusia sehingga mudah . dimengerti dan tidak tergantung mesinnya. Bahasa beraras tinggi umumnya digunakan pada komputer.

Pencipta bahasa C adalah Brian Wr Kemighan dan Denis M. Ritchi sekitar 1972. Penulisan program dalam bahasa C dilakukan dengan membagi dalam blok-blok sehingga bahasa C disebut bahasa terstruktur. Bahwa C dapat digunakan di berbagai mesin dengan mudah, mulai dari PC sampai mainframe, serta menggunakan berbagai sistem operasi misalnya DOS, UNIX, VMS, dan lain-lain.

### 2.5.1 Penulisan Bahasa C AVR

Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu sehingga bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca.

Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan *main()*. Program yang dijalankan berada dalam tubuh program dan dimulai dengan tanda kurung buka”{” dan diakhiri dengan kurung tutup”}” Semua yang tertulis di dalam tubuh program disebut blok.

Tanda ”( )” digunakan untuk mengapit argumen suatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam fungsi *main*, tidak ada argumen sehingga tak ada data dalam (). Dalam tubuh fungsi antara tanda ”[” dan tanda ”]”, ada sejumlah pernyataan yang merupakan perintah dan harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri tanda titik koma ”;”.

```

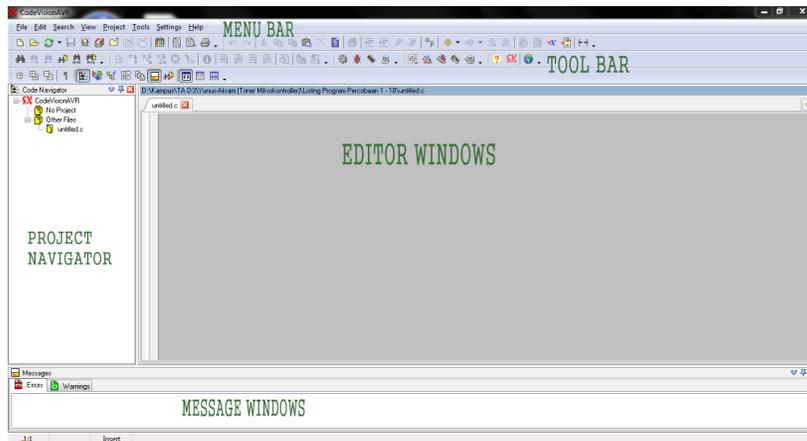
#include <mega16.h>           //memanggil file library
#include <delay.h>
#define Motor PORTA.0        //mendefinisikan fungsi PIN
int x;                        //membuat data dengan tipe int

void main(void) {
  DDRA=0xFF;                 //isi program
  PORTA=0xFF;
  while(1) {                 //isi program yang berulang
  }
}

```

Gambar 2.5: Contoh Penulisan Program Pada C AVR

Baris pertama `#include <...>` bukanlah pernyataan sehingga tak diakhiri tanda titik koma `;`. Baris tersebut meminta kompiler untuk menyatakan file yang namanya ada di antara tanda `<...>` dalam proses kompilasi. File-file ini (berekstensi `.h`) berisi deklarasi fungsi ataupun variable. File ini disebut header dan digunakan semacam perpustakaan untuk pernyataan yang ada di tubuh program.



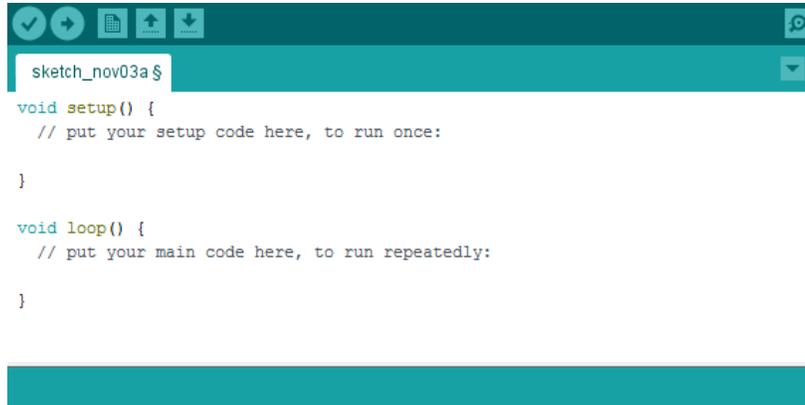
Gambar 2.6: Code Vision AVR

Ada beberapa program yang dapat digunakan sebagai editor dan *compiler* untuk mikrokontroler AVR, salah satunya adalah CodeVisionAVR. CodeVisionAVR adalah salahsatu alat bantu pemrograman (*programming tools*) yang bekerja dalam lingkungan pengembangan perangkat lunak yang terintegrasi (*Integrated Development Environment, IDE*). Seperti aplikasi IDE lainnya, CodevisionAVR dilengkapi dengan *sourcecode editor*, *compiler*, *linker*,

dan dapat memanggil Atmel AVR studio untuk debuggernya.

## 2.5.2 Penulisan Bahasa C Arduino

Penulisan program bahasa C di Arduino:



```
sketch_nov03a$  
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Gambar 2.7: Contoh Penulisan Program Pada C Arduino

Penjelasan tentang fungsi setiap sintak pada Arduino hampir sama dengan AVR. Beberapa bagian yang membedakannya adalah sebagai berikut:

- "void main(void)" menjadi "void setup()": penanda atau sebagai penunjuk isi dari program dalam pemrograman AVR, isi dari program dimulai dari tanda kurung kurawal ({} dan diakhiri pula dengan tanda tutup kurung kurawal ({})
- perintah "while(1)" sendiri juga dapat digunakan di arduino, tapi secara khusus aplikasi arduino sudah menyediakan tempat untuk menulis program yang memiliki karakteristik berulang (looping) yang tertulis "void loop()"

Secara Prinsip, *sketch* selalu melibatkan dua fungsi, yaitu *setup()* dan *loop()*. Kode Sketch minimal dapat dilihat pada gambar 2.7. Baris yang mengandung void dan nama fungsi adalah judul fungsi, sedangkan bagian { } dinamakan tubuh fungsi. Semua definisi fungsi melibatkan judul fungsi dan tubuh fungsi. Menurut Kadir (2015) fungsi sendiri adalah deretan instruksi yang diberi suatu nama, umumnya fungsi memberikan nilai ketika dipanggil. Nilai yang diberikan tersebut dinamakan nilai balik.

### 2.5.3 Tipe Data

Dalam Pemrograman Mikrokontroler, ram rom maupun register harus diisi data untuk menjalankan suatu program tertentu, data yang diisi pun beragam tergantung besar bit data yang digunakan. Jenis data tersebut dapat dilihat di tabel berikut:

Tabel 2.1: Tipe Data

Tipe	Ukuran (Bit)	Range
bit	1	0 - 1
char	8	-128 - 127
unsigned char	8	0 - 255
signed char	8	-128 - 127
int	16	-32768 - 32767
short int	16	-32768 - 32767
unsigned int	16	0 - 65535
signed int	16	-32768 - 32767
long int	32	-2147483648 - 2147483647
unsigned long int	32	0 - 4294967295
signed long int	32	-2147483648 - 2147483647
float	32	1.175e-38 - 3.402e38
doubel	32	1.175e-38 - 3.402e38

### 2.5.4 Operator

Operator adalah source code yang digunakan dalam mengelolah data baik itu membandingkan, menyamakan, menjumlahkan, dls.

Tabel 2.2: operator kondisi

Operator Kondisi	Keterangan
<	lebih kecil
<=	lebih kecil atau sama dengan
>	lebih besar
>=	lebih besar atau sama dengan
==	sama dengan
!=	tidak sama dengan

Tabel 2.3: Operator Aritmatika

Operator Aritmatika	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagi (Modulus)

Tabel 2.4: Operator Logika

Operator Logika	Keterangan
!	Boolean NOT
&&	Boolean AND
—	Boolean OR

Tabel 2.5: Operator Bitwise

Operator Bitwise	Keterangan
~	komplement bitwise
&	Bitwisw AND
—	Bitwise OR
^	Bitwise Exclusive OR
>>	Right Shift
<<	Left Shift

Tabel 2.6: Operator Assignment

Operator Assignment	Keterangan
=	Memasukkan nilai
+=	Menambahkan nilai dari keadaan semula
-=	Mengurangi nilai dari keadaan semula
*=	Mengalikan nilai dari keadaan semula
/=	Membagi nilai dari keadaan semula
%=	Mamasukkan nilai sisa dari pembagian
<<=	Memasukkan Shift Left
>>=	Memasukkan Shift Right
&=	Memasukkan Bitwise AND
=	Memasukkan Bitwisw XOR
\=	Memasukkan Bitwise OR

## Rangkuman

---

1. Mikrokontroler yang populer dalam kalangan masyarakat saat ini adalah mikrokontroler jenis AVR dan pengembangannya yang dikenal dengan nama Arduino.
2. Mikrokontroler AVR adalah jenis mikrokontroler yang tampilannya hanya sebatas IC mikrokontroler tanpa tambahan perangkat interfaces.
3. Arduino adalah Papan yang menggunakan jenis mikrokontroler AVR yang telah mengalami perkembangan di bagian penggunaan port dan uploading program (bootloader).
4. Bahasa pemrograman terdiri dari tiga level yaitu bahasa mesin, assembly, dan bahasa tingkat tinggi.
5. Bahasa pemrograman C adalah bahasa level tinggi yang populer digunakan di kalangan umum dan memiliki source code yang open source.
6. Dalam penggunaan bahasa C hal perlu diperhatikan ada beberapa hal yaitu, sintak, penulisan source code, tipe data, operator operator kondisi.

## Soal Formatif

---

1. Jelaskan apa itu mikrokontroler AVR?
2. Jelaskan apa itu Arduino?
3. Uraikanlah jenis-jenis bahasa pemrograman untuk mikrokontroler?
4. Tuliskan contoh perintah atau sintak dasar bahasa pemrograman C?

## Bab 3

# Interface Data Digital

### Tujuan Pembelajaran

---

1. Peserta Didik mampu menjelaskan pengertian data digital yang berkaitan dengan mikrokontroler.
2. Peserta Didik mampu menentukan data digital sebagai input dan output pada mikrokontroler.
3. Peserta Didik mampu mengimplementasikan fungsi antarmuka data digital pada mikrokontroler AVR dan Arduino.
4. Peserta Didik mampu membuat rangkaian dan pemrograman data digital pada mikrokontroler AVR dan Arduino.

### 3.1 Data Digital

Dalam ilmu elektronika, digital tidak menunjuk pada besar dari voltase atau arus pada suatu tempat dalam rangkaian, tetapi suatu keadaan yang berkaitan dengan voltase atau arus tertentu. Hanya terdapat dua keadaan, yaitu keadaan yang diartikan satu dan keadaan yang diartikana nol. Misalnya "ada" diartikan sebagai 1 dan voltase "tidak ada" diartikan sebagai 0. Dalam praktek kata voltase ada atau voltase tidak ada harus dijelaskan lebih rinci. Misalnya voltase ada terdapat kalau voltase pada sambungan tersebut antara 3V dan 5V dan tidak ada voltase berarti voltase pada sambungan tersebut lebih kecil dari 0.4V. Dengan cara ini voltase tidak perlu terlalu tepat, tetapi cukup kalau voltase tersebut memiliki kira-kira suatu nilai tertentu. Dengan ketentuan ini rangkaian-rangkaian digital menjadi kurang peka terhadap derau atau perubahan voltase supply atau gangguan yang lain.

Dengan mengartikan keadaan pada suatu rangkaian listrik sebagai angka 0 atau angka 1, maka suatu rangkaian listrik digital dapat dianalisis menggunakan bilangan dalam system dual. Dengan aljabar Boolean transformasi-transformasi tertentu bias dilakukan dengan bilangan-bilangan tersebut. Hal ini yang dilakukan dalam komputer.

Elektronika, khususnya elektronika digital, akan terus mengalami perkembangan. Perkembangan apapun, meskipun menuju ke arah perbaikan, selalu disertai kekurangan-kekurangan maupun hal-hal yang tidak menyenangkan. Para insinyur yang telah berpengalaman sekalipun kadang merasa tertekan untuk dapat mengikuti kepesatan perkembangan elektronika. Lebih-lebih bagi para pemula tentu saja menghadapi masalah yang jauh lebih berat.

Teknologi mutakhir yang paling mengagumkan dan yang memiliki fleksibilitas tinggi adalah komputer dan mikroprosesor. Komputer dan mikroprosesor dibangun dari rangkaian digital. Rangkaian digital terdiri dari sekelompok gerbang logika (logic gate) yang dapat menampilkan tugas-tugas yang sangat berguna. Rangkaian digital menjadi otak dunia teknologi. Rangkaian digital banyak digunakan untuk pengendalian proses (otomatisasi), mulai dari proses industri dengan tingkat kompleksitas yang tinggi, robot, peralatan laboratorium, alat rumah tangga, hiburan, hingga permainan anak.

Elektronika sering tampak seperti hutan belantara yang membingungkan oleh karena seakan-akan berisi hal-hal yang tidak jelas kaitannya. Di dalam suatu rangkaian terdiri dari komponen-komponen dengan nama-nama aneh, parameter-parameter yang tidak sederhana, dan teori yang rumit. Pernya-

taan ini tidak bertujuan untuk membuat kita menjadi pesimis, tetapi sebaliknya agar bersiap-siap untuk bekerja keras jika ingin berkecimpung dalam bidang elektronika. Thomas A. Edison pernah berpesan bahwa : Ada cara untuk menyempurnakan. Singkaplah !.

Penelitian yang tidak kenal lelah meneruskan berbagai penemuan untuk menyempurnakan yang sudah ada dan untuk mendapatkan hal-hal yang baru. Melalui evaluasi gagasan, penelitian, kreativitas, inspirasi dan kerja keras telah ditemukan hal-hal baru yang lebih inovatif dan semakin sempurna. Kita dapat mempelajari elektronika sampai sejauh yang kita perlukan. Oleh karenanya kita tidak perlu pesimis asal siap bekerja keras sampai dengan taraf tertentu kita dapat menguasainya.

### 3.2 Sistem Digital dan Sistem Analog

Perancang dan teknisi elektronik harus mempunyai pengetahuan baik sistem digital maupun analog. Perancang harus memutuskan apakah sistem akan menggunakan teknik analog atau digital atau kombinasi keduanya. Teknisi harus membangun protipe atau mencari kerusakan dan perbaikan pada sistem digital analog, dan kombinasi keduanya.

Sistem elektronika analog telah lebih populer di zaman dulu. Informasi dunia nyata yang berhubungan dengan pengukuran waktu, kecepatan, berat tekanan, intensitas cahaya dan posisi semuanya analog di alam.

Sistem digital diperlukan ketika data harus disimpan, digunakan untuk perhitungan atau diperagakan sebagai angka/huruf. Sesuatu yang lebih kompleks yang mengatur pengukuran banyaknya cairan dalam tangki air adalah sistem digital. Beberapa keuntungan yang diberikan dalam menggunakan rangkaian digital dibandingkan dengan analog adalah sebagai berikut :

- IC yang tidak mahal dapat digunakan dengan sedikit komponen eksternal.
- Informasi dapat disimpan untuk periode pendek atau tak didefinisikan
- Data dapat digunakan untuk perhitungan presisi.
- Sistem dapat didesain lebih muda menggunakan kelompok logika digital compatible/praktis.
- Sistem dapat deprogram dan menunjukkan kemampuan berdasar.

Batasan rangkaian digital adalah sebagai berikut :

- Kebanyakan kejadian dunia nyata adalah analog dalam lingkungannya.
- Pemroses analog biasanya sederhana dan lebih cepat.

Rangkaian digital kelihatannya lebih menonjodan lebih produktif terutama karena IC digital yang diandalkan harganya murah. Alasan lain untuk perkembangan popularitas sistem digital adalah keakuratannya, ditambah stabilitas, kemudahan dipindah, memori, kenikmatan pemakaian, dan kesederhanaan desainnya.

### 3.3 Aplikasi Interface Data Digital

Berikut beberapa contoh komponen atau perangkat antarmuka yang menggunakan data digital baik input maupun output:

#### 3.3.1 Led Emiting Diode

Light Emitting Diode (LED) adalah komponen elektronika yang dapat memancarkan cahaya. Sesuai dengan namanya, LED adalah salah satu jenis diode. Sebagaimana yang diketahui, diode adalah komponen yang hanya dapat mengalirkan arus listrik satu arah. Arus listrik hanya mengalir kalau tegangan positif dihubungkan ke kaki yang disebut anode dan tegangan negatif dihubungkan ke kaki yang dinamakan katode.

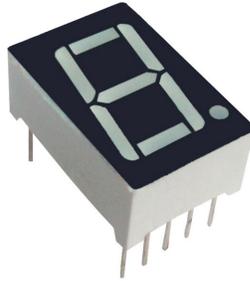


Gambar 3.1: Led Emitting Diode

Pada pengaplikasian LED yang terkontrol lewat IC mikrokontroler, yang diperlukan hanya menghubungkan salah satu PIN dari mikrokontroler ke kaki anode LED, dan kaki Katode ke GND rangkaian. Dengan memberikan sinyal *high* (nilai 1 atau bertegangan) ke kaki anode maka LED akan menyala, sebaliknya jika memberikan sinyal *low* (nilai 0) maka LED akan dalam keadaan tidak menyala atau mati.

### 3.3.2 Seven Segment

Prinsip kerja seven segment sama dengan led, seven segment terdiri dari 7 led yang disusun membentuk angka delapan dan tambahan 1 led untuk titik.

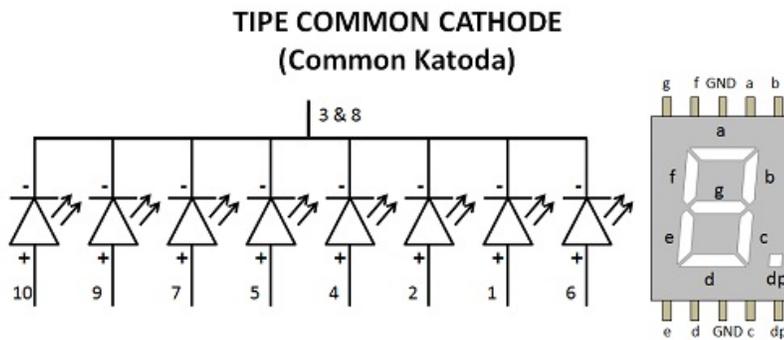


Gambar 3.2: Seven Segment

Terdapat dua jenis seven segment yaitu seven segment commod cathode (katoda) dan seven segment commod anode(anoda).

#### Seven Segment Common Cathode

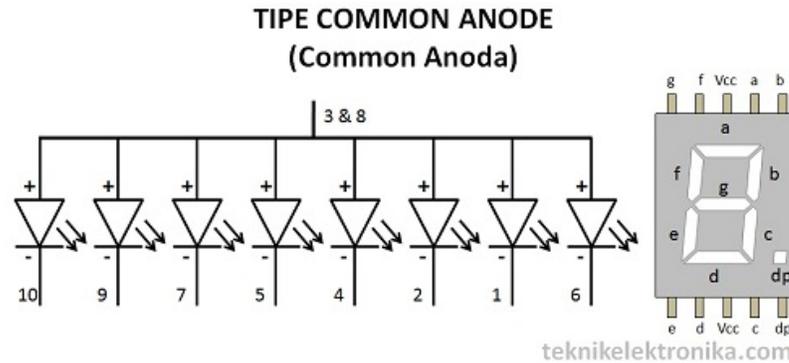
Pada seven Segmen jenis Common Cathode (Katoda), Kaki Katoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan Kaki Anoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Katoda yang terhubung menjadi 1 Pin ini merupakan Terminal Negatif (-) atau Ground sedangkan Signal Kendali (Control Signal) akan diberikan kepada masing-masing Kaki Anoda Segmen LED.



Gambar 3.3: Seven Segment Common Cathode

### Seven Segment Common Anode

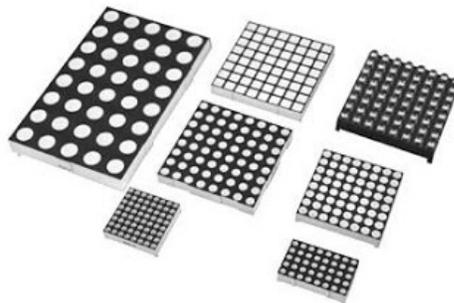
Pada seven Segmen jenis Common Anode (Anoda), Kaki Anoda pada semua segmen LED adalah terhubung menjadi 1 Pin, sedangkan kaki Katoda akan menjadi Input untuk masing-masing Segmen LED. Kaki Anoda yang terhubung menjadi 1 Pin ini akan diberikan Tegangan Positif (+) dan Signal Kendali (control signal) akan diberikan kepada masing-masing Kaki.



Gambar 3.4: Seven Segment Common Anode

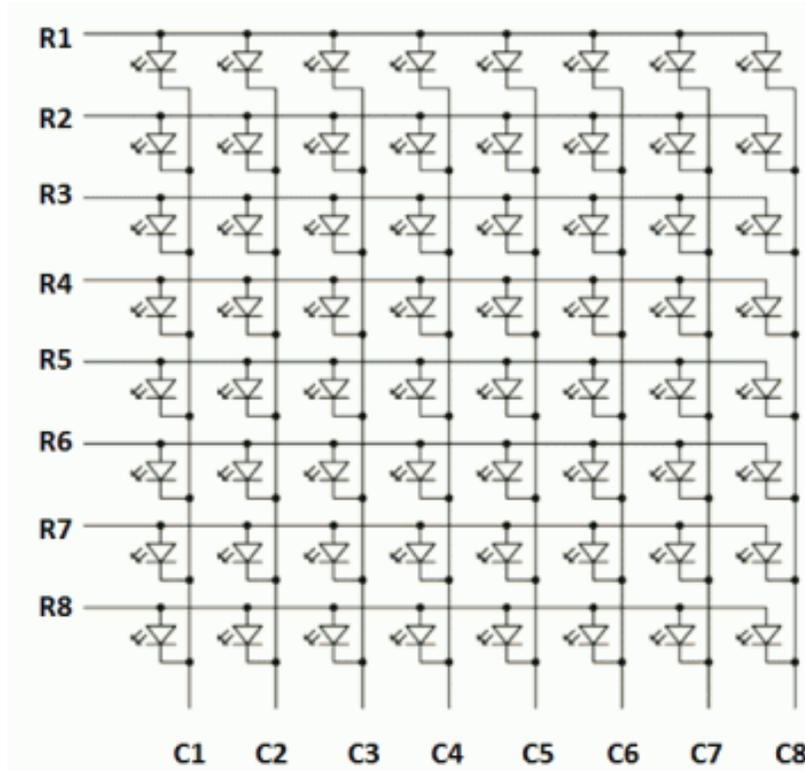
### 3.3.3 Led Matrix

Dot Matrix atau matrix led merupakan kumpulan titik cahaya yang tersusun menjadi sejumlah kolom dan baris. Pada komponen ini led tersusun pada kolom dan baris yang sudah ditentukan, dan setiap pin di kolom dan baris tersebut harus tersambung ke data digital di mikrokontroler untuk dikontrol. Contoh matriks dapat dilihat di gambar berikut.



Gambar 3.5: Led Matrix

Pada dasarnya led matrix adalah display led yang disusun sedemikian rupa sehingga untuk membuat sebuah karakter dibutuhkan kombinasi tegangan antara pin baris dan kolom.

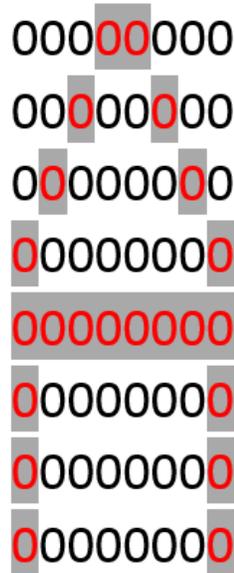


Gambar 3.6: Rangkaian Led penyusun Led Matrikx

Kemudian untuk mempermudah kontrol dan menghemat pin maka diperlukan proses scanning pada bagian colom sedangkan pada baris diberikan bit sesuai huruf/karakter yang akan ditampilkan yang bersesuaian dengan posisi scanning.

Scanning untuk kolom dimaksud memberikan "1" *high* ( untuk common katoda) atau "0" untuk common anoda , untuk kolom lainnya diberi nilai negasi dari kolom yg diberi nilai 1 sebelumnya begitu selanjutnya untuk kolom berikutnya sampai kolom terakhir dari rangkaian led matrix dan berulang dari depan lagi. Proses ini dilakukan sangat cepat sehingga mata kita melihatnya tetap sebagai suatu karakter yg diam.

misalnya ingin menampilkan huruf A , secara umum digambarkan sebagai berikut:



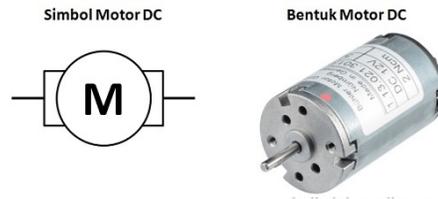
Gambar 3.7: Menampilkan Karakter "A" pada Led Matrikx

dari gambar dapat dijelaskan dengan logika saat kolom pertama ( scan kolom 1 = 1 /*high*) maka bit yg diberi pada baris berupa "1110000", sehingga jika ketemu 1 vs 1 tidak ada arus mengalir, jadi LED mati, jika 1 ketemu 0 maka arus mengalir, led menyala. Begitu juga untuk kolom kedua, ketika kolom kedua diberi tegangan maka pada baris bit yg diberikan adalah "1101011" dan seterusnya, ketika kolom nya mencapai ujung maka akan diulang ke kolom 1 lagi. Untuk melakukan scanning biasanya dilakukan dengan memanfaatkan shift register, atau paling sederhana dengan menggunakan IC 4017 yang di cascade menjadi led berjalan yang panjang.

### 3.3.4 Motor

Motor atau lebih tepatnya motor listrik adalah komponen elektrik yang sifatnya merubah besaran elektrik menjadi fisik berupa putaran. Motor listrik terdiri atas tiga bagian utama yang disebut stator (bagian yang diam) dan rotor (bagian yang bergerak) dan air gap. Rotor dapat berputar karena adanya medan magnet yang dipengaruhi oleh arus listrik. Air gap adalah bagian yang memisahkan rotor dan stator. Motor terdiri dari dua jenis yaitu motor AC dan motor DC. Motor DC menggunakan 2 data yang harus ter-

sambung ke data digital mikrokontroler namun ketika harus menggunakan motor DC yang membutuhkan arus tinggi harus menggunakan rangkaian penguat (driver) dari motor ke mikrokontroler. Sedangkan Motor AC biasanya memerlukan rangkaian berupa relay untuk menjalankannya, karena tegangan yang berasal dari mikrokontroler adalah DC sedangkan tegangan yang dibutuhkan adalah tegangan AC.



Gambar 3.8: Motor DC

### 3.3.5 Buzzer

Buzzer adalah komponen yang merubah besaran elektrik menjadi besaran suara. Penggunaan bazzer untuk anatarmuka mikrokontroler sama prinsipnya dengan LED. Yang diperlukan hanya menghubungkan salah satu PIN dari mikrokontroler ke kaki positif buzer, dan kaki satunya ke GND rangkaian.



Gambar 3.9: Buzzer

### 3.3.6 Push Button

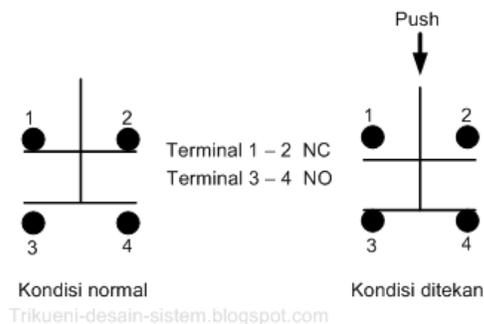
Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci). Sistem kerja unlock disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal. Sebagai



Gambar 3.10: Push Button

device penghubung atau pemutus, push button switch hanya memiliki 2 kondisi, yaitu On dan Off (1 dan 0). Istilah On dan Off ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi On dan Off.

Karena sistem kerjanya yang unlock dan langsung berhubungan dengan operator, push button switch menjadi device paling utama yang biasa digunakan untuk memulai dan mengakhiri kerja mesin di industri. Secanggih apapun sebuah mesin bisa dipastikan sistem kerjanya tidak terlepas dari keberadaan sebuah saklar seperti push button switch atau perangkat lain yang sejenis yang bekerja mengatur pengkondisian On dan Off. Berdasarkan



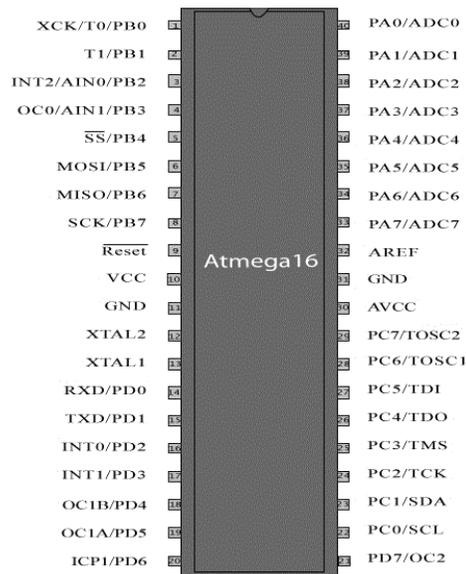
Gambar 3.11: Prinsip Kerja Push Button

fungsi kerjanya yang menghubungkan dan memutuskan, push button switch mempunyai 2 tipe kontak yaitu NC (Normally Close) dan NO (Normally Open).

- NO (Normally Open), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir). Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (Close) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem circuit (Push Button ON).
- NC (Normally Close), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus listrik). Dan ketika tombol saklar push button ditekan, kontak NC ini akan menjadi membuka (Open), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem circuit (Push Button Off).

### 3.4 Data Digital Mikrokontroler AVR

IC mikrokontroler dikemas (packaging) dalam bentuk yang berbeda. Namun pada dasarnya fungsi kaki yang ada pada IC memiliki persamaan. Gambar salah satu bentuk IC seri mikrokontroler AVR. Berikut adalah contoh mikrokontroler jenis AVR dengan 40 Pin dan memiliki 32 Pin I/O yang bisa digunakan sebagai pin Digital.



Gambar 3.12: Contoh Susunan Pin ATmega

Terlihat pada gambar 3.21 sebagai contoh, gambar tersebut adalah data sheet ATmega16, terdapat 4 buah port yaitu Port A(PA), Port B(PB), Port C(PC), Port D(PD), keempat port tersebut merupakan jalur bi-directional yang semuanya dapat diprogram sebagai input ataupun output dengan pilihan internal puul-up.

**Port A** Merupakan 8-bit directional port I/O. Setiap pinnya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer Port A dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port A (DDRA) harus disetting terlebih dahulu sebelum Port A digunakan. Bit-bit DDRA diisi 0 jika ingin memfungsikan pin-pin port A yang bersesuaian sebagai input, atau diisi 1 jika sebagai output. Selain itu, kedelapan pin port A juga digunakan untuk masukan sinyal analog bagi A/D converter.

**Port B** Merupakan 8-bit directional port I/O. Setiap pinnya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer Port B dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port B (DDRB) harus disetting terlebih dahulu sebelum Port B digunakan. Bit-bit DDRB diisi 0 jika ingin memfungsikan pin-pin port B yang bersesuaian sebagai input, atau diisi 1 jika sebagai output. Pin-pin port B juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

**Port C** Merupakan 8-bit directional port I/O. Setiap pinnya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer Port C dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port C (DDRC) harus disetting terlebih dahulu sebelum Port C digunakan. Bit-bit DDRC diisi 0 jika ingin memfungsikan pin-pin port C yang bersesuaian sebagai input, atau diisi 1 jika sebagai output. Selain itu, dua pin port C (PC6 dan PC7) juga memiliki fungsi alternatif sebagai oscillator untuk timer/counter 2.

**Port D** Merupakan 8-bit directional port I/O. Setiap pinnya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer Port D dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port D (DDRD) harus disetting terlebih dahulu sebelum Port D digunakan. Bit-bit DDRD

diisi 0 jika ingin memfungsikan pin-pin port D yang bersesuaian sebagai input, atau diisi 1 jika sebagai output. Selain itu, pin-pin port D juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

**RESET** RST pada pin 9 merupakan reset dari AVR. Jika pada pin ini diberi masukan low selama minimal 2 machine cycle maka system akan di-reset.

**XTAL1** XTAL1 adalah masukan ke inverting oscillator amplifier dan input ke internal clock operating circuit.

**XTAL2** XTAL2 adalah output dari inverting oscillator amplifier.

**AVcc** Avcc adalah kaki masukan tegangan bagi A/D Converter. Kaki ini harus secara eksternal terhubung ke Vcc melalui lowpass filter.

**AREF** AREF adalah kaki masukan referensi bagi A/D Converter. Untuk operasionalisasi ADC, suatu level tegangan antara AGND dan Avcc harus dibeikan ke kaki ini.

Setiap port tersebut mempunyai tiga buah register bit, yaitu DDxn, PORTxn dan PINxn. Huruf x mewakili nama port, sedangkan n mewakili nama bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx dan bit PINxn terdapat pada I/O address PINx.

Port I/O pada mikrokontroler AVR dapat dikonfigurasi sebagai input/output, dengan cara mengubah I/O register data direction register. Bit DDxn dalam register DDRx (Data Direction Register) menentukan arah pin.

Tabel 3.1: My caption

	DDR bit=1	DDRbit=0
PORT bit=1	Output High	Input pull-up
PORT bit=0	Output Low	Input Floating

Misalnya, jika ingin port B dikonfigurasi sebagai output, maka bit Data Direction Register di Port B harus dirubah menjadi 1. Contoh: **DDRB.1=1;** : artinya bit 1 (pin1) pada port B telah dikonfigurasi menjadi pin output. Dan sebaliknya jika ingin port B dikonfigurasi sebagai input, maka bit Data Direction Register di Port B harus dirubah menjadi 0,. Contoh:

**DDRB.1=0;** : artinya bit 1 (pin1) pada port B telah dikonfigurasi menjadi pin input.

Lebih jelasnya berikut adalah contoh pemrograman menggunakan code-  
visionAVR.

```
#include <mega16a.h>

void main(void)
{
  PORTA=0xFF;    //output pin di port A berlogika 1
  DDRA=0xFF;    //semua pin di port A dikonfigurasi output

  PORTB=0xFF;    //input pin di port B pull-up aktif
  DDRB=0x00;    //semua pin di port B dikonfigurasi input
}
```

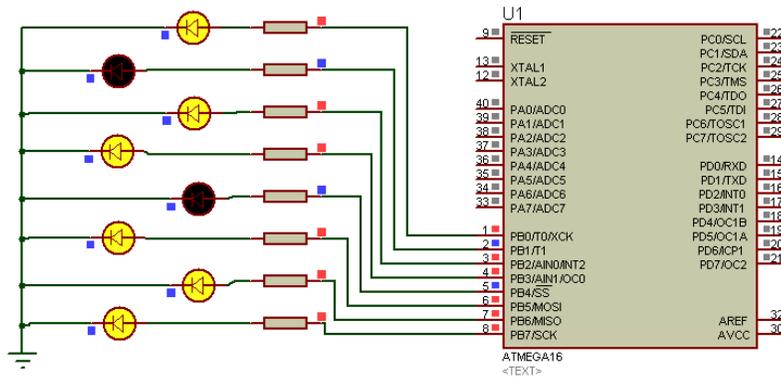
Gambar 3.13: Listing Program Konfogurasi pin digital I/O

- `#include <mega16a.h>` : memanggil librari sesuai dengan Ic mikro-kontroleryng digunakan
- `void main(void)` : menentukan isi program
- `PORT?` : menentukan logika jika sebgai output, dan menentukan pull-up jika sebagai input.
- `DDR?` : menentukan input atau output.
- `0xFF` : data dalam bilangan heksa, bisa juga menggunakan bilangan desimal dengan menggantinya menjadi "255" atau bilangan biner "0b11111111".

Tegangan keluaran pada pin I/O mikrokontroler ketika logiak "1" besarnya sekitar 4,2v dan arusnya 20mA. Port I/O sebagai output hanya memberikan arus (sourcing) sebesar 20 mA, keluaran dari suatu port mikrokontroler hanya dapat mengemudikan perangkat output dengan arus keci, sehingga untuk perlatan elektronik yang membutuhkan arus besar misalnya untuk menggurakkan motor DC perlu memberikan penguat tambahan, oleh karena itu biasanyadipergunakan transistor atai IC penguat (driver) agar port sebagai sinking current, seperti pada port untuk menyalakan led, yang akan menyala saat port diberikan logika low (0) dan mati ketika port diberi logika high(1).

### 3.4.1 Aplikasi Kontrol Output LED

8 buah led dipasang dengan menghubungkan masing-masing kaki anoda led ke port B mikrokontroler ATmega16 dan menggunakan resistor untuk menurunkan tegangan sehingga led dapat menyala normal.



Gambar 3.14: Contoh Rangkaian Interface dengan LED

Dari rangkaian pada gambar 3.14 dapat dibuat listing program sebagai berikut.

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

#include <mega16.h>

// Declare your global variables here

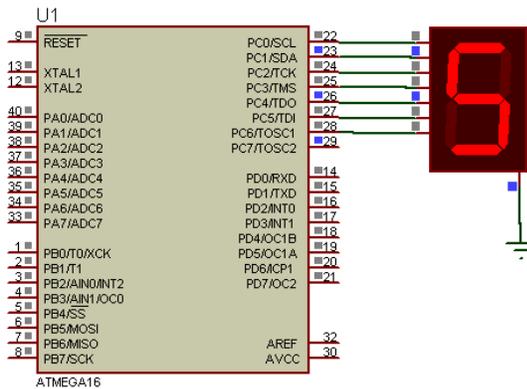
void main(void)
{
    PORTB=0xED;    //menentukan kondisi nyala / mati
    DDRB=0xFF;     //menentukan input / output
}

```

### 3.4.2 Aplikasi Kontrol Output Seven Segment

#### Rangkaian 7segment Commond Cathode

satu buah seven segment dihubungkan dengan port C pada mikrokontroler ATmega16 dan pada seven segment jenis commond cathode membutuhkan pin gnd untuk catu daya.



Gambar 3.15: Contoh Rangkaian Interface 7segment

Dari rangkaian pada gambar 3.15 dapat dibuat listing program sebagai berikut.

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

#include <mega16.h>

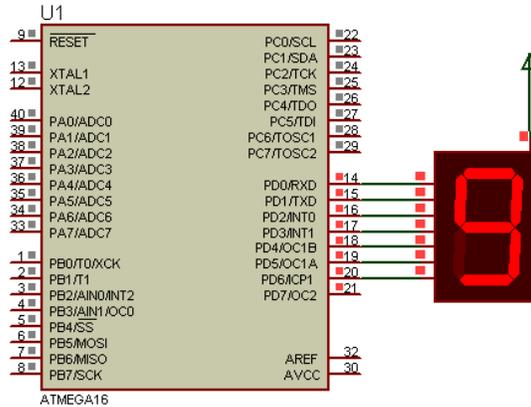
void main(void)
{
    PORTC=0x00;
    DDRC=0xFF;

    PORTC=0b01101101;    //membuat angka "5"
}

```

### Rangkaian 7segment Common Anode

satu buah seven segment dihubungkan dengan port D pada mikrokontroler ATmega16 dan pada seven segment jenis common Anode membutuhkan pin vcc(5v) untuk catu daya.



Gambar 3.16: Contoh Rangkaian Interface 7segment

Dari rangkaian pada gambar 3.16 dapat dibuat listing program sebagai berikut.

Listing Program:

```

/*****
Chip type           : ATmega16
Program type       : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model      : Small
External RAM size  : 0
Data Stack size   : 256
*****/

#include <mega16.h>

void main(void)
{
    PORTD=0x00;
    DDRD=0xFF;

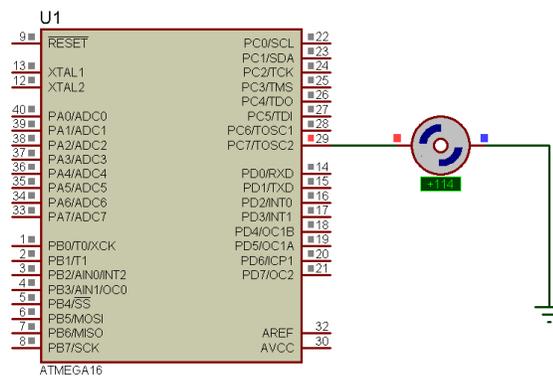
    PORTD=0b10010000; //membuat angka "9"
}

```

### 3.4.3 Aplikasi Kontrol Output Motor DC

#### Rangkaian Motor DC (satu data)

Pada dasarnya Motor DC dapat berputar ketika satu pinnya berlogika 1 dan yang satunya berlogika 0. Pada rangkaian berikut hanya menggunakan 1 pin dari motor ke mikrokontroler dan satu pinnya lagi duhubungkan ke ground (berlogika 0), sehingga untuk memutar motor DC cukup dengan memberikan sinyal high (logika 1) ke pin yang menghubungkan mikrokontroler dan motor DC. Pada rangkaian berikut menggunakan pin 7 pada Port C untuk mengontrol motor DC.



Gambar 3.17: Contoh Rangkaian Motor DC 1 data

Dari rangkaian pada gambar 3.17 dapat dibuat listing program sebagai berikut.

Listing Program:

```

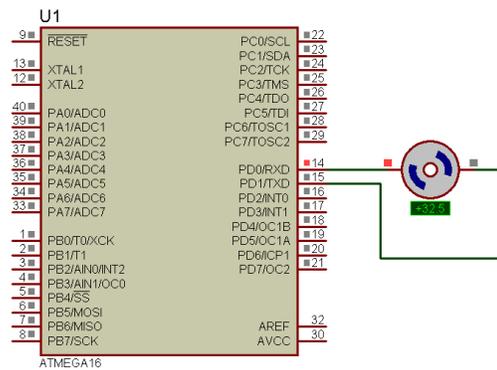
/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
*****/
#include <mega16.h>
// Declare your global variables here

void main(void)
{
  DDRC.7=1;
  PORTC.7=1;
}

```

## Rangkaian Motor DC (dua data)

Keunggulan dari pengontrolan motor dengan 2 data adalah arah rotasi yang dapat dirubah tiap saat sesuai dengan instruksi program, seperti memberikan waktu untuk putar kiri kemudian setelah selang waktu 5 detik akan merubah rotasi menjadi putar ke kanan. Contoh rangkaian dapat dilihat pada gambar berikut:



Gambar 3.18: Contoh Rangkaian Motor DC 2 data

Dari rangkaian pada gambar 3.18 dapat dibuat listing program sebagai berikut.

Listing Program:

```

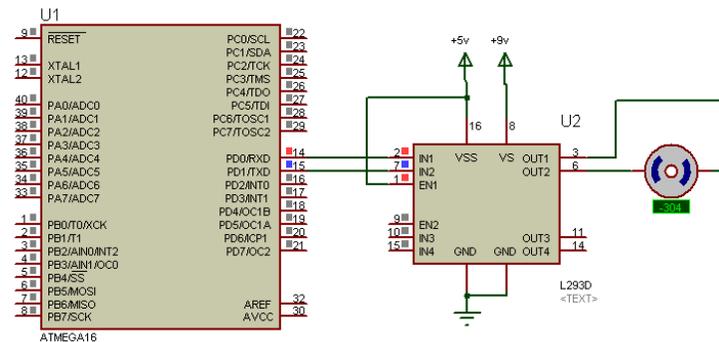
/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
*****/
#include <mega16.h>
#include <delay.h>
// Declare your global variables here

void main(void)
{
  DDRD.0=1; DDRD.1=1;           //menentukan input/output
  PORTD.0=1;PORTD.1=0;         //berputar ke kiri;
  delay_ms(5000);              //tunda selama 5 detik
  PORTD.0=0;PORTD.1=1;         //berputar ke kanan;
}

```

## Rangkaian Motor DC dengan Penguat

Penggunaan Motor DC akan sangat idel ketika ditambahkan rangkaian penguat baik berupa rangkaian penguat transistor maupun rangkaian penguat yang sudah dipaket dalam satu IC. Berikut adalah contoh ranaikain motor dengan tambahan rangkaian penguat L293D untuk menguatkan arus dan tegangan ke tengangan 9 volt.



Gambar 3.19: Contoh Rangkaian Motor dengan Penguat L293D

Pada dasarnya pemrograman tanpa rangkaian penguat atau menggunakan rangkaian penguat adalah sama. Listing porgram dibuat berdasarkan rangkaian pada gambar 3.19.

Listing Program:

```

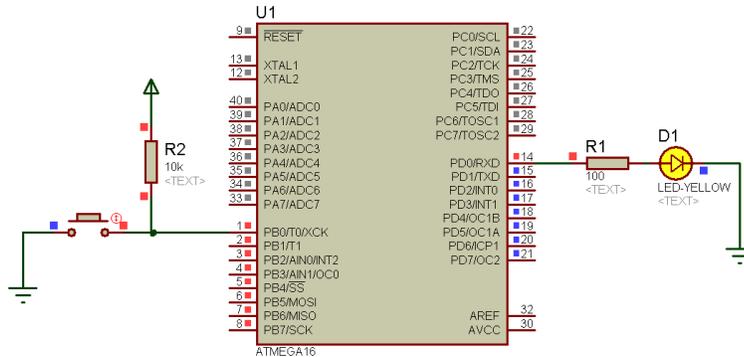
/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
*****/
#include <mega16.h>
#include <delay.h>
// Declare your global variables here

void main(void)
{
  DDRD.0=1; DDRD.1=1;           //menentukan input/output
  PORTD.0=1;PORTD.1=0;         //berputar ke kiri;
  delay_ms(5000);              //tunda selama 5 detik
  PORTD.0=0;PORTD.1=1;         //berputar ke kanan;
}

```

### 3.4.4 Aplikasi Input dengan Push Button

Berikut adalah rangkaian menyalakan Led dengan kondisi menyala ketika tombol ditekan.



Gambar 3.20: Rangkaian Kontrol LED dengan Input Push Button

Dari rangkaian pada gambar 3.20 dapat dibuat listing program sebagai berikut.

Listing Program:

```
AVR Core Clock frequency: 16,000000 MHz
Memory model      : Small
External RAM size  : 0
Data Stack size   : 256
*****/

#include <mega16.h>

void main(void)
{
    PORTB=0xFF;      //input mode pull-up
    DDRB=0x00;      //fungsi port sebagai input (untuk LED)
    PORTD=0x00;     //output start aktif low
    DDRD=0xFF;      //fungsi port sebagai output (push button)

    while (1)
    {
        if (PINB.0==0){PORTD.0=1;}
    }
}
```

### 3.5 Data Digital Arduino

Semua pin pada Arduino, baik pin digital maupun pin analog, dapat digunakan sebagai pin digital. Digital berarti sinyal yang dikirimkan/diterima bernilai 1 atau 0, on atau off, HIGH atau LOW, ada atau tidak ada sinyal. Berbeda dengan sinyal analog yang nilainya bersifat kontinu, yakni nilai antara 0 dan 1 dipertimbangkan. Pin digital berarti pin dapat menerima/mengirim sinyal digital.

Secara umum pin pada Arduino dapat dikonfigurasi ke dalam dua mode, yaitu mode input dan output. Mode input berarti mengeset pin agar dapat digunakan untuk menerima masukan sinyal. Mode output berarti mengeset pin agar dapat mengirimkan sinyal.

Untuk mengeset mode pin, kita gunakan fungsi `pinMode()`. Fungsi ini biasanya dipanggil di dalam fungsi `setup()`. fungsi ini memerlukan dua parameter, `pinMode([nomorPin], [mode])`. Parameter pertama diisi oleh nomor pin, dan parameter kedua diisi oleh konstanta `INPUT` atau `OUTPUT`, sesuai dengan mode yang ingin kita gunakan. Sebagai contoh, lihat pada kode berikut.



```
sketch_nov03b $
#define led 1
#define tombol 2

void setup() {
  pinMode(1, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
}

Done compiling.
```

Gambar 3.21: Listing Program Input/Output Digital

Pada dasarnya semua pin yang ada pada Arduino (ATMega) berada pada mode input secara default. Jadi ketika kita ingin menggunakan suatu pin sebagai input, maka kita tidak mesti menuliskan `pinMode(nomorPin, INPUT);`.

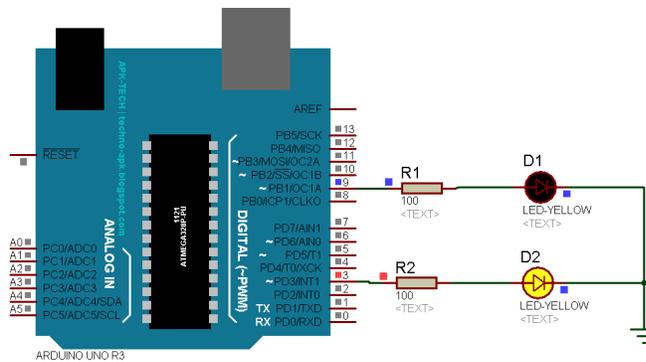
Untuk menerima input digital yang masuk ke pin, kita gunakan fungsi `digitalRead(nomorPin)`. Fungsi ini menerima satu parameter, yaitu nomor pin mana yang akan dibaca nilai inputnya. Fungsi ini akan mengembalikan

nilai 1 dan 0, atau HIGH dan LOW (HIGH adalah konstanta dengan nilai 1 dan LOW adalah konstanta dengan nilai 0)

Untuk mengirimkan sinyal digital, kita gunakan fungsi `digitalWrite(nomorPin, nilaiDigital)`. Fungsi ini dapat digunakan pada pin yang sebelumnya sudah diset ke mode OUTPUT. Parameter kedua adalah set nilai HIGH atau LOW. Apabila pin diset dengan nilai HIGH, maka voltase pin tersebut akan diset ke 5V (atau 3.3V pada board bertipe 3.3V) dan bila pin diset ke LOW, maka voltase pin tersebut akan diset ke 0V.

### 3.5.1 Aplikasi Kontrol Output LED

2 buah led dipasang dengan menghubungkan masing-masing kaki anoda led ke pin digital mikroontroler Arduino dan menggunakan resistor untuk menurunkan tegangan sehingga led menyala normal.



Gambar 3.22: Rangkaian Interface dengan LED

Dari rangkaian pada gambar 3.22 dapat dibuat listing program sebagai berikut.

Listing Program:

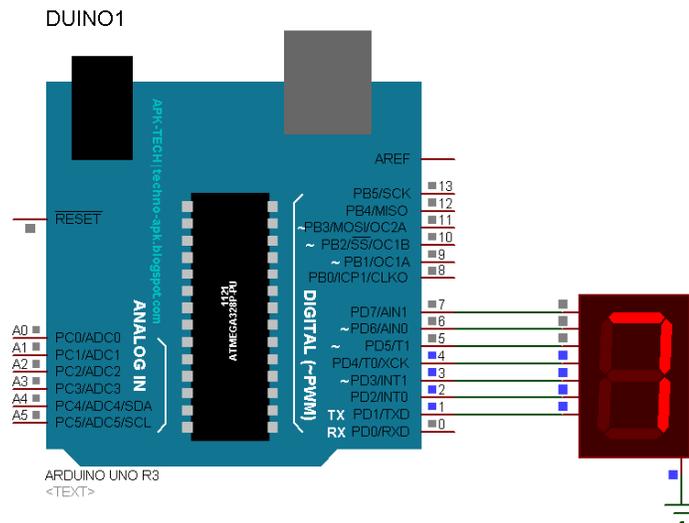
```
led
void setup() {
  pinMode(3,OUTPUT);
  pinMode(9,OUTPUT);
}

void loop() {
  digitalWrite(3,HIGH); //Led Menyala
  digitalWrite(9,LOW); //Led Mati
}
```

### 3.5.2 Aplikasi Kontrol Output Seven Segment

#### Rangkaian 7segment Common Cathode

Sebuah seven segment dihubungkan dengan pin digital pada Arduino UNO, dan pada seven segment jenis common cathode membutuhkan pin gnd untuk catu daya serta membutuhkan sinyal high(1) dari mikrokontroler untuk menyala.



Gambar 3.23: Contoh Rangkaian Interface 7segment

Dari rangkaian pada gambar 3.23, Program dibuat dengan menampilkan angka "7" pada 7segment.

#### Listing Program:

```

_7segment_co_cathode $
#define a 7 //mendefinisikan pin a pada 7segment terhubung dengan pin digital 7
#define b 6
#define c 5
#define d 4
#define e 3
#define f 2
#define g 1

void setup() {
  pinMode(a,OUTPUT); //set ke mode pn output
  pinMode(b,OUTPUT);

```

```

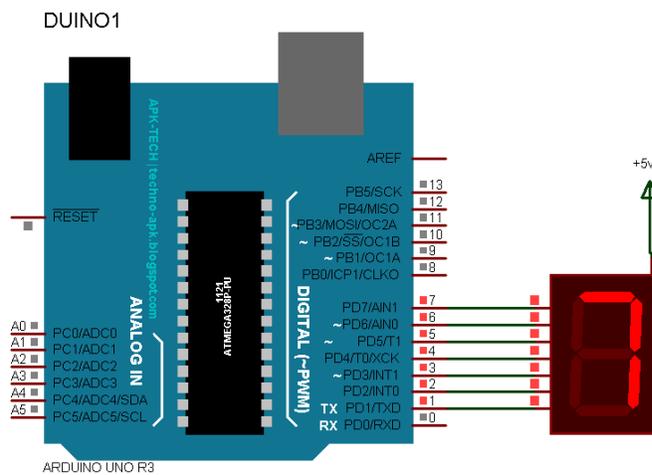
pinMode(c, OUTPUT);
pinMode(d, OUTPUT);
pinMode(e, OUTPUT);
pinMode(f, OUTPUT);
pinMode(g, OUTPUT);
}

void loop() {
digitalWrite(a, HIGH); //memberikan sinyal HIGH (menyala) ke pin a 7segment
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
}

```

### Rangkaian 7segment Commond Anode

Sama halnya dengan seven segment jenis commond cathode, seven segment commond anode dihubungkan dengan pin digital pada Arduino UNO dan pada seven segment jenis commond Anode membutuhkan pin vcc(5v) untuk catu daya serta memerlukan sinyal low dari mikrokontroler untuk mnyalakannya. Berikut adalah rangkaian seven segment yang menampilkan angka "7" sesuai instruksi program.



Gambar 3.24: Contoh Rangkaian Interface 7segment

Dari rangkaian pada gambar 3.24, program dibuat dengan menampilkan angka "7" pada seven segment.

Listing Program:

```

_7segment_co_anodhe
#define a 7 //mendefinisikan pin a pada 7segment terhubung dengan pin digital 7
#define b 6
#define c 5
#define d 4
#define e 3
#define f 2
#define g 1

void setup() {
  pinMode(a,OUTPUT); //set ke mode pn output
  pinMode(b,OUTPUT);
  pinMode(c,OUTPUT);
  pinMode(d,OUTPUT);
  pinMode(e,OUTPUT);
  pinMode(f,OUTPUT);
  pinMode(g,OUTPUT);
}

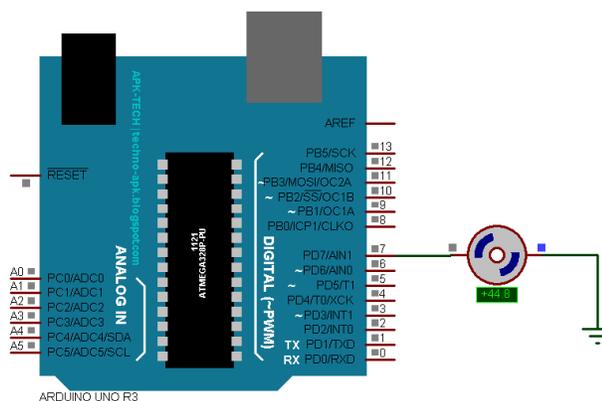
void loop() {
  digitalWrite(a,LOW); //memberikan sinyal LOW (menyala) ke pin a 7segment
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
}

```

### 3.5.3 Aplikasi Kontrol Output Motor DC

#### Rangkaian Motor DC (satu data)

Pada dasarnya Motor DC dapat berputar ketika satu pinnya berlogika 1 dan yang satunya berlogika 0.



Gambar 3.25: Contoh Rangkaian Motor DC 1 data

Pada rangkaian diatas hanya menggunakan 1 pin dari motor ke mikrokontroler dan satu pinnya lagi duhubungkan ke ground (berlogika 0), sehingga untuk memutar motor DC cukup dengan memberikan sinyal high (logika 1) ke pin yang menghubungkan mikrokontroler dan motor DC. Pada rangkaian diatas menggunakan pin digital 7 mikrokontroler arduino untuk mengontrol motor DC.

Dari rangkaian pada gambar 3.25 dapat dibuat listing program sebagai berikut.

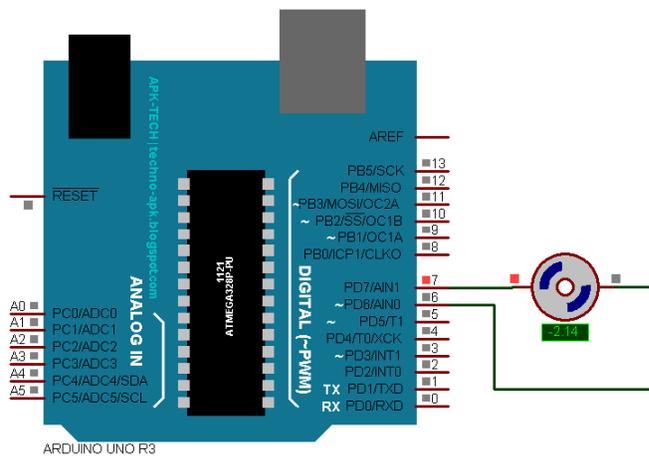
Listing Program:

```
Motor_1_Data
void setup() {
  pinMode(7,OUTPUT); //set ke mode pin output
}

void loop() {
  digitalWrite(7,HIGH); //memebrikan sinyal HIGH sehingga motor berputar
}
```

### Rangkaian Motor DC (dua data)

Keunggulan dari pengontrolan motor dengan 2 data adalah arah rotasi yang dapat dirubah tiap saat sesuai dengan instruksi program, seperti memberikan waktu untuk putar kiri kemudian setelah selang waktu 5 detik akan merubah rotasi menjadi putar ke kanan. Contoh rangkaian dapat dilihat pada gambar berikut:



Gambar 3.26: Contoh Rangkaian Motor DC 2 data

Dari rangkaian pada gambar 3.26 dapat dibuat listing program sebagai berikut.

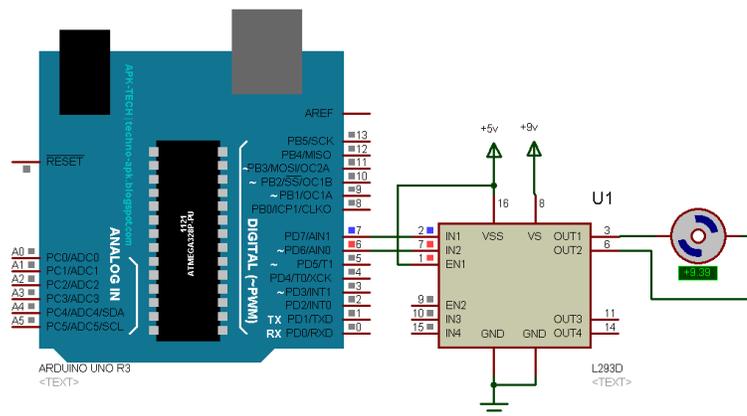
Listing Program:

```
Motor_2_Data
void setup() {
  pinMode(7,OUTPUT); //set ke mode pin output
  pinMode(6,OUTPUT);
}

void loop() {
  digitalWrite(7,HIGH);digitalWrite(6,LOW); //putar arah ke kanan
  delay(1000);
  digitalWrite(7,LOW);digitalWrite(6,HIGH); //putar arah ke kiri
  delay(1000);
}
```

### Rangkaian Motor DC dengan Penguat

Penggunaan Motor DC akan sangat idel ketika ditambahkan rangkaian penguat baik berupa rangkaian penguat transistor maupun rangkaian penguat yang sudah dipaket dalam satu IC. Berikut adalah contoh rangkaian motor dengan tambahan rangkaian penguat L293D untuk menguatkan arus dan tegangan ke tengangan 9 volt.



Gambar 3.27: Contoh Rangkaian Motor dengan Penguat L293D

Pada dasarnya pemrograman tanpa rangkaian penguat atau menggunakan rangkaian penguat adalah sama. Listing program dibuat berdasarkan rangkaian pada gambar 3.27.

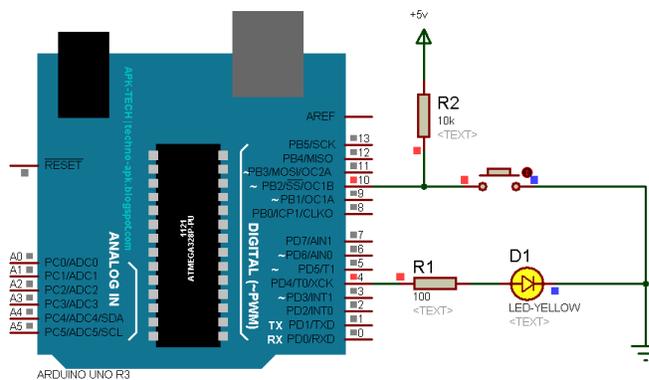
Listing Program:

```
Motor_2_Data
void setup() {
  pinMode(7,OUTPUT); //set ke mode pin output
  pinMode(6,OUTPUT);
}

void loop() {
  digitalWrite(7,HIGH);digitalWrite(6,LOW); //putar arah ke kanan
  delay(1000);
  digitalWrite(7,LOW);digitalWrite(6,HIGH); //putar arah ke kiri
  delay(1000);
}
```

### 3.5.4 Aplikasi Input dengan Push Button

Berikut adalah rangkaian menyalakan Led dengan kondisi menyala ketika tombol ditekan.



Gambar 3.28: Rangkaian Kontrol LED dengan Input Push Button

Dari rangkaian pada gambar 3.28 dapat dibuat listing program sebagai berikut.

Listing Program:

```
tombol_le $
void setup() {
  pinMode(4,OUTPUT); //set ke mode pin output (LED)
  pinMode(10,INPUT); //set ke mode pin input (Push Button)
}

void loop() {
  if(digitalRead(10)==0){ //jika tombol ditekan
    digitalWrite(4,HIGH); //maka led menyala
  }
}
```

## Rangkuman

---

1. Data digital adalah data yang hanya memproses 2 keadaan yaitu ada (1) atau tidak ada (0)
2. Komponen antarmuka yang dapat dikontrol sebagai input maupun output dengan data digital antaranya adalah, led, matrix, seven segment, buzzer, motor, pushbutton.
3. Syarat komponen yang dapat dikontrol oleh mikrokontroler adalah stand dengan dua keadaan yaitu ada atau tidak ada.
4. Pada umumnya pin pada IC mikrokontroler jenis AVR maupun Arduino, semua dapat digunakan sebagai data digital.

## Soal Formatif

---

1. Jelaskan pengertian data digital?
2. Jelaskan perbedaan data digital dengan data analog?
3. Led, 7segment, MotorDC, PushButton, Tentukanlah yang termasuk komponen interface yang dapat dikontrol menggunakan data input digital? Berikan alasannya.
4. Berikan 2 contoh implementasi data digital yang dapat dibuat menggunakan mikrokontroler AVR maupun Arduino?

## Tugas Latihan

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 2 buah komponen pushbutton dan 1 buah led. Buatlah dengan algoritma, jika tombol pertama ditekan maka LED menyala, dan ketika tombol kedua ditekan maka LED akan padam.

## Bab 4

# Interface Dengan LCD

### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan apa itu LCD.
2. Peserta didik mampu mengimplementasikan fungsi antarmuka LCD pada mikrokontroler AVR dan Arduino.
3. Peserta didik mampu membuat rangkaian dan pemrograman antarmuka LCD pada mikrokontroler AVR dan Arduino.

## 4.1 Pengenalan LCD

Menurut Adrianto (2015) LCD adalah suatu *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem *dot matriks*. LCD banyak digunakan sebagai *display* dari alat-alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya. Secara Garis Besar, terdapat dua jenis LCD yaitu LCD teks dan LCD grafik. LCD teks digunakan untuk menampilkan teks atau simbol-simbol tertentu. Adapun LCD grafik memungkinkan untuk menampilkan gambar.



Gambar 4.1: LCD 2x16 Karakter

Kemampuan LCD tidak hanya menampilkan angka, tetapi juga huruf, kata, dan semua sarana simbol dengan lebih bagus dan serbaguna daripada penampilan-penampilan yang menggunakan seven segment LED. Modul LCD mempunyai basic interfaces cukup baik yang sesuai dengan sistem mikrokontroler AVR maupun Arduino. Bentuk dan ukuran modul-modul berbasis karakter banyak ragamnya. Salah satu variasi bentuk dan ukuran yang tersedia dan dipergunakan pada peralatan ini adalah 16x2 karakter (panjang 16, baris 2, karakter 32) dan 16 pin.

Akses pin yang tersedia mempunyai 8 jalur hubungan data, 3 jalur hubungan kontrol, dan 3 jalur catu daya. Sementara pada modul LCD dengan fasilitas *back lighting* terdapat 2 jalur catu untuk *back lighting*. Dengan demikian, semua dapat ditampilkan dalam kondisi cahaya kecil.

Ketika power dinyalakan, display menampilkan sederet persegi gelap dan mungkin hanya pada sebagian display. Sel-sel karakter ini sebenarnya merupakan bagian yang mati. Modul display me-reset sendiri pada bagian awal ketika power dinyalakan, yaitu layar menjadi kosong sehingga karakter-karakter tidak dapat terlihat. Dengan demikian, Perlu memberikan perintah pada poin ini untuk menyalakan LCD.

## 4.2 Fungsi Pin-Pin Modul LCD

Modul LCD berukuran 16 karakter x 2 baris dengan fasilitas back lighting memiliki 16 pin yang terdiri atas 8 jalur data, 3 jalur kontrol, dan jalur catu daya.



Gambar 4.2: Data Seet LCD 2x16 Karakter

**Pin 1 dan 2** Merupakan sambungan catu daya, Vss, dan Vdd. Pin Vdd dihubungkan dengan tegangan positif catu daya, sedangkan Vss pada 0 volt atau ground. Meskipun demikian, data menentukan catu 5 Vdc (hanya beberapa mA), menyediakan 6V dan 4,5V yang keduanya bekerja dengan baik, bahkan 3V cukup untuk beberapa modul.

**Pin 3** Merupakan pin kontrol Vcc yang digunakan untuk mengatur kontras display. Idealnya pin ini dihubungkan dengan tegangan yang bisa diubah untuk memungkinkan pengaturan tingkatan kontras display sesuai kebutuhan.

**Pin 4** Merupakan register select (RS), masukan yang pertama dari 3 command control input. Dengan membuat RS menjadi high, data karakter dapat ditransfer dari dan menuju modulnya.

**Pin 5** Merupakan Read/Write (R/W). Cara memfungsikan perintah Write adalah R/W low atau menulis karakter ke modul. R/W high untuk membaca data karakter atau informasi status registernya.

**Pin 6** Merupakan Enable (E). Input ini digunakan untuk transfer aktual perintah-perintah atau karakter anatara modul dengan hubungan data. Ketika menulis ke display, data ditransfer hanya pada perpindahan

high/low. Namun, ketika membaca dari display, data akan menjadi lebih cepat tersedia setelah perpindahan dari low ke high dan tetap tersedia hingga sinyal low kembali.

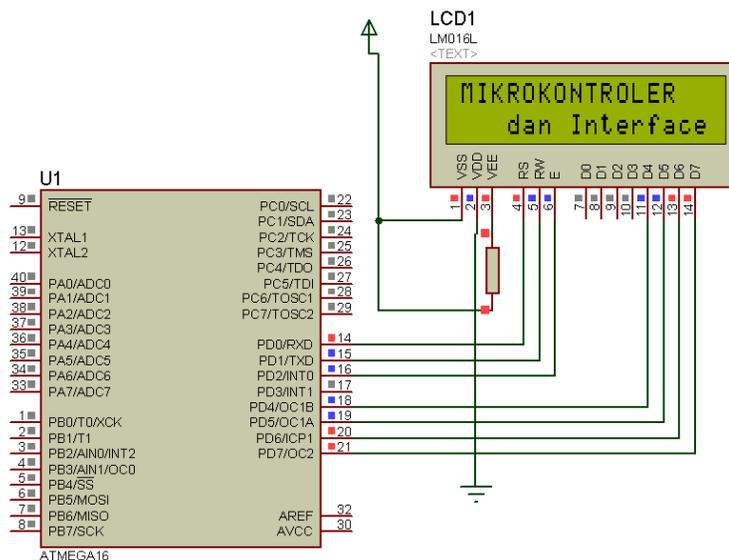
**Pin 7 sampai 14** Pin 7 sampai 14 adalah jalur 8 jalur data (D0-D7) di mana data dapat ditransfer ke dan dari display.

**Pin 15 dan 16** Pin 15 atau A (+) mempunyai level DC +5V dan berfungsi sebagai LED backlight +, sedangkan pin 16 atau K (-) memiliki level 0V dan berfungsi sebagai LED backlight.

### 4.3 Interface LCD di Mikrokontroler AVR

Penggunaan LCD dengan kontrol dari Mikrokontroler AVR menggunakan data output digital, yang dikirim sesuai dengan instruksi program. Pin data yang digunakan untuk berkomunikasi hanya menggunakan 4 pin data saja, yaitu D4-D7.

#### 4.3.1 Rangkaian Hardware



Gambar 4.3: Rangkaian Sederhana LCD

Sebagai contoh Gambar 4.3 merupakan rangkaian sederhana yang menghubungkan ATmega16 dengan LCD karakter 16x2. Rangkaian tersebut hanya menggunakan D4-D7 (mode 4 bit) yang dihubungkan dengan PORTD (port yang lain juga dapat digunakan). sehingga interfacingnya adalah sebagai berikut:

Tabel 4.1: Interface LCD

No Pin	Nama	Koneksi
4	RS	PORTD.0
5	RW	PORTD.1
6	E	PORTD.2
7-10	DB0 - DB3	No Connection
11	D4	PORTD.3
12	D5	PORTD.4
13	D6	PORTD.5
14	D7	PORTD.6

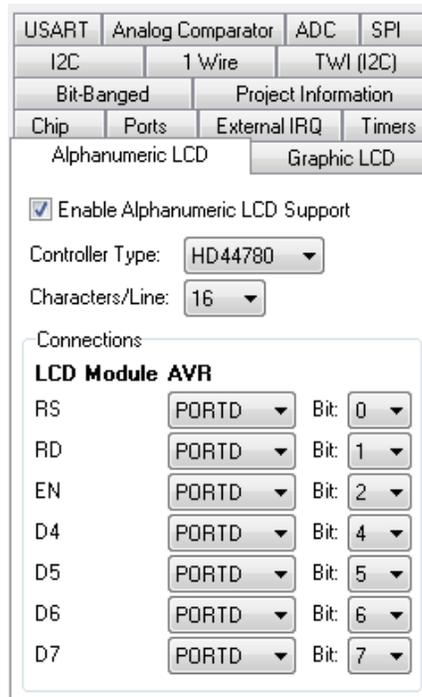
### 4.3.2 Pemrograman

Pada umumnya, pemrograman LCD pada AVR memanfaatkan library bawaan seperti yang tersedia pada aplikasi CodeVision AVR yaitu "*alcd.h*". Berikut adalah *source code* dan fungsinya pada penggunaan LCD.

Tabel 4.2: Program Dasar LCD

Source Code	Fungsi
<code>#include &lt;alcd.h&gt;</code>	Memanggil librari LCD untuk mengaktifkan.
<code>lcd_init(jml_kolom);</code>	Menentukan banyak kolom yang ditentukan
<code>lcd_gotoxy(x,y);</code>	Menentukanposisi kursor utuk memasukkan karakter. x adalh colom, dan y adalah baris
<code>lcd_puts("isi");</code> <code>lcd_putsf("isi");</code>	Menuliskan isi pada LCD
<code>lcd_clear();</code>	Membersihkan Tulisan pada LCD

Langkah pertama yang harus dilakukan adalah menentukan pin atau port mikrokontroler yang akan digunakan untuk mengontrol LCD, proses ini dilakukan pada *CodeWizardAVR*. Selanjutnya menuliskan kode program untuk menginput karakter ke LCD. Untuk Pemahaman lebih lanjut silahkan perhatikan gambar berikut.



Gambar 4.4: Kode Wizard untuk menentukan PIN

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

#include <mega16.h>

```

```

// Alphanumeric LCD functions
#include <alcd.h>

// Declare your global variables here

void main(void)
{
  lcd_init(16);

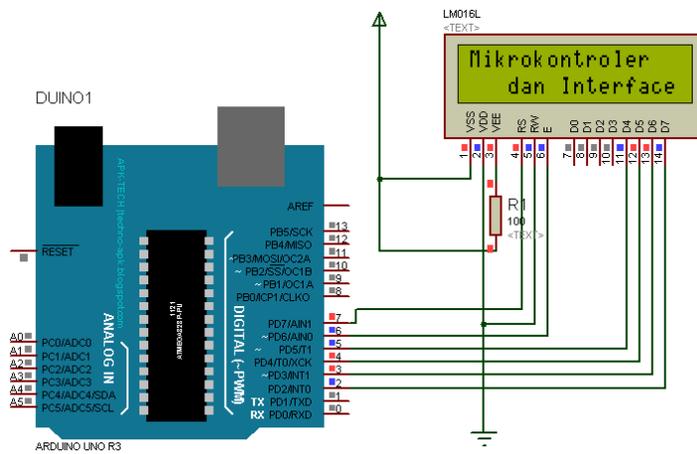
  while (1)
  {
    lcd_gotoxy(0,0);           //posisi penulisan
    lcd_puts("MIKROKONTROLER"); //tulisan
    lcd_gotoxy(3,1);
    lcd_puts("dan Interface");
  }
}

```

### 4.4 Interface LCD di Arduino

Sama seperti mikrokontroler AVR, LCD dengan kontrol dari Arduino menggunakan data output digital yang dikirim sesuai instruksi program.

#### 4.4.1 Rangkaian Hardware



Gambar 4.5: Arduino dengan Antarmuka LCD

Sebagai contoh Gambar 4.5 merupakan rangkaian sederhana yang menghubungkan Arduino UNO dengan LCD karakter 16x2. Rangkaian tersebut hanya menggunakan D4-D7 (mode 4 bit) yang dihubungkan dengan Pin Digital Arduino (Pin Analog juga bisa digunakan).

#### 4.4.2 Pemrograman Software

Pada aplikasi Arduino IDE memerlukan satu library yang bernama "*LiquidCrystal.h*", library ini menyediakan perintah dasar yang langsung dapat digunakan, perintah dasar dapat dilihat pada tabel berikut.

Tabel 4.3: Perintah Dasar Kontrol LCD pada Arduino

Source Code	Fungsi
<code>#include &lt;LiquidCrystal&gt;</code>	Memanggil librari LCD untuk mengaktifkan.
<code>LiquidCrystal lcd(12, 11, 5, 4, 3, 2);</code>	Menentukan Pin Interface LCD
<code>lcd.begin(16,2);</code>	Menentukan banyak kolom yang ditentukan
<code>lcd.setCursor(0,1);</code>	Menentukan posisi kursor untuk memasukkan karakter. x adalah colom, dan y adalah baris
<code>lcd.print("isi")</code>	Menuliskan isi pada LCD
<code>lcd.clear();</code>	Membersihkan Tulisan pada LCD

Berikut adalah contoh source code pada aplikasi arduino.IDE untuk menampilkan kata "*MikrokontrolerdanInterface*" pada LCD.

Listing Program:

```

lcd $
// Memanggil Librari
#include <LiquidCrystal.h>

// Pengaturan PIN Interface LCD
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);          // menggunakan LCD 16x2
}

```

```
void loop() {  
  lcd.setCursor(0,0);          //posisi tulisan  
  lcd.print("Mikrokontroler"); //tulisan  
  lcd.setCursor(3,1);  
  lcd.print("dan Interface");  
}
```

---

## Rangkuman

---

1. LCD adalah suatu display dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. LCD banyak digunakan sebagai display dari alat-alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya.
2. Terdapat dua jenis LCD yaitu LCD teks dan LCD grafik. LCD teks digunakan untuk menampilkan teks atau simbol-simbol tertentu. Adapun LCD grafik memungkinkan untuk menampilkan gambar.
3. Modul LCD berukuran 16 karakter x 2 baris dengan fasilitas back lighting memiliki 16 pin yang terdiri atas 8 jalur data, 3 jalur kontrol, dan jalur catu daya.
4. Penggunaan LCD dengan kontrol dari Mikrokontroler AVR maupun Arduino menggunakan data output digital, yang dikirim sesuai dengan instruksi program. Pin data yang digunakan untuk berkomunikasi hanya menggunakan 4 pin data saja, yaitu D4-D7.

## Soal Formatif

---

1. Jelaskan pengertian LCD?
2. Jelaskan perbedaan LCD teks dan LCD grafik?
3. Jelaskan fungsi setiap pin yang ada pada LCD?

## **Tugas Latihan**

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 1 buah LCD, dan 1 buah pushbutton. Buatlah dengan algoritma, jika tombol ditekan akan tampil tulisan WELCOME di LCD, dan jika ditekan lagi untuk kedua kalinya akan muncul NAMA anda, dan terakhir jika ditekan lagi akan muncul TANGGAL LAHIR anda.



## Bab 5

# Interface Input Analog (ADC)

### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan pengertian data Analog.
2. Peserta didik mampu menjelaskan fitur ADC.
3. Peserta didik mampu mengimplementasikan fungsi antarmuka ADC pada mikrokontroler AVR dan Arduino.
4. Peserta didik mampu membuat rangkaian dan pemrograman antarmuka ADC pada mikrokontroler AVR dan Arduino.

## 5.1 Data Analog

Untuk menyatakan besaran analog kita membutuhkan besaran persamaan (analogi). Misalnya pada hitungan analog menunjukkan bilangan 1 maka pada besaran tegangan menyatakan 1 volt, untuk bilangan 2 menyatakan tegangan 2 volt, untuk bilangan 15,75 menyatakan 15,74 volt dan seterusnya. Pada contoh diatas antara besar bilangan dan besaran tegangan yang dinyatakan mempunyai nilai kesepadanan. Perubahan nilai bilangan baik naik maupun turun akan selalu menunjukkan nilai yang sepadan dengan tegangan.

Data analog adalah sinyal data dalam bentuk gelombang yang yang kontinyu, yang membawa informasi dengan mengubah karakteristik gelombang. Dua parameter/karakteristik terpenting yang dimiliki oleh isyarat analog adalah amplitudo dan frekuensi. Isyarat analog biasanya dinyatakan dengan gelombang sinus atau setengah lingkaran, mengingat gelombang sinus merupakan dasar untuk semua bentuk isyarat analog. Hal ini didasarkan kenyataan bahwa berdasarkan analisis fourier, suatu sinyal analog dapat diperoleh dari perpaduan sejumlah gelombang sinus.

Dengan menggunakan sinyal analog, maka jangkauan transmisi data dapat mencapai jarak yang jauh, tetapi sinyal ini mudah terpengaruh oleh *noise*. Gelombang pada sinyal analog yang umumnya berbentuk gelombang sinus memiliki tiga variable dasar, yaitu amplitudo, frekuensi dan phase.

- Amplitudo merupakan ukuran tinggi rendahnya tegangan dari sinyal analog.
- Frekuensi adalah jumlah gelombang sinyal analog dalam satuan detik.
- Phase adalah besar sudut dari sinyal analog pada saat tertentu.

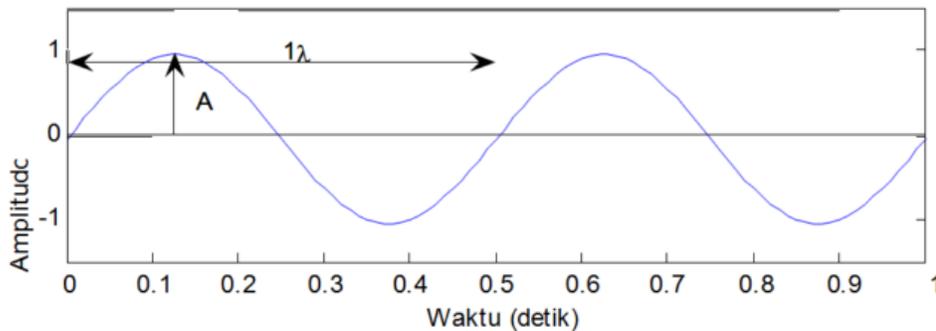
Data Analog disebarluaskan melalui gelombang elektromagnetik (gelombang radio) secara terus menerus, yang banyak dipengaruhi oleh faktor pengganggu. Analog merupakan bentuk komunikasi elektromagnetik yang merupakan proses pengiriman sinyal pada gelombang elektromagnetik dan bersifat variable yang berurutan. Jadi sistem analog merupakan suatu bentuk sistem komunikasi elektromagnetik yang menggantungkan proses pengiriman sinyalnya pada gelombang elektromagnetik.

Kecepatan gelombang ini disebut dengan Hertz (Hz) yang diukur dalam satuan detik. Misal dalam satu detik gelombang dikirim sebanyak 1000, maka disebut dengan 1000 Hertz. Kekurangan sistem analog ini adalah pengiriman sinyal agak lambat dan sering terjadi error. Hal-hal seperti ini

tidak terjadi pada sistem digital. Oleh karenanya saat ini banyak peralatan maupun aplikasi yang beralih dari sistem analog menjadi sistem digital.

## 5.2 Sinyal Analog

Bentuk sinyal analog yang paling sederhana dapat digambarkan sebagai gelombang sinus. Namun dalam keadaan nyata suatu sinyal analog merupakan gabungan dari beberapa gelombang sinus yang disebut dengan sinyal komposit. Dengan teknik yang ditemukan oleh seorang ilmuwan Perancis bernama Jean-Baptiste Fourier sinyal komposit dapat didekomposisi ke dalam beberapa gelombang sinus untuk kepentingan analisis. Teknik ini disebut dengan analisis Fourier. Gelombang sinus memiliki beberapa properti penting yang akan segera kita bahas, yaitu amplitudo, frekuensi, periode, fasa, dan panjang gelombang.



Gambar 5.1: Sinyal dalam bentuk gelombang sinus

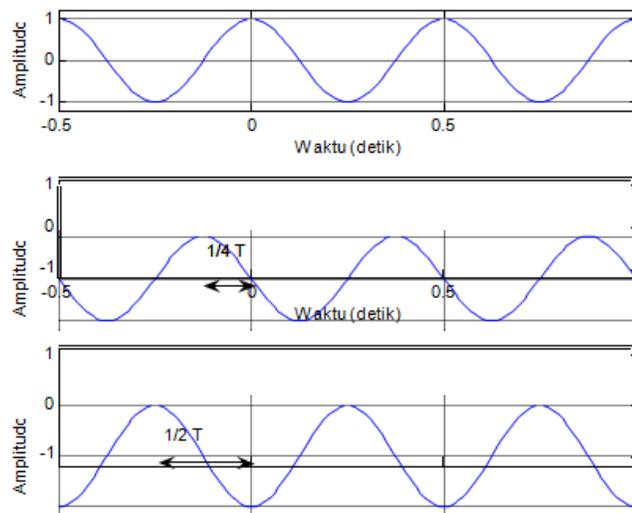
Amplitudo adalah suatu nilai yang merujuk pada ketinggian intensitas sinyal pada setiap waktu. Intensitas sinyal yang tertinggi disebut dengan amplitudo puncak. Intensitas sinyal ini berkaitan dengan jumlah energi yang dibawa oleh gelombang tersebut. Sebagai contoh pada sinyal listrik, amplitudo diukur dengan satuan volt.

Frekuensi dinyatakan sebagai jumlah periode yang dilalui oleh satu gelombang dalam waktu 1 detik. Dalam Gambar 5.1 terlihat bahwa dalam 1 detik gelombang melalui 2 siklus, karena itu gelombang dalam gambar 2.3 memiliki frekuensi = 2 siklus/detik (atau 2 Hertz). Frekuensi juga dapat dinyatakan sebagai jumlah perubahan per satuan waktu. Apabila suatu sinyal memiliki jumlah perubahan banyak sekali maka kita katakan sinyal tersebut memiliki frekuensi tinggi, sebaliknya apabila suatu sinyal memiliki

jumlah perubahan sedikit sekali maka kita katakan sinyal tersebut memiliki frekuensi rendah. Apabila suatu sinyal berubah secara instan (tiba-tiba berubah) maka sinyal tersebut memiliki frekuensi tak terhingga. Apabila suatu sinyal tidak berubah sama sekali maka sinyal tersebut memiliki frekuensi nol. Misalnya, sinyal direct current (DC) yang dikeluarkan oleh sebuah baterai akan menghasilkan sinyal sebesar 1.5 volt terus menerus, karena itu frekuensi dari sinyal DC adalah nol.

Periode adalah waktu yang dibutuhkan untuk menempuh 1 siklus gelombang. Dalam Gambar 5.1, satu siklus gelombang ditempuh dalam waktu 0,5 detik. Karena itu periode dari gelombang adalah 0,5 detik. Yang mana  $f$  adalah frekuensi dalam satuan Hertz atau siklus/detik dan  $T$  adalah periode dalam satuan detik.

Panjang gelombang adalah jarak yang dilalui untuk menempuh satu siklus gelombang dalam satuan meter. Properti terakhir yang akan kita bahas adalah fasa. Fasa yang diukur dalam satuan derajat atau radian merupakan jarak pergeseran sinyal relatif terhadap titik 0. Apabila fasa bernilai positif, maka sinyal bergeser ke kiri relatif terhadap titik 0. Sebaliknya apabila fasa bernilai negatif, maka sinyal bergeser ke kanan relatif terhadap titik 0.



Gambar 5.2: Gelombang sinus dengan pergeseran fasa

## 5.3 Analog to Digital Converter

### 5.3.1 Pengertian ADC

ADC (Analog To Digital Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinyu) menjadi sinyal digital. Perangkat ADC (Analog To Digital Conversion) dapat berbentuk suatu modul atau rangkaian elektronika maupun suatu chip IC. ADC (Analog To Digital Converter) berfungsi untuk menjembatani pemrosesan sinyal analog oleh sistem digital.

Alat bantu digital yang paling penting untuk teknologi kontrol proses adalah yang menerjemahkan informasi digital ke bentuk analog dan juga sebaliknya. Sebagian besar pengukuran variabel-variabel dinamik dilakukan oleh piranti ini yang menerjemahkan informasi mengenai variabel ke bentuk sinyal listrik analog. Untuk menghubungkan sinyal ini dengan sebuah komputer atau rangkaian logika digital, sangat perlu untuk terlebih dahulu melakukan konversi analog ke digital (A/D). Hal-hal mengenai konversi ini harus diketahui sehingga ada keunikan, hubungan khusus antara sinyal analog dan digital.

Analog To Digital Converter (ADC) adalah pengubah input analog menjadi kode kode digital. ADC banyak digunakan sebagai Pengatur proses industri, komunikasi digital dan rangkaian pengukuran/ pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistim komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistim digital (komputer). ADC (Analog to Digital Converter) memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi.

**Kecepatan sampling** suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam sample per second (SPS).

**Resolusi ADC** menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ( $2^n - 1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit.

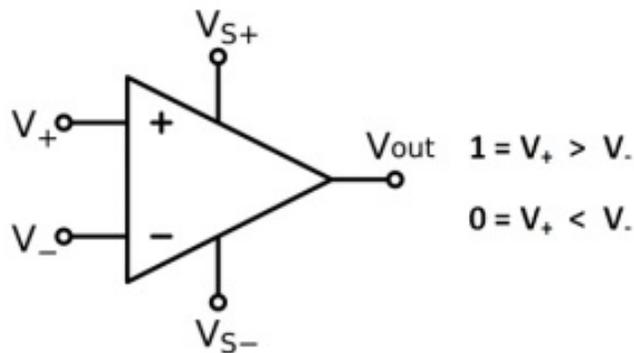
### 5.3.2 Prinsip Kerja ADC

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar  $60\% \times 255 = 153$  (bentuk decimal) atau 10011001 (bentuk biner).

$$\begin{aligned} \text{signal} &= (\text{sample}/\text{max\_value}) * \text{reference\_voltage} \\ &= (153/255) * 5 \\ &= 3 \text{ Volts} \end{aligned}$$

### 5.3.3 Komparator ADC

Bentuk komunikasi yang paling mendasar antara wujud digital dan analog adalah piranti (biasanya berupa IC) disebut komparator. Piranti ini, yang diperlihatkan secara skematik pada gambar dibawah, secara sederhana membandingkan dua tegangan pada kedua terminal inputnya. Bergantung pada tegangan mana yang lebih besar, outputnya akan berupa sinyal digital 1 (high) atau 0 (low). Komparator ini digunakan secara luas untuk sinyal alarm ke komputer atau sistem pemroses digital. Elemen ini juga merupakan satu bagian dengan konverter analog ke digital dan digital ke analog yang akan didiskusikan nanti. Gambar diatas memperlihatkan sebuah kompara-



Gambar 5.3: Konsep Komparator Pada ADC (Analog to Digital Converter)

tor merubah keadaan logika output sesuai fungsi tegangan input analog. Sebuah komparator dapat tersusun dari sebuah opamp yang memberikan output terpotong untuk menghasilkan level yang diinginkan untuk kondisi

logika (+5 dan 0 untuk TTL 1 dan 0). Komparator komersil didesain untuk memiliki level logika yang diperlukan pada bagian outputnya.

### 5.3.4 Jenis-Jenis ADC

**ADC Simultan** atau biasa disebut flash converter atau parallel converter. Input analog  $V_i$  yang akan diubah ke bentuk digital diberikan secara simultan pada sisi + pada komparator tersebut, dan input pada sisi tergantung pada ukuran bit converter. Ketika  $V_i$  melebihi tegangan input dari suatu komparator, maka output komparator adalah high, sebaliknya akan memberikan output low.

**Counter Ramp ADC** didalamnya terdapat DAC yang diberi masukan dari counter, masukan counter dari sumber Clock dimana sumber Clock dikontrol dengan cara meng AND kan dengan keluaran Comparator. Comparator membandingkan antara tegangan masukan analog dengan tegangan keluaran DAC, apabila tegangan masukan yang akan dikonversi belum sama dengan tegangan keluaran dari DAC maka keluaran comparator = 1 sehingga Clock dapat memberi masukan counter dan hitungan counter naik.

**ADC SAR** Yaitu dengan memakai konfigurasi yang hampir sama dengan counter ramp tetapi dalam melakukan trace dengan cara tracking dengan mengeluarkan kombinasi bit MSB = 1 ==> 1000 0000. Apabila belum sama (kurang dari tegangan analog input maka bit MSB berikutnya = 1 ==> 1100 0000) dan apabila tegangan analog input ternyata lebih kecil dari tegangan yang dihasilkan DAC maka langkah berikutnya menurunkan kombinasi bit ==> 10100000.

## 5.4 Komponen Antar Muka ADC

Penggunaan data analog terkhusus pada pembacaan ADC lebih banyak digunakan pada perangkat interface yang berupa sensor yang memiliki kondisi lebih dari 2 keadaan standar. Disinilah titik bedanya dengan pembacaan digital, pembacaan digital hanya mengenal ada atau tidak ada, sedangkan pembacaan analog mampu mengenal sampai ribuan keadaan yang berbeda seperti tidak ada, ada, setengah, seperempat, dan seterusnya. Beberapa sensor yang menggunakan data ADC:

### 5.4.1 Sensor Cahaya

Sensor cahaya adalah komponen elektronika yang merubah besaran fisik dalam hal ini cahaya menjadi besaran elektrik. Contoh sensor cahaya yang biasa digunakan adalah Photodiode / LDR (Light Defendent Resistor). Sensor ini bisa menggunakan output digital dengan syarat harus melalui komparator terlebih dahulu sehingga keluarannya hanya ada dua keadaan yaitu terang atau gelap.



Gambar 5.4: Light Defendent Resistor

Penggunaan data analog untuk mengelolah data dari sensor cahaya dapat memungkinkan pengguna dapat mendapatkan informasi lebih mendetail seperti, gelap sekali, gelap, terang, terang sekali, redup, senja, dan seterusnya.



Gambar 5.5: photodiode

Photodiode dirangkai dengan cara menyambungkan kaki katode ke pin data mikrokontroler dan menambahkan resistor pullup ke vcc 5v, dan kaki anode disambungkan ke groud. Rangkaian LDR dirangkai sama dengan

photo diode, dikarenakan LDR tidak berpolaritas maka pemasngannya bisa dibolak balik. Dalam rangkaian umum penggunaan sensor harus menambahkan resistor sebagai pullup untuk membantu dalam pengambilan data. Rangkaian yang menggunakan pullup dapat memperbesar rentang nilai pembacaan sensor, sehingga data mudah dibaca.

### 5.4.2 Sensor Suhu

Sensor suhu adalah komponen elektronika baik aktif maupun pasif yang dapat merespon perubahan temperature atau suhu disekitar komponen tersebut dan menghasilkan perubahan elektrik sesuai dengan perubahan suhu atau temperature yang direspon komponen tersebut. Sensor suhu banyak digunakan dalam kehidupan kita, sebagai contoh alat yang menggunakan sensor suhu adalah termometer digital.

Sensor suhu dibagi dalam 4 golongan utama, dari tiap jenis sensor suhu ini memiliki beberapa tipe dan bentuk yang berbeda. Berikut adalah 4 jenis utama sensor suhu.

**Thermocouple** pada intinya terdiri dari sepasang transduser panas dan dingin yang disambungkan dan dilebur bersama, dimana terdapat perbedaan yang timbul antara sambungan tersebut dengan sambungan referensi yang berfungsi sebagai pembanding.

**Resistance Temperature Detector (RTD)** memiliki prinsip dasar pada tahanan listrik dari logam yang bervariasi sebanding dengan suhu. Kesebandingan variasi ini adalah presisi dengan tingkat konsisten/kestabilan yang tinggi pada pendeteksian tahanan. Platina adalah bahan yang sering digunakan karena memiliki tahanan suhu, kelinearan, stabilitas dan reproduksibilitas.

**Termistor** adalah resistor yang peka terhadap panas yang biasanya mempunyai koefisien suhu negatif, karena saat suhu meningkat maka tahanan menurun atau sebaliknya. Jenis ini sangat peka dengan perubahan tahan 5% per C sehingga mampu mendeteksi perubahan suhu yang kecil.

**IC Sensor** adalah sensor suhu dengan rangkaian terpadu yang menggunakan chipsilikon untuk kelemahan penginderanya. Mempunyai konfigurasi output tegangan dan arus yang sangat linear.

Salah satu jenis IC sensor suhu adalah IC sensor suhu tipe LM35. IC sensor suhu LM 35 ini memiliki output yang linier dan bekerja dengan tegangan



darah orang dewasa secara otomatis. Alat tersebut dilakukan dengan manset yang dipasang di lengan pasien, kemudian dipompa sampai pada tekanan tertentu yang selanjutnya baru dilakukan pengukuran tekanan darah.

#### 5.4.4 Sensor Kecepatan

Proses penginderaan sensor kecepatan merupakan proses kebalikan dari suatu motor, dimana suatu poros/object yang berputar pada suatu generator akan menghasilkan suatu tegangan yang sebanding dengan kecepatan putaran object. Kecepatan putar sering pula diukur dengan menggunakan sensor yang mengindera pulsa magnetis (induksi) yang timbul saat medan magnetis terjadi. Contohnya pada alat pengukur kecepatan speedometer. Alat tersebut mengukur kecepatan laju motor dalam kilometer perjam.

#### 5.4.5 Sensor Magnet

Sensor Magnet atau disebut juga relai buluh, adalah alat yang akan terpengaruh medan magnet dan akan memberikan perubahan kondisi pada keluaran. Seperti layaknya saklar dua kondisi (on/off) yang digerakkan oleh adanya medan magnet di sekitarnya. Biasanya sensor ini dikemas dalam bentuk kemasan yang hampa dan bebas dari debu, kelembapan, asap ataupun uap. Implementasi dari alat ini seperti, Pengukuran medan magnet berbasis komputer terdiri dari sensor medan magnet UGN3503, Op-Amp LM358 dan ADC 0804. Prinsip kerja alat adalah mendekatkan magnet pada sensor. Keluaran sensor berupa tegangan akan dikuatkan oleh op-amp agar dapat diproses oleh ADC.

#### 5.4.6 Sensor Penyandi

Sensor Penyandi (Encoder) digunakan untuk mengubah gerakan linear atau putaran menjadi sinyal digital, dimana sensor putaran memonitor gerakan putar dari suatu alat. Sensor ini biasanya terdiri dari 2 lapis jenis penyandi, yaitu; Pertama, Penyandi rotari tambahan (yang mentransmisikan jumlah tertentu dari pulsa untuk masing-masing putaran) yang akan membangkitkan gelombang kotak pada objek yang diputar. Kedua, Penyandi absolut (yang memperlengkapi kode binary tertentu untuk masing-masing posisi sudut) mempunyai cara kerja sang sama dengan perkecualian, lebih banyak atau lebih rapat pulsa gelombang kotak yang dihasilkan sehingga membentuk suatu pengkodean dalam susunan tertentu. Contoh pengimplementasiannya yaitu sensor ini dapat dibuat menjadi suatu sistem yang dapat

menghitung kekuatan gempa bumi dengan menggunakan sensor incremental rotary encoder dan diolah oleh mikrokontroler.

#### 5.4.7 Flow Meter Sensor

Flow Meter merupakan Sensor yang digunakan untuk mengetahui flow dari suatu material baik solid maupun liquid. Di Dunia Industri terdapat macam-macam jenis dari Sensor Flow ini. Untuk Yang Liquid biasanya menggunakan jenis Turbin, Elektromagnetic, VenturiMeter dan lain-lain. Sedangkan untuk Solid material biasanya digunakan dari kombinasi beberapa peralatan instrument yang dijadikan Flow Meter, contohnya Weigh Feeder.

#### 5.4.8 Flame Sensor

Flame sensor ini dapat mendeteksi nyala api dengan panjang gelombang 760 nm – 1100 nm. Dalam banyak pertandingan robot, pendeteksian nyala api menjadi salah satu aturan umum perlombaan yang tidak pernah ketinggalan. Oleh sebab itu sensor ini sangat berguna, yang dapat Anda jadikan 'mata' bagi robot untuk dapat mendeteksi sumber nyala api, atau mencari bola. Cocok digunakan pada robot fire-fighting dan soccer robot. Sensor nyala api ini mempunyai sudut pembacaan 60 derajat, dan beroperasi pada suhu 25 -85 derajat Celcius. Dan tentu saja untuk Anda perhatikan, bahwa jarak pembacaan antara sensor dan objek yang dideteksi tidak boleh terlalu dekat, untuk menghindari kerusakan sensor.

### 5.5 Interface ADC di Mikrokontroler AVR

Pembacaan ADC pada mikrokontroler jenis AVR pada umumnya mempunyai resolusi pembacaan 8bit atau 10 bit, artinya nilai pembacaa analog dari sensor akan dirubah menjadi nilai digital 8bit(256 keadaan) atau 10bit(1024 keadaan), berikut adalah rumus mencari nilai ADC

$$ADC = \frac{V_{in} \times DigitalBitRate}{V_{ref}} \quad (5.1)$$

Keterangan:

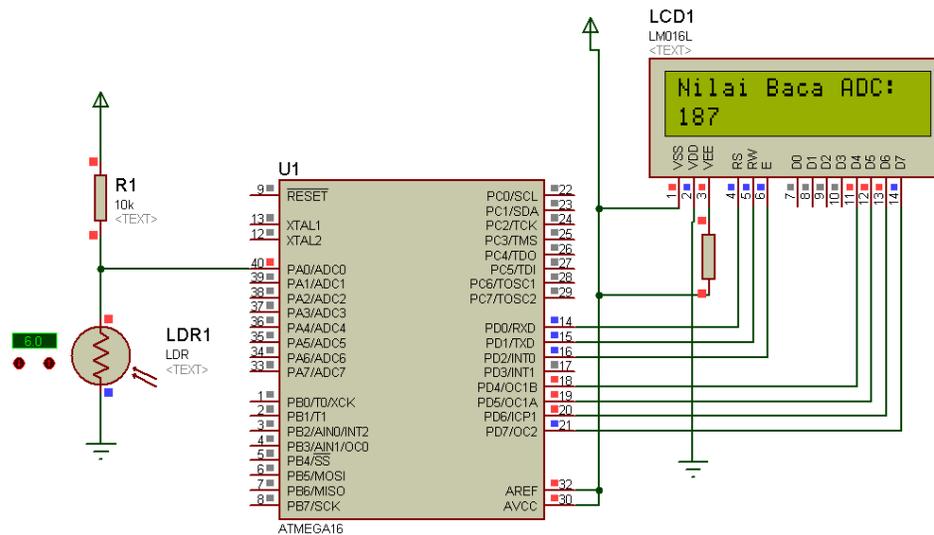
- ADC: analog to digital konverter
- $V_{in}$  : Hasil Ukuran tegangan pada sensor

- digital bit rate : resolusi bit bisa menggunakan 8bit atau 10bit.
- Vreff: tegangan referensi yang akan digunakan untuk mengelolah ADC biasanya menggunakan tegangan 5v

Berikut adalah contoh rangkaian sederhana dan pemrograman interface ADC menggunakan Mikrokontroler AVR ATmega16.

### 5.5.1 Rangkaian

Berikut adalah contoh rangkaian mikrokontroler AVR ATmega16 yang mrrgunakan interface ADC sensor Cahaya Photodiode ditambahkan dengan interface LCD untuk menampilkan nilai digital atau hasil conversi dari tegangan analog.

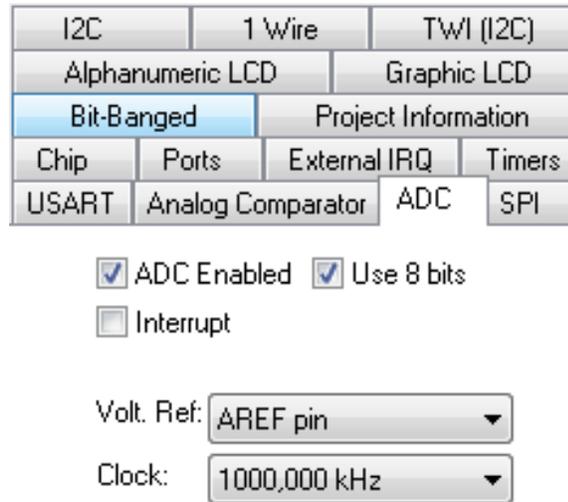


Gambar 5.8: Interface ADC dengan LCD

Pada mikrokontroler AVR untuk dapat berkomunikasi menggunakan antarmuka input analog harus menggunakan kaki pin ADC. Beberapa mikrokontroler keluaran Atmel masih belum menggunakan fitur pin ini jadi pada penggunaan mikrokontroler yang tidak memiliki pin ini harus menggunakan rangkaian pembanding terlebih dahulu (komparator ADC). Adapun contoh mikrokontroler keluaran Atmel seri AVR yang telah menyediakan fitur pin ADC adalah mikrokontroler AVR ATmega8538, ATmega8, ATmega16, ATmega32 dan ATmega128.

### 5.5.2 Pemrograman

Berdasarkan dari rangkaian diatas, dapat diatur pada aplikasi codevisionAVR dengan mengaktifkan fitur ADC dan menggunakan rate 8 bit pembacaan.



Gambar 5.9: pengaturan untuk mengaktifkan ADC 8 bit

Setelah melakukan pengaturan seperti gambar 5.9, maka pada source code akan langsung otomatis tersedia perintah untuk membaca data ADC. Perintah dalam pembacaan ADC adalah "read\_adc(pin);" contoh "read\_adc(0)" artinya kita akan mengelolah data sensor yang terhubung pada pin ADC.0. Berikut source code lengkap untuk membaca data ADC dan menampilkannya pada LCD.

## Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

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

#define ADC_VREF_TYPE 0x20

char tampil[16];

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

void main(void)
{
    ACSR=0x80;
    SFIOR=0x00;
    ADMUX=ADC_VREF_TYPE & 0xff;
    ADCSRA=0x84;

    lcd_init(16);

    while (1)
    {
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_puts("Nilai Baca ADC:");
        lcd_gotoxy(0,1);
        sprintf(tampil,"%i",read_adc(0));
        lcd_puts(tampil);
        delay_ms(100);
    }
}

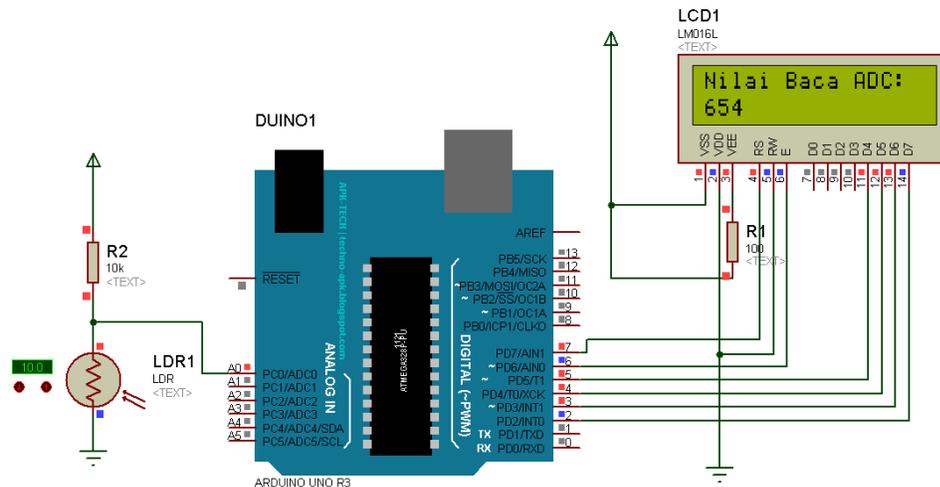
```

## 5.6 Interface ADC di Arduino

Pembacaan ADC pada Arduino umumnya menggunakan resolusi pembacaan 10 bit (1024 keadaan/kondisi).

### 5.6.1 Rangkaian

Berikut adalah contoh rangkaian Arduino UNO yang menggunakan interface ADC sensor Cahaya ditambahkan dengan Interface LCD untuk menampilkan nilai digital atau nilai hasil konversi dari tegangan analog pada sensor.



Gambar 5.10: Interface ADC dengan LCD

### 5.6.2 pemrograman

pada program arduino sudah otomatis menyediakan fitur yang membaca data analog (ADC), fitur tersebut adalah "analog.Read(pin);" contoh "analog.Read(A0);" artinya program akan membaca data sensor yang terhubung pada pin A0. Berikut source code lengkap untuk membaca data ADC dan menampilkannya di LCD.

## Listing Program:

```
adc
// Memanggil Librari
#include <LiquidCrystal.h>

// Pengaturan PIN Interface LCD
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);           // menggunakan LCD 16x2
  pinMode(A0, INPUT);
}

void loop() {
  lcd.setCursor(0,0);
  lcd.print("Nilai Baca ADC:");
  lcd.setCursor(0,1);
  //membca data dari PIN A0 dan menampilkannya di LCD:
  lcd.print(analogRead(A0));
}
```

## Rangkuman

---

1. Data analog adalah sinyal data dalam bentuk gelombang yang kontinyu, yang membawa informasi dengan mengubah karakteristik gelombang
2. Gelombang pada sinyal analog yang umumnya berbentuk gelombang sinus memiliki tiga variable dasar, yaitu amplitudo, frekuensi dan phase.
3. ADC (Analog To Digital Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinyu) menjadi sinyal digital
4. Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal input dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan input 3 volt, rasio input terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar  $60\% \times 255 = 153$  (bentuk decimal) atau 10011001 (bentuk biner)
5. Jenis ADC ada tiga yaitu ADC Simultan, Counter Ramp ADC, dan ADC SAR.
6. Penggunaan data analog terkhusus pada pembacaan ADC lebih banyak digunakan pada perangkat interface yang berupa sensor yang memiliki kondisi lebih dari 2 keadaan standar seperti sensor cahaya, suhu, tekanan, kecepatan, magnet, panyandi, aliran, flame.
7. ADC pada mikrokontroler AVR maupun Arduino menggunakan resolusi pembacaan 8 bit atau 10 bit.
8. Pemrograman ADC pada mikrokontroler AVR menggunakan perintah `read_adc(pin);`, sedangkan pada Arduino menggunakan perintah `analoRead(pin);`

## Soal Formatif

---

1. Jelaskan pengertian data Analog?
2. Jelaskan pengertian ADC?
3. Jelaskan komponen apa saja yang mampu dikelola menggunakan ADC?
4. Jelaskan perangkat yang dapat diimplementasikan menggunakan data ADC menggunakan mikrokontroler AVR maupun Arduino?

## Tugas Latihan

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 1 buah LCD, dan 1 buah sensor warna menggunakan Photodiode. Buatlah dengan algoritma, jika benda warna merah didekatkan pada sensor maka akan muncul tulisan pada LCD "MERAH", begitupun untuk warna lain, buatlah agar alat mengenali minimal 3 warna.



## Bab 6

# Interface Output PWM

### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan pengertian Timer/Counter.
2. Peserta didik mampu menjelaskan pengertian dan prinsip kerja PWM.
3. Peserta didik mampu mengimplementasikan fungsi antarmuka PWM pada mikrokontroler AVR dan Arduino.
4. Peserta didik mampu membuat rangkaian dan pemrograman antarmuka PWM pada mikrokontroler AVR dan Arduino

## 6.1 Timer/Counter

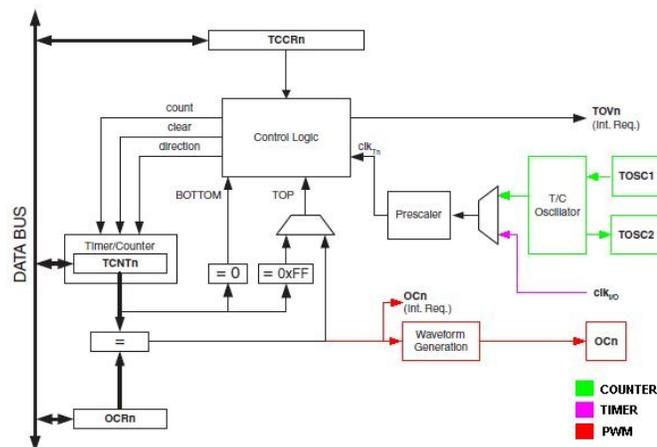
Timer/Counter adalah Seperangkat pencacah biner (binary counter) yang terhubung langsung ke saluran data mikrokontroler, sehingga mikrokontroler bisa membaca kondisi pencacah dan bila diperlukan mikrokontroler dapat pula merubah kondisi pencacah tersebut. Saat sinyal clock yang diberikan sudah melebihi kapasitas pencacah, maka pencacah akan memberikan sinyal overflow/limpahan. Limpahan pencacah ini dicatat dalam suatu register.

*Timer* pada dasarnya hanya menghitung pulsa *clock*. Frekuensi pulsa *clock* yang dihitung tersebut sama dengan frekuensi kristal yang dipasang atau dapat diperlambat menggunakan *prescaler* dengan faktor 8, 64, 256 atau 1024.

Sinyal clock yang diberikan ke pencacah dibedakan menjadi 2 macam :

1. Sinyal *Clock* dengan frekuensi tetap yang sudah diketahui besarnya pencacah bekerja sebagai Timer (Pewaktu), karena kondisi pencacah tsb setara dengan waktu yang bisa ditentukan secara pasti.
2. Sinyal *Clock* dengan frekuensi yang bisa bervariasi pencacah bekerja sebagai Counter (Pencacah), kondisi pencacah tersebut menyatakan banyaknya pulsa clock yang sudah diterima.

### 6.1.1 Timer / Counter 8 Bit



Gambar 6.1: Blok diagram Timer / Counter 8 bit

Timer/counter 0 dan timer/counter 2 adalah timer/counter 8 bit yang mempunyai multifungsi. fitur-fiturnya yaitu:

1. Counter satu kanal
2. Timer dinolkan saat match compare (autoreload)
3. Ghitch-free, phase correct pulse width modulator (PWM)
4. Frekuensi generator
5. 10 bit clock prescaler
6. Interupsi timer yang disebabkan timer overflow (TOV<sub>n</sub>) dan compare match (OCF<sub>n</sub>)

Block diagram dari timer/counter n ditunjukkan pada gambar 6.1, n adalah nomor yang mewakili timer/counter 8 bit ( n = 0,2 ).

Register-register Timer 8 bit:

1. Register TCNT (Timer/Counter Register) berfungsi sebagai register pencacah dari 0 sampai nilai maximum yang kita tentukan.
2. Register TCCR (Timer/Counter Control Register) berfungsi untuk pengaturan mode operasi timer/Counter.
3. Register TIMSK (Timer/Counter Interrupt Mask Register) berfungsi untuk memilih Timer/Counter mana yang aktif.
4. Register TIFR (Timer/Counter Interrupt Flag Register) berfungsi untuk mengetahui adanya interupsi akibat operasi Timer/Counter.
5. Register OC (Output Compare) berfungsi untuk menyimpan nilai pembandingan dengan nilai pada register TCNT.

Timer/Counter 8 bit dapat menghitung maksimal hingga 255 (00 FF ) hitungan, dimana periode setiap hitungan (*clock*-nya) tergantung dari setting *prescaler*-nya. Untuk mengatur jenis mode operasi dari timer/counter dan mengatur *prescaler* digunakan register timer/counter control register TCCR<sub>n</sub> (n=0,2). Mode-mode operasi timer:

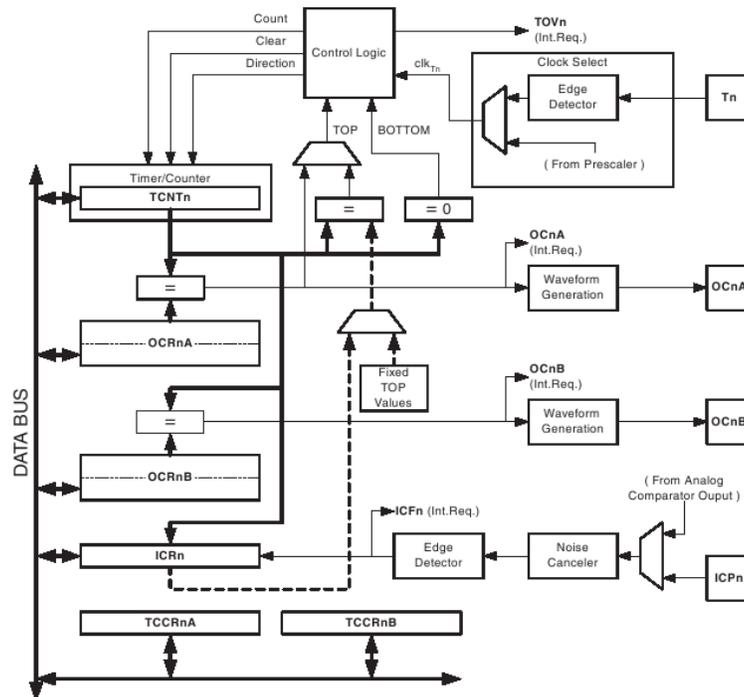
**Mode Normal** Timer digunakan untuk menghitung saja, membuat delay, menghitung selang waktu.

**Mode PWM, phase correct** Memberikan bentuk gelombang phase correct PWM resolusi tinggi. Mode phase correct PWM berdasarkan operasi dualslope. Counter menghitung berulang-ulang dari BOTTOM ke MAX dan dari MAX ke BOTTOM.

**CTC (clear timer on compare match)** Pada mode CTC, nilai timer yang ada pada TCNTn akan dinolkan lagi jika TCNTn sudah sama dengan nilai yang ada pada register OCRn, sebelumnya OCR diset dulu, karena timer 0 dan 2 maksimumnya 255, maka range OCR 0-255.

**Fast PWM** Memberikan pulsa PWM frekuensi tinggi. Fast PWM berbeda dengan mode PWM lain, fast PWM berdasarkan operasi single slope. Counter menghitung dari BOTTOM hingga TP kemudian kembali lagi mulai menghitung berawal dari BOTTOM.

### 6.1.2 Timer / Counter 16 Bit



Gambar 6.2: Blok diagram Timer / Counter 16 bit

Pada mode normal, TCNT1 akan menghitung naik dan membangkitkan interrupt Timer/Counter 1 ketika nilainya berubah dari 0xFFFF ke 0x0000. Untuk menggunakan timer yang menghitung mundur cukup dengan memasukkan nilai yang diinginkan ke TCNT1 dan menunggu sampai terjadi interrupt, tetapi untuk timer yang menghitung maju, maka nilai yang dimasukkan ke dalam TCNT1 nilainya harus 65536 (*timer value*).

Berikut register-register penting dalam mengaktifkan Timer 16 bit.

**TCNT1** Timer/Counter register digunakan untuk menyimpan nilai timer yang diinginkan. TCNT1 dibagi menjadi 2 register 8 bit yaitu TCNT1H dan TCNT1L.

**TIMSK & TIFR** Timer Interrupt Mask Register (TIMSK) dan Timer Interrupt Flag (TIFR) register digunakan untuk mengendalikan interrupt mana yang diaktifkan dengan cara melakukan pengaturan pada TIMSK dan untuk mengetahui interrupt mana yang sedang terjadi.

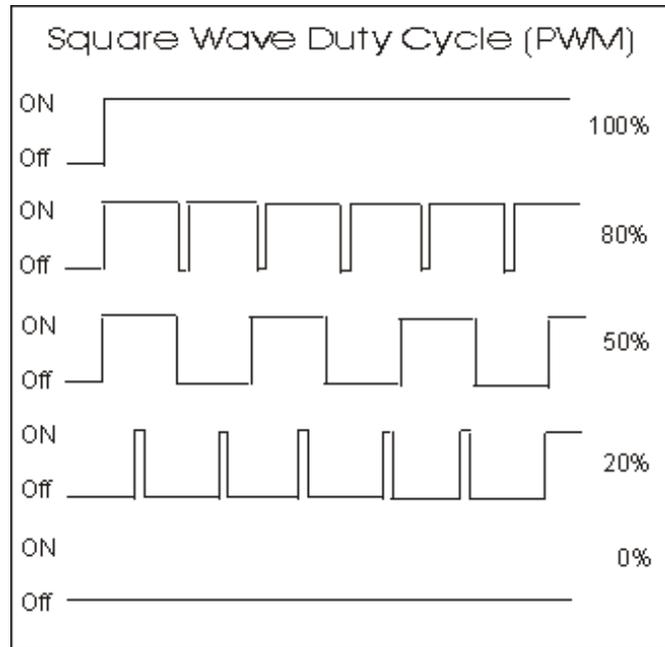
## 6.2 PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*duty cycle*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi high kemudian berada di zona transisi ke kondisi low. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. *Duty Cycle* merupakan representasi dari kondisi logika high dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan range 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi high terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan *high* sama dengan keadaan *low* maka sinyal mempunyai *duty cycle* sebesar 50%.

Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai *duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cycle*-nya kecil maka motor akan bergerak lambat.

Untuk membandingkannya terhadap tegangan DC, PWM memiliki 3 mode operasi yaitu :

**Inverted Mode** Pada mode inverted ini jika nilai sinyal lebih besar dari pada titik pembanding (*compare level*) maka output akan di set high



Gambar 6.3: bentuk Sinyal PWM

(5v) dan sebaliknya jika nilai sinyal lebih kecil maka output akan di set low (0v) seperti pada gelombang A pada gambar di atas.

**Non Inverted Mode** Pada mode non inverted ini output akan bernilai high (5v) jika titik pembanding (compare level) lebih besar dari pada nilai sinyal dan sebaliknya jika bernilai low (0v) pada saat titik pembanding lebih kecil dari nilai sinyal seperti pada gelombang B pada gambar di atas.

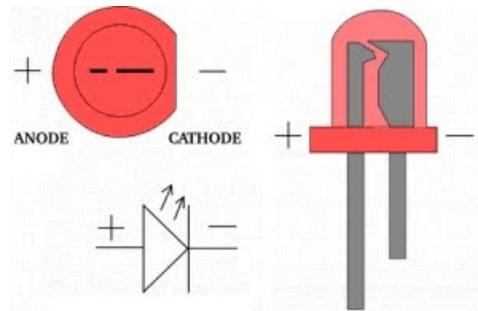
**Toggle Mode** Pada mode toggle output akan beralih dari nilai high (5v) ke nilai low (0v) jika titik pembanding sesuai dan sebaliknya beralih dari nilai low ke high.

### 6.3 Komponen dengan kontrol output PWM

komponen / peralatan yang dapat dikontrol menggunakan sinyal PWM adalah komponen yang memiliki banyak kondisi keluaran, berikut adalah daftar komponen yang dapat dikontrol dengan output PWM:

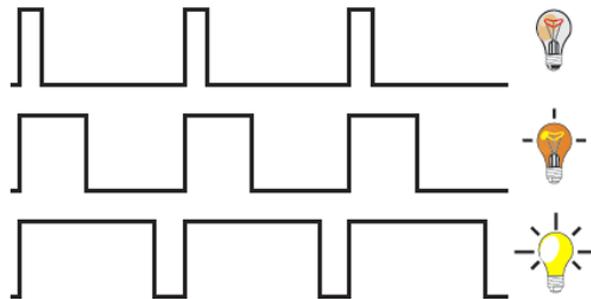
### 6.3.1 LED

Komponen LED ini telah dibahas sebelumnya pada BAB 3 tentang interface data digital. Light Emitting Diode (LED) adalah komponen elektronika yang dapat memancarkan cahaya. Sesuai dengan namanya, LED adalah salah satu jenis diode. Sebagaimana yang diketahui, diode adalah komponen yang hanya dapat mengalirkan arus listrik satu arah. Arus listrik hanya mengalir kalau tegangan positif dihubungkan ke kaki yang disebut anode dan tegangan negatif dihubungkan ke kaki yang dinamakan katode.



Gambar 6.4: Led Emitting Diode

Pada pengaplikasian LED yang terkontrol lewat IC mikrokontroler, yang diperlukan hanya menghubungkan salah satu PIN dari mikrokontroler ke kaki anode LED, dan kaki Katode ke GND rangkaian. Terkhusus jika menggunakan sinyal PWM, maka kaki anode LED harus disambungkan ke pin PWM.

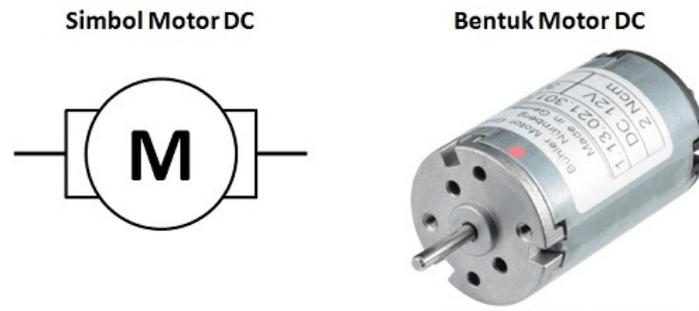


Gambar 6.5: Contoh Kondisi LED dengan output PWM

Selain dapat dikontrol dengan data Digital, komponen ini juga mampu dikontrol dengan output PWM. Perbedaannya adalah, jika menggunakan output digital maka LED hanya akan memiliki 2 kondisi yaitu menyala dan tidak menyala, sedangkan jika menggunakan output PWM maka LED dapat memiliki lebih dari 2 kondisi yaitu mati, menyala terang, menyala terang sekali, menyala redup, menyala redup sekali, dan seterusnya.

### 6.3.2 Motor DC

Motor atau lebih tepatnya motor listrik adalah komponen elektrik yang sifatnya merubah besaran elektrik menjadi fisik berupa putaran. Motor listrik terdiri atas tiga bagian utama yang disebut stator (bagian yang diam) dan rotor (bagian yang bergerak) dan air gap. Rotor dapat berputar karena adanya medan magnet yang dipengaruhi oleh arus listrik. Air gap adalah bagian yang memisahkan rotor dan stator. Motor terdiri dari dua jenis yaitu motor AC dan motor DC. Motor DC menggunakan 2 data yang harus tersambung ke data digital mikrokontroler namun ketika harus menggunakan motor DC yang membutuhkan arus tinggi harus menggunakan rangkaian penguat (driver) dari motor ke mikrokontroler. Sedangkan Motor AC biasanya memerlukan rangkian berupa relay untuk menjalankannya, karena tegangan yang berasal dari mikrokontroler adalah DC sedangkan tegangan yang dibutuhkan adalah tegangan AC.



Gambar 6.6: Motor DC

Penjelasan mengenai Motor hampir sama dengan LED, selain dapat dikontrol dengan data Digital, komponen ini juga mampu dikontrol dengan output PWM. Perbedaannya adalah, jika menggunakan output digital maka Motor hanya akan memiliki 2 kondisi yaitu diam dan berputar, sedangkan jika menggunakan output PWM maka motor dapat memiliki lebih dari

2 kondisi yaitu mati, pelansekali, pelan, normal, cepat, cepat sekali, dan seterusnya.

### 6.3.3 Motor Servo

Motor servo adalah sebuah aktuator yang bergerak dalam poros yang mempunyai spesifikasi untuk control posisi sudut yang presisi. Banyak jenis motor servo. Ada yang dikontrol secara serial ataupun dengan PWM. PWM harus diatur agar memenuhi standar sinyal input untuk motor servo sehingga motor servo dapat bergerak sesuai dengan perintah yang kita kirimkan lewat PWM.

Terdapat dua jenis tipe motor servo yaitu servo standard dan servo rotation (continuous). Dimana biasanya untuk tipe standar hanya dapat melakukan pergerakan sebesar 180 sedangkan untuk tipe continuous dapat melakukan rotasi atau 360. Contoh gambar di bawah ini adalah Servo dari Hitech dan isinya.

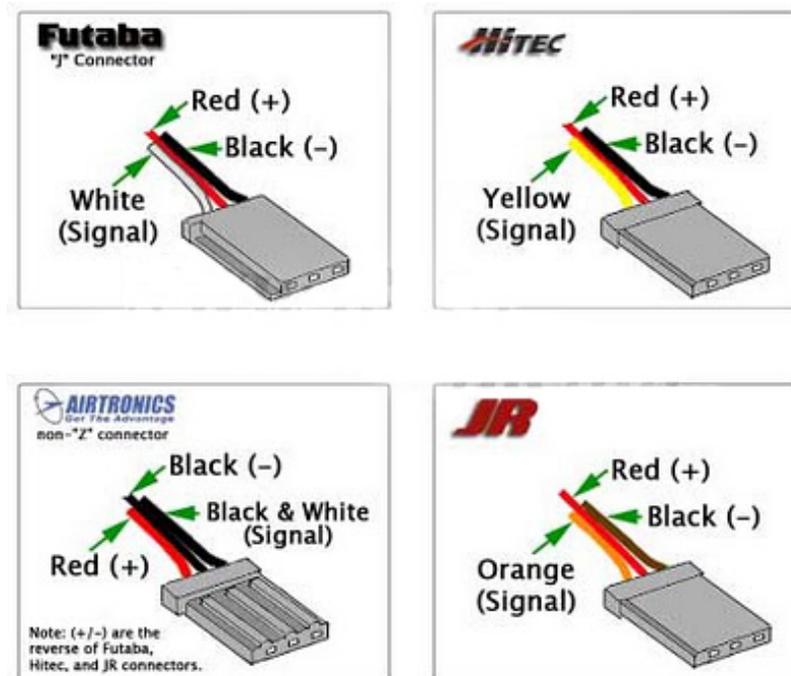


Gambar 6.7: Motor Servo

Didalam motor servo tersebut terdapat motor DC, komposisi gearbox dan rangkaian kontrolnya. Rangkaian kontrol pada motor servo digunak-

an untuk mengendalikan motor DC yang ada pada motor servo tersebut, oleh sebab itu untuk mengendalikan motor servo cukup hanya dengan cara memberikan pulsa-pulsa tertentu kepada rangkain kontrolernya. Gear-box pada motor servo berfungsi untuk meningkatkan torsi. Dikarenakan bentuknya yang compact motor servo sering digunakan di dunia robotika, aeromodeling, car remote controler dsb.

PIN pada motor servo yaitu mempunyai 3 buah pin yang terdiri dari VCC, GND dan SIGNAL/KONTROL. Berikut adalah konfigurasi pin motor servo berdasarkan merk.



Gambar 6.8: Data Pin Motor Servo

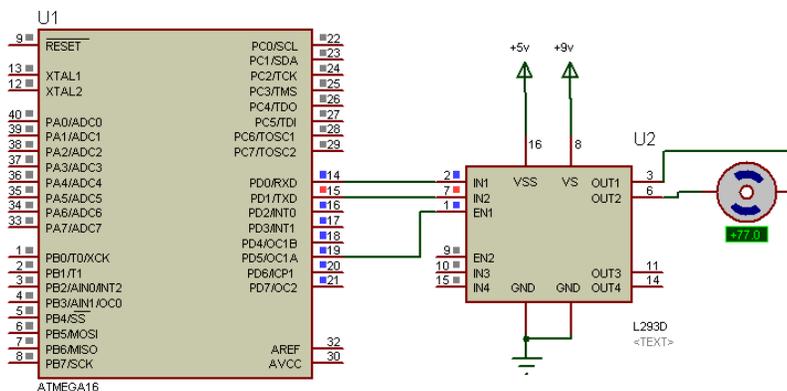
Timing penggunaan servo tersebut biasanya dijelaskan di datasheetnya tergantung dari produsen motor servo. Secara umum untuk mengakses motor servo tipe standard adalah dengan cara memberikan pulsa high selama 1,5 ms dan mengulanginya setiap 20 ms, maka posisi servo akan berada di-tengah atau netral (0). Untuk pulsa 1 ms maka akan bergerak berkebalikan arah jarum jam dengan sudut -90. Dan pulsa high selama 2 ms akan bergerak searah jarum jam sebesar 90.

## 6.4 Interface PWM di mikrokontroler AVR

Pulse Width Modulation menggunakan gelombang kotak dengan duty syle tertentu menghasilkan berbagai nilai rata-rata dari suatu bentuk gelombang, Duty syle merupakan presentase keadaan logika high (pulse) dalam satu periode sinyal. Satu siklus diawali dengan oleh transmisi low atau high dari sinyal dan berakhir pada transmisi berikutnya. Selama satu siklus, jika waktu sinyal pada keadaan high dsama dengan low maka dikatakan sinyal mempunyai duty syle 50%. Berikut adalah contoh rangkaian dan program menggunakan interface motor DC yang dikontrol dengan sinyal PWM dari mikrokontroler AVR.

### 6.4.1 Interface PWM menggunakan Motor DC

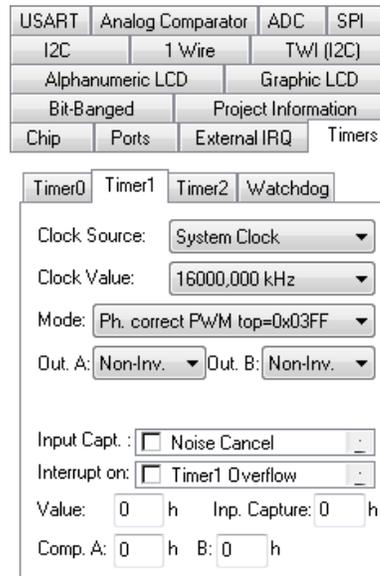
Rangkaian berikut adalah rangkaian mikrokontroler AVR ATmega16 yang dihubungkan dengan motor DC, pada dasarnya rangkaian motor DC yang membutuhkan arus kuat harus menggunakan sebuah driver motor. Fungsi dari driver ini adalah penguat arus dan tegangan. Pada rangkaian berikut menggunakan PORTD.0 dan PORTD.1 untuk memberikan sinyal putaran ke motor sekaligus sebagai pengatur arah rotasi putaran motor DC, dan PORTD.5 atau OC1A sebagai pin yang akan mengeluarkan sinyal kotak PWM yang dihubungkan ke kaki enable1 pada IC driver untuk mengatur kecepatan putaran motor DC. Rangkaian lengkapnya dapat dilihat pada gambar berikut.



Gambar 6.9: Rangkaian Interface dengan motor DC

PWM adalah sinyal yang dibangkitkan lewat fitur timer yang ada pada

mikrokontroler, maka sebelum melakukan pemrograman, timer perlu diaktifkan terlebih dahulu.



Gambar 6.10: Membangkitkan sinyal PWM dengan timer 1

Sebagai Contoh, pada gambar 6.10 menggunakan timer1 untuk membangkitkan sinyal PWM, meskipun timer yang lain pun juga dapat digunakan. Contoh source kode program yang menggunakan timer 1 dapat dilihat pada gambar 6.4.1

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/
#include <mega16.h>

```

```
-
void putarkanan () {
    PORTD.0=0;
    PORTD.1=1;
}
-
void putarkiri () {
    PORTD.0=1;
    PORTD.1=0;
}
-
void stop () {
    PORTD.0=0;
    PORTD.1=0;
}
-

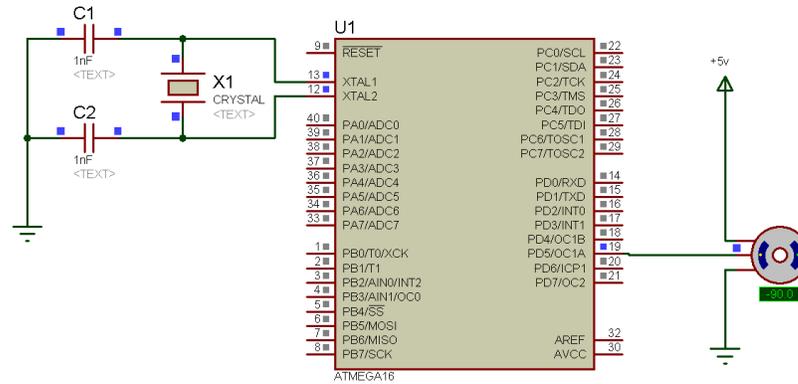
void main(void)
{
    PORTD=0x00;
    DDRD=0xFF;

    TCCR1A=0xA1;
    TCCR1B=0x01;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x00;
    ICR1L=0x00;
    OCR1AH=0x00;
    OCR1AL=0x00;
    OCR1BH=0x00;
    OCR1BL=0x00;

    while (1)
    {
        //memanggil program rotasi.
        putarkanan();
        //memberikan sinyal PWM sebesar 100 (mode 8bit).
        OCR1A=100;
    }
}
```

#### 6.4.2 Interface PWM menggunakan Motor Servo

Seperti penjelasan sebelumnya motor servo dapat berfungsi menggunakan satu data saja dengan menggunakan kontrol PWM pada fitur timer mikrokontroler. Berikut rangkaian interface motor servo yang dihubungkan pada pin OCR1A Mikrokontroler ATmega16.



Gambar 6.11: Interface Motor Servo dengan PWM

Berdasarkan dari rangkaian pada gambar 6.11 dapat dibuat listing program sebagai berikut.

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/
#include <mega16.h>
#include <delay.h>

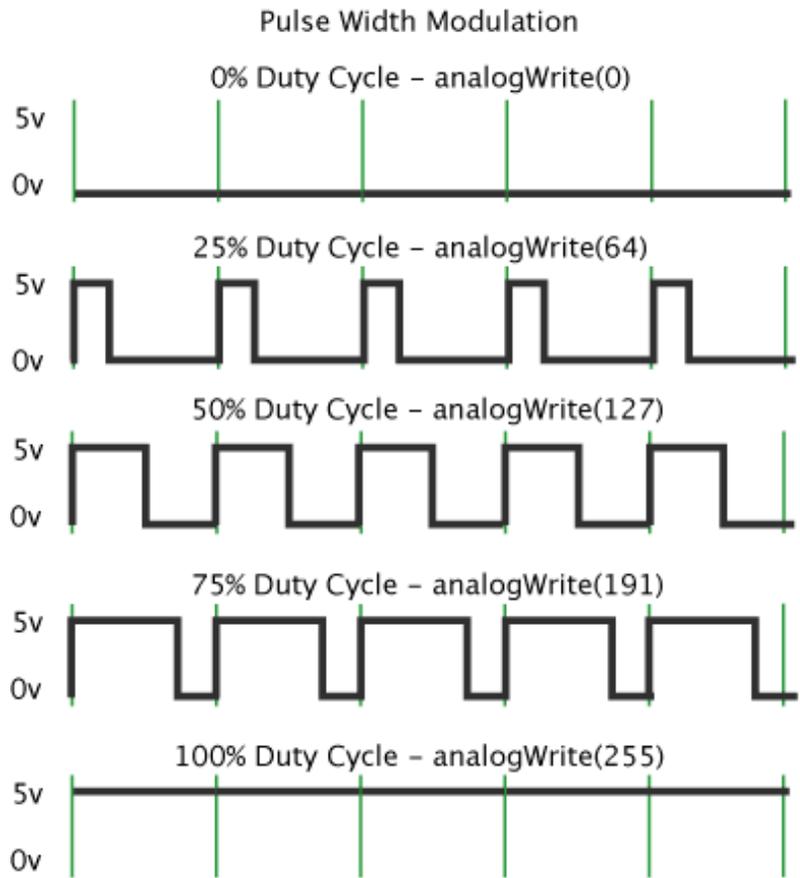
void set_servo(int sudut)
{
    OCR1A= (((sudut + 90)/90)*1000)/4;
}

void main(void)
{
    //iniliasiasi port sebagai output
    PORTD=0x00;
    DDRD=0x20;
    // iniliasiasi timer 1 untuk PWM
    TCCR1A=0x82;
    TCCR1B=0x1B;
}

```

```
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x13;  
ICR1L=0x87;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
//set servo ke 90 derajat  
set_servo(90);  
}
```

### 6.5 Interface PWM di Arduino



Gambar 6.12: PWM pada Arduino

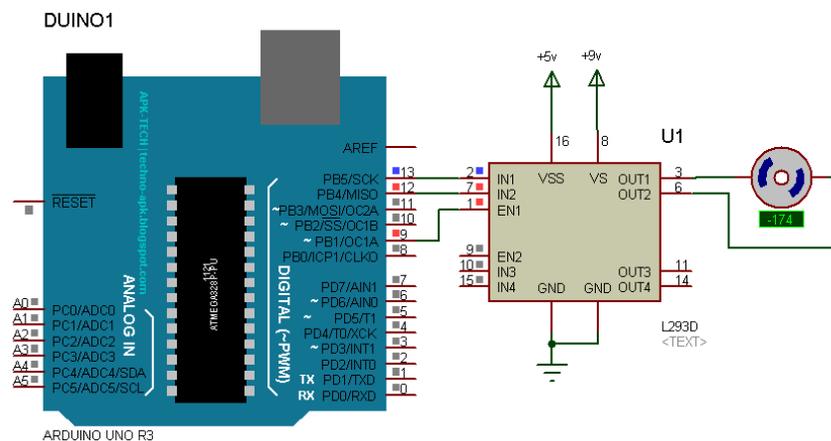
Pulse Width Modulation, atau PWM, adalah teknik untuk mendapatkan hasil analog dengan sarana digital. Kontrol digital digunakan untuk membuat gelombang persegi, sebuah sinyal beralih antara on dan off. Pola on-off ini dapat mensimulasikan tegangan di antara voltase penuh (5 Volt) dan mati (0 Volts) dengan mengubah bagian dari waktu yang dihabiskan sinyal versus waktu yang dihabiskan sinyal. Durasi "tepat waktu" disebut dengan lebar pulsa. Untuk mendapatkan berbagai nilai analog, Anda mengubah, atau memodulasi, lebar pulsa itu. Jika Anda mengulangi pola on-off ini cukup cepat dengan Motor DC misalnya, hasilnya seolah sinyal adalah voltase stabil antara 0 dan 5V yang mengendalikan kecepatan Motor DC.

Pada gambar 6.12, garis hijau mewakili periode waktu reguler. Durasi atau periode ini adalah kebalikan dari frekuensi PWM. Dengan kata lain, dengan frekuensi PWM Arduino sekitar 500Hz, garis hijau akan mengukur 2 milidetik masing-masing. Sebuah panggilan ke `analogWrite()` ada pada skala 0 - 255, sehingga `analogWrite(255)` meminta siklus kerja 100% (always on), dan `analogWrite(127)` adalah siklus tugas 50% (setengahnya) untuk contoh.

Arduino telah menyediakan pin yang bisa digunakan sebagai output PWM yang ditandai dengan "~", berikut adalah contoh penggunaannya dengan menggunakan interface motor DC dan motor Servo.

### 6.5.1 Interface PWM dengan Motor DC

Berikut adalah contoh rangkaian motor DC



Gambar 6.13: Arduino dengan Interface Motor DC dengan output PWM

Dalam pemrograman PWM dengan arduino dapat menggunakan perintah "analogWrite(pin,nilai);", pin adalah kaki yang yang digunakan sebagai output pwm, dan nilai diisi dengan nilai besar duty sycle yang akan diberikan pada output pwm. untuk lebih jelasnya perhatikan listing program berikut. Listring program sudah disesuaikan dengan rangkaian pada gambar 6.13.

Listing Program:

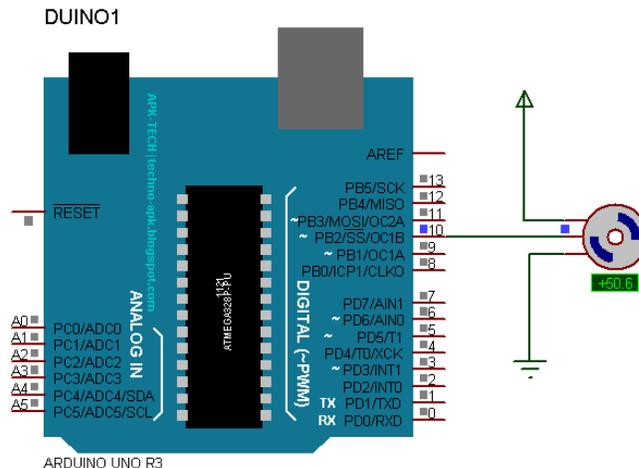
```
Motor_2_Data_PWM
void setup() {
  pinMode(13,OUTPUT); //set ke mode pin output
  pinMode(12,OUTPUT);
  pinMode(9,OUTPUT);
}

void loop() {
  //mengatur motor agar berputar.
  digitalWrite(12,HIGH);digitalWrite(13,LOW);

  //memberikan sinyal dutycycle PWM sebesar 100.
  //fungsinya adalah untuk mengatur kecepatan motor.
  analogWrite(9,150);
}
```

### 6.5.2 Interface PWM dengan Motor Servo

Berikut adalah contoh rangkaian interface motor servo yang terhubung dengan arduino pada pin 10.



Gambar 6.14: Arduino dengan Interface Motor DC dengan output PWM

Dalam pemrograman PWM pada servo dengan arduino dapat menggu-

nakan perintah "analogWrite(pin,nilai);", pin adalah kaki yang yang digunakan sebagai output pwm, dan nilai diisi dengan nilai besar duty sycle yang akan diberikan pada output pwm.

Pada pemrograman pwm untuk servo, arduino telah menyediakan pustaka yang dapat memudahkan pengguna untuk melakukan pemrograman, pustaka tersebut adalah "Servo.h". Penggunaan pustakan dan listring program untuk mengontrol servo dapat dilihat dibawah, listing program sudah disesuaikan dengan rangkaian pada gambar 6.14.

Listing Program:

```
Servo_PWM
#include <Servo.h>

Servo iniservo;

void setup() {
  //mendefinisikan servo terhubung pada pin digital 10 Arduino
  iniservo.attach(10);
}

void loop() {
  //mengatur derajat arah servo
  iniservo.write(120);
}
```

## Rangkuman

---

1. Timer/Counter adalah Seperangkat pencacah biner (binary counter) yang terhubung langsung ke saluran data mikrokontroler, sehingga mikrokontroler bisa membaca kondisi pencacah dan bila diperlukan mikrokontroler dapat pula merubah kondisi pencacah tersebut. Saat sinyal clock yang diberikan sudah melebihi kapasitas pencacah, maka pencacah akan memberikan sinyal overflow/limpahan. Limpahan pencacah ini dicatat dalam suatu register.
2. Sinyal clock yang diberikan ke pencacah dibedakan menjadi 2 macam yaitu Sinyal Clock dengan frekuensi tetap dan frekuensi bervariasi
3. Bit rate Timer / Counter dibagi menjadi 2 yaitu 8 bit dan 16 bit.
4. PWM ( Pulse Width Modulation) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (duty cycle) dengan nilai amplitudo dan frekuensi yang tetap.
5. Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo.
6. Pemrograman PWM pada mikrokontroler AVR menggunakan perintah "OCR1A=nilai;", sedangkan pada Arduino menggunakan perintah "analogWrite(pin,nilai);"

## Soal Formatif

---

1. Jelaskan pengertian Timer/Counter?
2. Jelaskan pengertian PWM?
3. Jelaskan komponen apa saja yang mampu dikelola menggunakan output PWM?
4. Jelaskan apa berbeda antara PWM dan DAC(*digital to analog conversion*)?
5. Jelaskan

## **Tugas Latihan**

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 2 buah pushbutton, dan 1 buah servo. Buatlah dengan algoritma, jika pushbutton pertama ditekan maka servo akan berkerak ke kanan, dan ketika pushbutton kedua ditekan maka servo akan bergerak ke kiri.

## Bab 7

# Interface Serial USART

### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan pengertian Komunikasi Serial
2. Peserta didik mampu menjelaskan pengertian Interface Serial USART
3. Peserta didik mampu mengimplementasikan fungsi antarmuka serial USART pada mikrokontroler AVR dan Arduino.
4. Peserta didik mampu membuat rangkaian dan pemrograman antarmuka serial USART pada mikrokontroler AVR dan Arduino.

## 7.1 Komunikasi Serial

Komunikasi Serial merupakan salah satu alternatif yang relatif lebih murah untuk menggantikan komunikasi paralel, karena transfer data paralel menggunakan 8 jalur konduktor/kawat untuk mentransfer 8bit sekaligus.

Dengan menggunakan komunikasi serial, maka hal tersebut dapat dilakukan dengan hanya menggunakan satu kawat konduktor saja, tetapi ditransfer bit demi bit sebanyak 8 kali untuk dapat menyelesaikan transfer satu byte data.

Selain relatif lebih murah, komunikasi serial juga memberikan jangkauan transmisi yang lebih panjang dari komunikasi paralel. Bandingkan dengan komunikasi paralel yang hanya bisa 1 hingga 2 meter saja, dengan komunikasi serial maka jangkauan tersebut bisa dilakukan berlipat-lipat ganda. Misalnya dengan menggunakan standar komunikasi serial EIA RS232 yang lebih dikenal dengan standard RS-232, dapat melakukan transmisi sejauh kurang lebih 50 feet dengan baud rate 9600. Transmisi yang lebih jauh dapat dilakukan pada data rate yang lebih rendah, dan jarak transmisi menjadi lebih pendek pada data rate yang lebih besar dari 9600 baud.

Device yang melakukan konversi paralel ke serial dan konversi serial ke paralel salah satunya disebut *universal asynchronous receiver-transmitter* (UART). Selain itu terdapat pula yang disebut *universal synchronous asynchronous receiver-transmitter* (USART), di mana selain dapat melakukan komunikasi serial secara asinkron juga dapat dengan cara sinkron. Baik cara sinkron maupun asinkron keduanya sama-sama mempunyai keuntungan dan kelemahannya masing-masing.

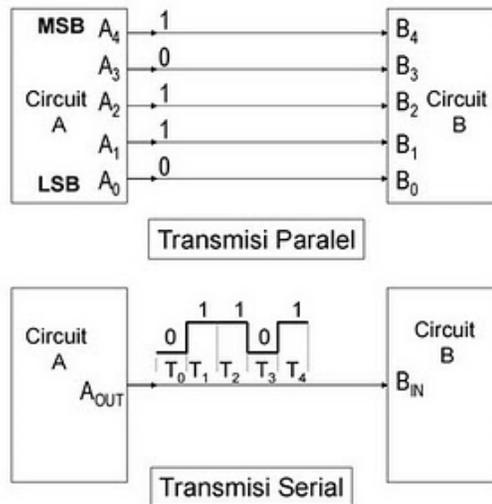
## 7.2 Antarmuka Serial dan USART

Ada dua jenis cara pengiriman data SERIAL USART, kedua cara tersebut dapat dibedakan berdasarkan sinyal detak(*clock*) yang dipakai untuk membawa data serial tersebut. Jika detak dikirim bersamaan dengan data serial maka cara pengiriman disebut dengan istilah pengiriman data serial secara sinkron (*Synchronous*). Bila pengiriman data dilakukan secara tak sinkron (*Asynchronous*) maka detak dikirim bersamaan dengan data serial, sehingga rangkaian penerima data harus membangkitkan sendiri detak pendorong data serial. .

Mikrokontroler AVR maupun Arduino telah mendukung kedua teknik pengiriman data serial tersebut, baik secara sinkron maupun asinkron. Penggabungan kedua teknik tersebut dikenal istilah *Universal Synchronous*

and Asynchronous serial Receiver and Transmitter (USART). Contoh media yang digunakan oleh antarmuka komunikasi Serial USART adalah RS-232, RS-422, dan RS-485.

Karena data dalam prosesor (CPU) diproses dalam bentuk paralel maka transfer data input/output serial harus dimulai dan diakhiri dengan data paralel. Pada dasarnya konversi paralel ke serial mudah dilakukan. Data paralel dimuat ke dalam sebuah register geser (shift register), kemudian register geser diberikan clock. Data kemudian dikeluarkan dari register geser mulai LSB (1 bit untuk setiap siklus clock). Bit pertama dari sebuah transmisi serial adalah data LSB (least significant bit). Bit kedua adalah bit LSB berikutnya, dan seterusnya. Bit data yang terakhir adalah MSB (most significant bit). Konversi paralel ke serial ditunjukkan pada Gambar 7.1



Gambar 7.1: Transmisi data paralel dan serial

Penerimaan data serial dan konversinya ke data paralel merupakan operasi kebalikannya. Data serial digeser ke dalam sebuah register geser, kemudian setelah semua bit di-clock ke dalam register geser maka data diambil secara paralel untuk ditransfer ke prosesor. Jadi dengan kata lain digunakan dua register geser, yang pertama register jenis PISO (parallel in serial out) dan di sisi yang lain digunakan jenis SIPO (serial in parallel out).

Pentransmisi data 8-bit dengan asinkron sebenarnya memerlukan pengiriman lo-bit. Ada tambahan bit start dan bit stop yang mengapit bit-bit data yang 8-bit tersebut. Bit pertama menyatakan penerimaan UART bahwa data telah tiba. Kemudian bit yang terakhir menyatakan bahwa bit data

(word) yang jumlahnya 1 byte (8-bit) telah selesai/lengkap. Pada Gambar 11.2 ditunjukkan sebuah word data 8-bit dengan penambahan 1 bit start dan 1 bit stop. Bit start umumnya bit dengan logika 0 (low level), sedangkan bit stop logika 1 (high level),

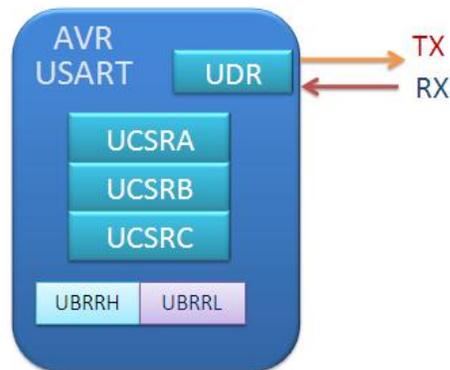
Kecepatan transmisi data komunikasi serial disebut baud rate. Baud rate atau signaling rate menyatakan berapa banyak bit yang ditransmisikan per detik. Misalnya, sebuah transmisi 1200 baud mengambil laju (rate) 120 karakter 10-bit (1 bit start, 8 bit data, dan 1 bit stop) per detik. Nilai-nilai baud rate yang lazim adalah 300 baud, 600 baud, 1200 baud, 2400 baud, 4800 baud, 9600 baud, 19200 baud, 38400 baud,

### 7.3 Register USART

AVR USART (Universal Synchronous Asynchronous Receiver Transmitter) adalah komunikasi serial dua arah yang terdapat di AVR yang melibatkan register register sbb:

UDR	USART I/O Data Register							
UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
UBRRH	USART Baud Rate Register Low Byte							
UBRRH	URSEL	–	–	–	UBRR[11:8]			
UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

Gambar 7.2: Register Penyusun USART



Gambar 7.3: Diagram Penyusun USART

Dari Gambar Register dapat dibagi menjadi beberapa blok yaitu:

### 7.3.1 Register UCSRA

- Bit 7 RXC: USART Receive Complete, RXC otomatis akan bernilai 1, jika ada data baru di bufer penerima. RXC otomatis akan bernilai 0, jika data sudah dibaca atau bufer penerima kosong.
- Bit 6 TXC: USART Transmit Complete, TXC otomatis akan bernilai 1, jika data di buffer selesai dikirim.
- Bit 5 UDRE: USART Data Register Empty, UDRE otomatis akan bernilai 1, jika register UDR kosong transmitter siap mengirim data. UDRE=0, UDR berisi data yg belum selesai dikirim .
- Bit 4 FE: Frame Error, FE otomatis akan bernilai 1, jika ada frame eror.
- Bit 3 DOR: Data OverRun, DOR otomatis akan bernilai 1, jika data datang ketika bufer penuh(terjadi antrian).
- Bit 2 PE: Parity Error, PE otomatis akan bernilai 1, jika terjadi parity eror.
- Bit 1 U2X: Double the USART Transmission Speed, kita set U2X=0, kecepatan normal. U2X=1 kecepatan 2x baudrate.
- Bit 0 MPCM: Multi-processor Communication Mode, kita set MCM=1 byte pertama yg diterima harus 9 bit, jika tdk data byte akan diabaikan. bit ini terjadi hanya untuk penerimaan saja pd komunikasi banyak microcontroller.

### 7.3.2 Register UCSRB

- Bit 7 RXCIE: RX Complete Interrupt Enable, kita set RXCIE=1, interupsi receive complete aktif.
- Bit 6 TXCIE: TX Complete Interrupt Enable, kita set TXCIE=1, interupsi transmit complete aktif.
- Bit 5 UDRIE: USART Data Register Empty Interrupt Enable, kita set UDRIE=1, interupsi UDRE aktif.
- Bit 4 RXEN: Receiver Enable, kita set RXEN=1, USART receiver aktif. micon bisa mnerima data.

- Bit 3 TXEN: Transmitter Enable, kita set TXEN=1, Usart Transmitter aktif. micon bisa mengirim data.
- Bit 2 UCSZ2: Character Size, kita set UCSZ2:UCSZ1:UCSZ0 = 011 , panjang data 8 BIT. (bit UCSZ1 dan UCSZ0 ada di register UCSRC)
- Bit 1 RXB8: Receive Data Bit 8, RXB8 menjadi bit ke-9 jika panjang data yg diterima 9 bit .
- Bit 0 TXB8: Transmit Data Bit 8, TXB8 menjadi bit ke-9 jika panjang data yg dikirim 9 bit.

### 7.3.3 Register UCSRC

- Bit 7 URSEL: Register Select . memilih UCSRC atau UBRRH, kita set URSEL=1 , UCSRC aktif ,UBRRH tdk aktif, kita set URSEL=0 , UBRRH aktif , UCRSC tdk aktif.
- Bit 6 UMSEL: USART Mode Select, kita set UMSEL=1 , mode synchronous. UMSEL=0 mode asynchronous
- Bit 5:4 UPM1:UMP0: Parity Mode,
- Bit 3 USBS: Stop Bit Select, kita set USBS=0, stop bit =1 bit , USBS=1 panjang stop bit = 2 bit.
- Bit 2:1 UCSZ1:0: Character Size, kita set UCSZ2:UCSZ1:UCSZ0 = 011 , panjang data 8 BIT. (bit UCSZ2 ada di register UCSRB)
- Bit 0 UCPCOL: Clock Polarity bit ini digunakan untuk mode synchronous saja, kita set UCPCOL=0 trnasmisi clock naik, UCPCOL=1 transmisi clock turun. (khusus yg ini dont care krn kita menggunakan mode asynchronous)

## 7.4 Fitur pada Serial USART

*UniversalSynchronousandAsynchronousserialReceiverandTransmitter(USART)* merupakan salah satu perangkat komunikasi serial yang mempunyai tingkat fleksibilitas komunikasi yang sangat tinggi. Pada mikrokontroler Atmel AVR, biasanya fitur USART mencakup:

1. Operasi *full duplex*.

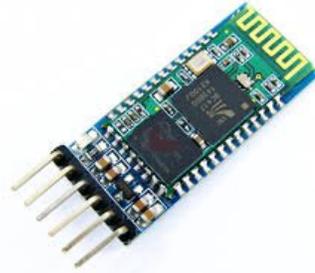
2. Dapat beroperasi pada mode *Asynchronous* dan *Synchronous*.
3. Dapat bekerja dengan resolusi *baud rate* yang tinggi.
4. Mendukung *serial frames* dengan data bit 5,6,7,8 atau 9 data bit dan dilengkapi dengan 1 atau 2 stop bit.
5. Dilengkapi dengan fasilitas *parity check* dan dapat bekerja pada *parity Odd* atau *Even*.
6. Dapat beroperasi sebagai *Master* atau *Slave Clock Synchronous*.
7. Dilengkapi dengan fitur data *Over Run Detection*.
8. Terdapat fasilitas *Frame Error Detection*.
9. Dilengkapi dengan filter untuk menyaring *noise* yaitu digital *low pass filter*.
10. Mempunyai tiga interupsi yang terpisah pada *Tx Complete*, *Data Register Empty* dan *Rx complete*.
11. Mendukung komunikasi multiprocessor.
12. Bekerja pada mode komunikasi *Double Speed Asynchronous*.

## 7.5 Komponen dengan antarmuka USART

Syarat komponen/alat untuk berkomunikasi dengan mikrokontroler lewat USART adalah komponen atau alat modul harus memiliki pin rx (receiver) dan pin tx(transmitter), biasanya komponen ini sudah berupa modul.

### 7.5.1 Bluetooth

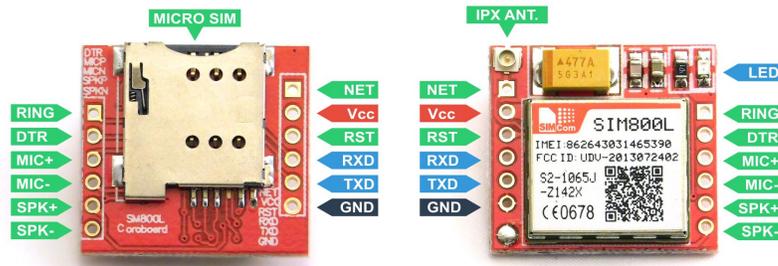
Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi pada pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frekuensi hopping transceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter). Bluetooth juga merupakan spesifikasi industri untuk jaringan kawasan pribadi (Personal Area Networks atau PAN) tanpa kabel. Bluetooth dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan (gadget).



Gambar 7.4: Bluetooth HC

### 7.5.2 Modul GSM

Modul GSM adalah perangkat kontrol elektronika yang peruntukannya pada sistem kontrol jarak jauh, Terdapat beberapa tipe *breakoutboard*, tetapi yang paling populer penggunaannya adalah versi mini dengan kartu gsm jenis micro SIM. Salah satu yang paling populer adalah Modul SIM800L. Modul SIM800L adalah salah satu Module GSM/GPRS Serial yang dapat digunakan bersama Arduino maupun mikrokontroler AVR.



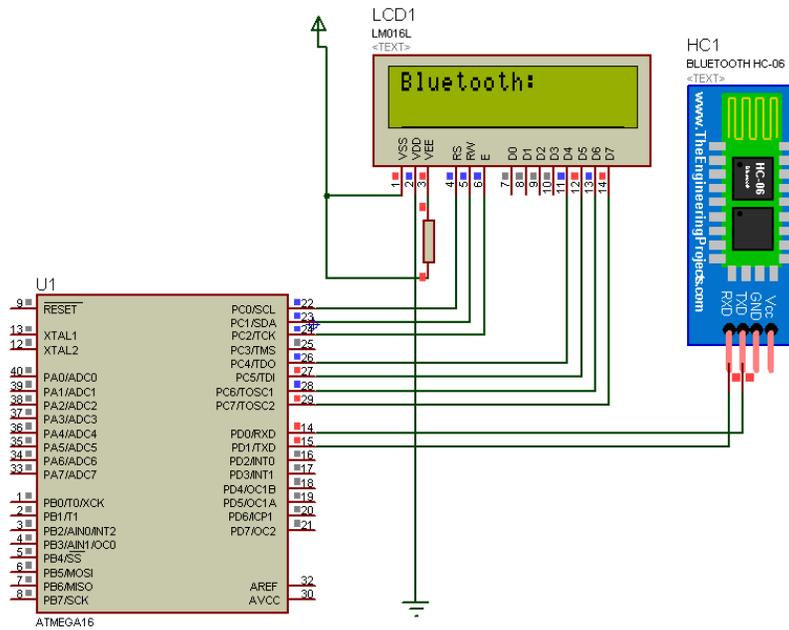
Gambar 7.5: Modul GSM SIM800L

## 7.6 Interface Serial USART pada Mikrokontroler AVR

Serial USART pada mikrokontroler AVR adalah komunikasi yang menggunakan 2 data sebagai pengirim (Tx) dan penerima (Tx) yang harus ada pada kedua perangkat yang akan berkomunikasi.

### 7.6.1 Rangkaian USART

Interface USART menggunakan 2 jalur data yaitu RX dan TX, berikut adalah contoh rangkaian yang menggunakan komunikasi Serial USART.



Gambar 7.6: Interface Bluetooth dengan Mikrokontroler ATmega16

pada gambar 7.6 dapat dilihat bahwa untuk mengkomunikasikan bluetooth dengan mikrokontroler hanya menggunakan 2 jalur data yaitu Rx dan Tx. Harus diperhatikan bahwa untuk dapat bertukar data atau berkomunikasi dalam interface data USART ini, pin Rx pada mikrokontroler harus dipasangkan dengan pin Tx pada Bluetooth, begitupun pin Tx pada mikrokontroler harus dipasangkan dengan pin Rx pada bluetooth.

### 7.6.2 Pemrograman USART

Berdasarkan dari rangkaian pada gambar 7.6 dapat dilakukan konfigurasi untuk mengaktifkan fungsi USART pada pin Rx dan TX di mikrokontroler pada aplikasi CodevisionAVR.

I2C	1 Wire	TWI (I2C)	
Alphanumeric LCD		Graphic LCD	
Bit-Banged		Project Information	
Chip	Ports	External IRQ	Timers
USART	Analog Comparator	ADC	SPI

Receiver       Rx Interrupt  
 Transmitter       Tx Interrupt

Baud Rate:   x2  
 Baud Rate Error: 0,2%  
 Communication Parameters:  
  
 Mode:

Gambar 7.7: Mengaktifkan Fungsi UART mikrokontroler

Berikut adalah contoh listing program pengaplikasian interface USART menggunakan bluetooth, listing program telah disesuaikan dengan rangkaian pada gambar 7.6

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

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

int bt;
char lcd[16];

```

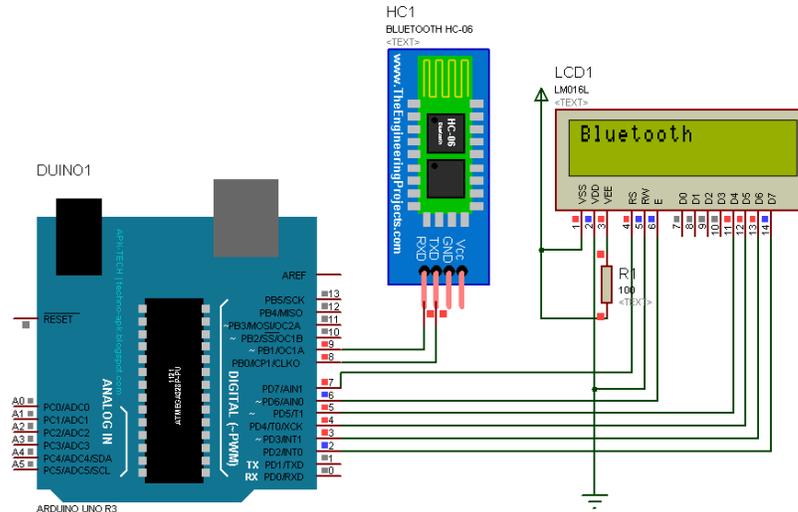
```
/******  
Chip type           : ATmega16  
Program type        : Application  
AVR Core Clock frequency: 16,000000 MHz  
Memory model        : Small  
External RAM size   : 0  
Data Stack size     : 256  
*****/  
  
#include <mega16.h>  
#include <alcd.h>  
#include <stdio.h>  
#include <delay.h>  
  
int bt;  
char lcd[16];  
  
void main(void)  
{  
  UCSRA=0x00;  
  UCSRB=0x18;  
  UCSRC=0x86;  
  UBRRH=0x00;  
  UBRRL=0x67;  
  
  lcd_init(16);  
  lcd_gotoxy(0,0);  
  lcd_puts("Bluetooth:");  
  
  while(1){  
    bt=getchar();  
    lcd_gotoxy(0,1);  
    sprintf(lcd,"%c",bt);  
    lcd_puts(lcd);  
    delay_ms(5);  
  }  
}
```

## 7.7 Interface Serial pada Arduino

Sama seperti mikrokontroler AVR, Interface USART pada mikrokontroler adalah komunikasi yang menggunakan 2 data sebagai pengirim (Tx) dan penerima (Tx) yang harus ada pada kedua perangkat yang akan berkomunikasi.

### 7.7.1 Rangkaian USART

Interface USART menggunakan 2 jalur data yaitu RX dan TX, berikut adalah contoh rangkaian yang menggunakan komunikasi Serial USART.



Gambar 7.8: Interface Bluetooth pada Arduino

Pada gambar 7.8 dapat dilihat bahwa untuk mengkomunikasikan bluetooth dengan arduino hanya menggunakan 2 jalur data yaitu Rx dan Tx. Harus diperhatikan bahwa untuk dapat bertukar data atau berkomunikasi dalam interface data USART ini, pin Rx pada mikrokontroler harus dipasangkan dengan pin Tx pada Bluetooth, begitupun pin Tx pada mikrokontroler harus dipasangkan dengan pin Rx pada bluetooth. Pengembangan yang diadakan oleh Arduino memungkinkan semua pinnya dipakai sebagai jalur data USART.

### 7.7.2 Pemrograman USART

pada pemrograman arduino pin USART telah tersedia pada pin digital 0 (Rx) dan pin digital 1 (Tx). Namun seperti diketahui semua pin pada Arduino dapat digunakan sebagai pin USART dengan cara memanggil library "SoftwareSerial.h", library ini memberikan kemungkinan pin selain pin 0 dan 1 dapat digunakan sebagai pin USART. Berikut adalah listing program untuk mengelolah data serial USART, listing program disesuaikan dengan rangkaian pada gambar 7.8.

## Listing Program:

```
lcd_bluetoth $
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
SoftwareSerial BT(9, 8); //TX,RX
String data;

void setup() {
  lcd.begin(16, 2);           // menggunakan LCD 16x2
  BT.begin(9600);           // bit rate USART
}
void loop() {
  lcd.setCursor(0,0);lcd.print("Bluetooth");
  while (BT.available()) {
    delay(10);
    char c = BT.read();
    data += c;
  }
  if(data.length() > 0) {
    Serial.println(data);
    // perintah dibawah ini dapat anda sesuaikan dengan perintah anda.
    if (data == "A")
    {lcd.setCursor(0,1);lcd.print("data A"); }
    else if(data == "B")
    {lcd.setCursor(0,1);lcd.print("data B"); }
    data="";
  }
}
```

Pada listing program diatas dapat diperhatikan bahwa pin digital 8 digunakan sebagai pin TX dan pin digital 9 digunakan sebagai RX. Data yang terkirim dan diterima oleh bluetooth akan dikelolah dan ditampilkan ke LCD 16x2.

## Rangkuman

---

1. Komunikasi Serial merupakan salah satu alternatif yang relatif lebih murah untuk menggantikan komunikasi paralel, karena transfer data paralel menggunakan 8 jalur konduktor/kawat untuk mentransfer 8bit sekaligus.
2. Device yang melakukan konversi paralel ke serial dan konversi serial ke paralel salah satunya disebut universal asynchronous receiver-transmitter (UART).
3. Contoh perangkat yang berkomunikasi lewat interface USART adalah bluetooth. Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi pada pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frekuensi hopping transceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter).
4. Komunikasi USART menggunakan 2 pin data yaitu Rx dan Tx.
5. Pemrograman Komunikasi Serial USART menggunakan perintah "getchar()" untuk mendapatkan data, sedangkan pada pemrograman Arduino hanya menggunakan sebuah librari yaitu "SoftwareSerial.h"

## Soal Formatif

---

1. Jelaskan pengertian komunikasi serial?
2. Jelaskan pengertian komunikasi serial USART?
3. Jelaskan fungsi kedua pin (Rx dan Tx) yang ada pada komunikasi serial USART?
4. Berikan contoh implementasi interface serial USART yang dapat dibuat menggunakan mikrokontroler AVR dan Arduino?

## **Tugas Latihan**

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 1 buah komponen bluetooth dan 1 buah led dan sebuah aplikasi smartphone yang dapat memberikan umpan dat ke bluetooth. Buatlah dengan algoritma, smartphone dapat mengirimkan sinyal ke mikrokontroler untuk menyalakan dan mematikan LED.



## Bab 8

# Interface Serial SPI

### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan Komunikasi Serial SPI.
2. Peserta Didik mampu membuat rangkaian dan pemrograman innterface serial SPI pada mikrokontroler AVR dan Arduino.

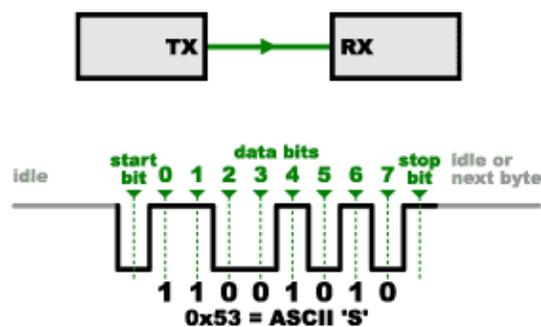
## 8.1 Antarmuka Serial dan SPI

Serial Peripheral Interface ( SPI ) adalah sebuah antarmuka bus yang biasa digunakan untuk mengirim data antara mikrokontroler dan perangkat kecil seperti register geser , sensor , dan kartu SD . Menggunakan jam dan data baris terpisah , bersama dengan garis pilih untuk memilih perangkat yang ingin Anda ajak bicara.

Serial Peripheral Interface (SPI) merupakan salah satu mode komunikasi serial synchronous kecepatan tinggi yang dimiliki oleh ATmega8535. Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART) juga merupakan salah satu mode komunikasi serial yang dimiliki oleh ATmega8535.

Sebuah port serial yang umum , jenis dengan TX dan RX baris , disebut asynchronous ( tidak sinkron ) karena tidak ada kontrol atas saat data dikirim atau jaminan bahwa kedua belah pihak berjalan tepat pada tingkat yang sama . Karena komputer biasanya mengandalkan segala sesuatu yang disinkronkan ke satu jam ( kristal utama yang melekat pada komputer yang mendorong segala sesuatu ) , ini bisa menjadi masalah ketika dua sistem dengan jam sedikit berbeda mencoba untuk berkomunikasi satu sama lain .

Untuk mengatasi masalah ini , koneksi serial asynchronous menambahkan start dan stop bit ekstra untuk setiap byte bantuan sync penerima up data setelah diterima. Kedua belah pihak juga harus setuju pada kecepatan transmisi ( seperti 9600 bit per detik ) di muka. Sedikit perbedaan dalam tingkat transmisi tidak masalah karena receiver re sync pada awal setiap byte .

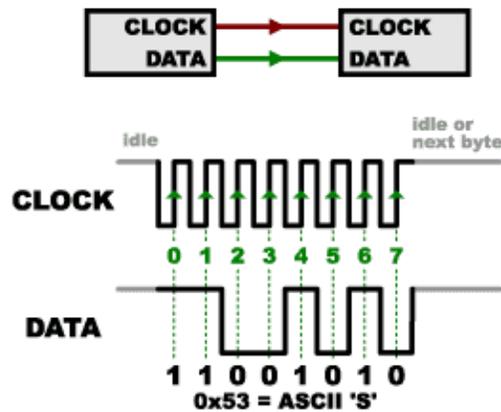


Gambar 8.1: Komunikasi Serial

Serial asynchronous bekerja dengan baik , namun memiliki banyak overhead baik di start dan stop bit ekstra dikirim dengan setiap byte , dan

hardware yang kompleks yang diperlukan untuk mengirim dan menerima data. Dan seperti yang Anda mungkin telah melihat dalam proyek Anda sendiri, jika kedua belah pihak tidak diatur dengan kecepatan yang sama, data yang diterima akan menjadi sampah. Hal ini karena penerima sampling bit pada waktu yang sangat spesifik (panah dalam diagram di atas). Jika penerima melihat pada saat yang salah, itu akan melihat bit yang salah.

SPI bekerja dalam cara yang sedikit berbeda. Ini adalah sinkron data bus, yang berarti bahwa ia menggunakan jalur terpisah untuk data dan jam yang membuat kedua belah pihak di sync sempurna. Jam adalah sinyal osilasi yang memberitahu penerima persis kapan untuk sampel bit pada baris data. Ini bisa menjadi naik (rendah ke tinggi) atau jatuh (tinggi ke rendah) tepi sinyal clock, datasheet akan menentukan mana yang akan digunakan. Ketika penerima mendeteksi bahwa tepi, segera akan melihat baris data untuk membaca bit berikutnya (lihat panah di bawah diagram). Karena jam tersebut dikirim bersama dengan data, menentukan kecepatan tidak penting, meskipun perangkat akan memiliki kecepatan puncak di mana mereka dapat beroperasi (Kita akan membahas memilih jam yang tepat tepi dan kecepatan dalam sedikit).

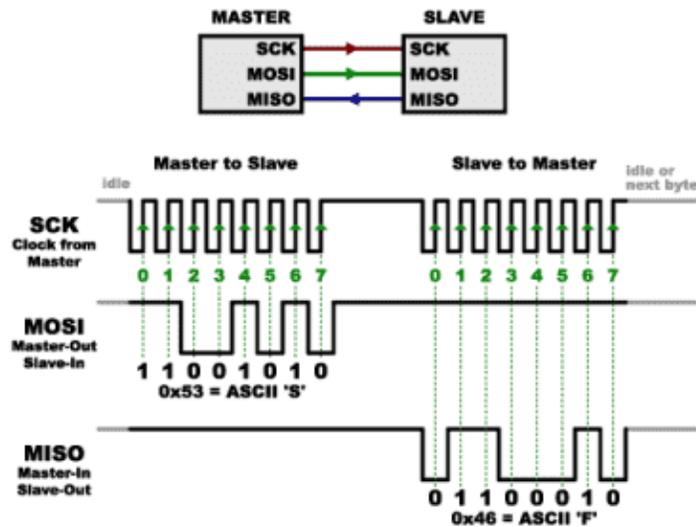


Gambar 8.2: Komunikasi Serial

Salah satu alasan bahwa SPI sangat populer adalah bahwa hardware penerima bisa menjadi register geser sederhana. Ini adalah jauh lebih sederhana (dan murah!) Bagian dari perangkat keras daripada UART full-up (Universal Asynchronous Receiver / Transmitter) bahwa serial asynchronous membutuhkan.

Dalam SPI , hanya satu sisi menghasilkan sinyal clock (biasanya disebut CLK atau SCK untuk Serial Clock) . Sisi yang menghasilkan jam disebut master , dan sisi lainnya disebut budak . Selalu hanya satu master ( yang hampir selalu mikrokontroler Anda ) , tetapi bisa ada beberapa budak ( lebih lanjut tentang ini dalam sedikit ) .

Ketika data dikirim dari master ke slave , itu dikirim pada baris data yang disebut MOSI , untuk Guru Out / Slave In . Jika slave harus mengirim respon kembali ke master , master akan terus menghasilkan sejumlah diatur sebelumnya clock cycle , dan budak akan menempatkan data ke jalur data ketiga yang disebut MISO , untuk Guru In / Out Slave .



Gambar 8.3: Komunikasi Serial

Perhatikan kita berkata sudah diatur sebelumnya dalam uraian di atas . Karena master selalu menghasilkan sinyal clock , maka harus mengetahui terlebih dahulu ketika slave perlu kembali data dan berapa banyak data yang akan dikembalikan . Hal ini sangat berbeda dari serial asynchronous , di mana jumlah acak data dapat dikirim di kedua arah setiap saat . Dalam prakteknya hal ini tidak masalah, karena SPI umumnya digunakan untuk berbicara dengan sensor yang memiliki struktur perintah yang sangat spesifik . Misalnya, jika Anda mengirim perintah untuk membaca data ke perangkat, Anda tahu bahwa perangkat akan selalu mengirimkan , misalnya , dua byte kembali . ( Dalam kasus di mana Anda mungkin ingin kembali sejumlah variabel data, Anda selalu bisa kembali satu atau dua byte menen-

tukan panjang dari data dan kemudian memiliki master mengambil jumlah penuh . )

Perhatikan bahwa SPI adalah full duplex ( memiliki mengirim dan menerima baris yang terpisah ) , dan , dengan demikian , dalam situasi tertentu , Anda dapat mengirim dan menerima data pada waktu yang sama ( misalnya , meminta pembacaan sensor baru saat mengambil data dari sebelumnya ) . Datasheet perangkat Anda akan memberitahu Anda jika hal ini mungkin . Slave Pilih ( SS )

Ada satu baris terakhir Anda harus menyadari , yang disebut SS untuk Slave Pilih. Ini memberitahu budak yang harus bangun dan menerima / mengirim data dan juga digunakan ketika beberapa budak yang hadir untuk memilih salah satu yang ingin diajak bicara.

Garis SS biasanya terangkat tinggi , yang memutus hubungan budak dari bus SPI . ( Jenis logika dikenal sebagai aktif rendah , dan Anda akan sering melihat digunakan untuk mengaktifkan dan me-reset baris . ) Tepat sebelum data dikirim ke budak , garis dibawa rendah , yang mengaktifkan budak . Ketika Anda selesai menggunakan budak , garis dibuat tinggi lagi. Dalam sebuah register geser , ini sesuai dengan kait masukan , yang mentransfer data yang diterima ke garis output.

## 8.2 Komponen dengan antarmuka SPI

Syarat komponen/alat untuk berkomunikasi dengan mikrokontroler lewat SPI adalah komponen/alat harus memiliki pin untuk jalur data Mosi, Miso, dan Sck. Berikut beberapa modul yang menggunakan interface SPI:

### 8.2.1 Downloader ASP/ISP

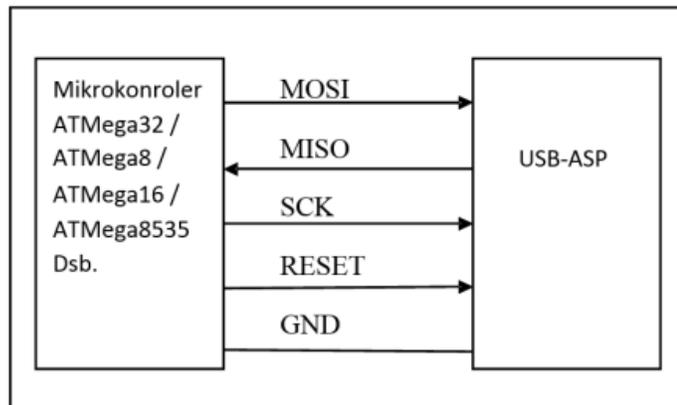
Downloader adalah sebuah alat yang digunakan untuk memasukkan program ke dalam mikrokontroler, baik itu yang berjenis MCS ataupun AVR semuanya butuh downloader. Downloader bisa juga diartikan sebagai jembatan penghubung antara komputer dengan mikrokontroler. Yang mana file.hex yang telah dibuat dari compile file.bas dari software BASCOM-AVR dimasukkan ke dalam mikrokontroler. Downloader yang umum digunakan untuk memasukkan data dari komputer ke mikrokontroler yaitu USB-ASP, namun ada pula yang tidak memakai USB-ASP yaitu menggunakan serial parallel port untuk melakukan download programnya.

Untuk sistem kerja dari downloader, terutama USB-ASP yaitu dengan cara memberikan pin MISO, MOSI, SCK, RESET, dan GROUND pada rangkaian downloader, dipasangkan ke mikrokontroler terutama ATmega



Gambar 8.4: Downloader

yaitu pada pin yang sama MOSI, MISO, SCK, RESET, GROUND. Sehingga akan nampak seperti bagan gambar berikut ini.



Gambar 8.5: Komunikasi ISP pada Downloader

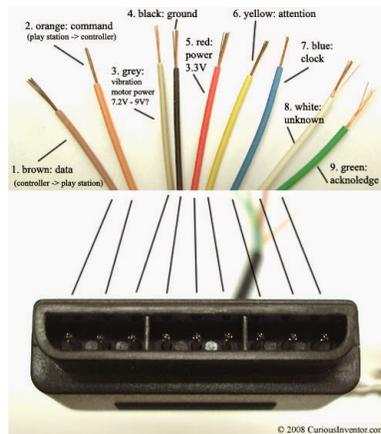
### 8.2.2 Joystick PS2

Joystick PS2 ada 2 tipe yang Wire / kabel dan Wireless tanpa kabel, pada ajang KRAI penggunaan joystick ini sangat familiar karena mudah penggunaan'y dan simple. Interface Joystick PS dengan Mikrokontroler ini menggunakan komunikasi SPI (Serial Peripheral Interface) atau yg dikenal dengan 3 kabel data.



Gambar 8.6: Joystick PS2

Joystick PS2 dengan mikrokontroler cukup sederhana hanya 4 pin yaitu Mosi, Miso, SCK, dan Pin bebas. Untuk pin SCK, MOSI, PB4(bebas) diberi resistor 220 ohm sebagai pengaman, sedangkan resistor 4k7 ohm pada MISO sebagai Pull-up. Pin SCK dihubungkan ke kabel joystick warna biru, PB4 (pin bebas) dihubungkan ke kabel joystick warna kuning, Pin MOSI dihubungkan ke kabel joystick warna Coklat dan Mosi dihubungkan ke kabel joystick warna Coklat dan diberi pull-up. Sedangkan untuk power joystick diberi tegangan 5 V (merah) dan ground (hitam).



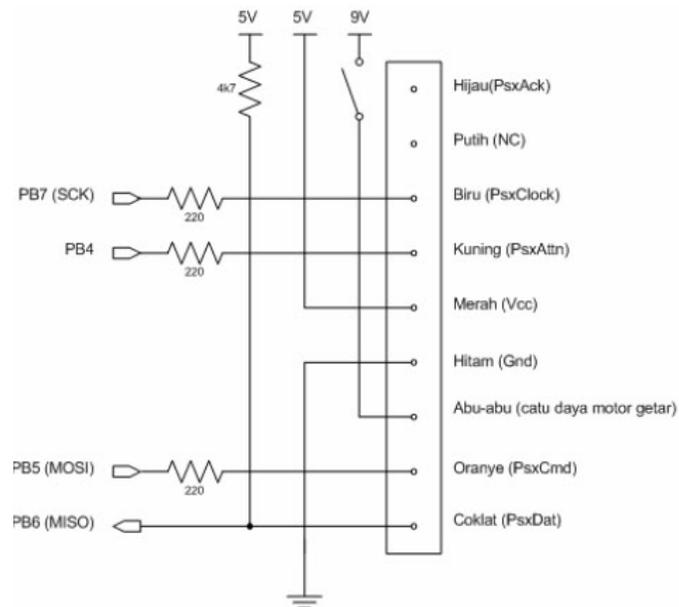
Gambar 8.7: Konfigurasi PIN pada Joystick PS2

### 8.3 Interface Serial pada Mikrokontroler AVR

Serial SPI pada mikrokontroler AVR adalah komunikasi yang menggunakan 3 data yaitu mosi, miso, dan sck yang harus ada pada kedua perangkat yang akan berkomunikasi.

#### 8.3.1 Rangkaian ISP

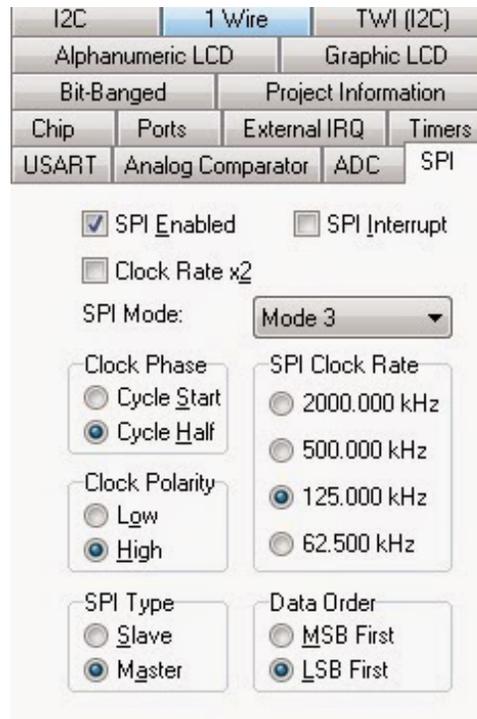
Berbeda pada rangkaian USART pemasangan pin Rx dan Tx harus ditukar untuk dapat berkomunikasi. Pada Rangkaian ISP pemasangannya harus lurus. Pada rangkaian tertentu biasanya menggunakan rangkaian tambahan berupa rangkaian penstabil sinyal tegangan berupa resistor seperti pada contoh berikut rangkaian komunikasi Joystick PS2 ke Mikrokontroler AVR.



Gambar 8.8: Rangkaian Konfigurasi PIN pada Joystick PS2

#### 8.3.2 Pemrograman SPI

mengakses Stik Ps2 ke mikrokontroler AVR menggunakan jalur SPI seperti yang terlihat pada gambar ???. Kontroler stik ps2 ini biasa digunakan untuk mengontrol robot secara manual. Pertama adalah mengaktifkan fitur SPI



Gambar 8.9: Mengaktifkan Fungsi SPI mikrokontroler

Program SPI lebih rumit dibandingkan dengan USART dikarenakan setiap bit yang diakses harus dilakukan registrasi untuk mendapat data, berikut adalah contoh listing program pengambilan data SPI dan menapilkankan data tersebut di LCD.

Listing Program:

```

/*****
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/
#include <mega16.h>
#include <stdlib.h>
// Alphanumeric LCD Module functions
#include <alcd.h>

```

```
// SPI functions
#include <spi.h>
#include <delay.h>
// Declare your global variables here
unsigned char spi(unsigned char data);

void main(void)
{
// Declare your local variables here
unsigned char byte4, byte5, byte6, byte7, byte8, byte9;
unsigned char ID, SS;
unsigned char *analog;
unsigned char *analog6;
unsigned char *analog7;
unsigned char *analog8;
unsigned char *analog9;

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=Out Func4=In Func3=Out Func2=Out
// State7=T State6=T State5=0 State4=T State3=0 State2=0 St
PORTB=0x04;
DDRB=0x2C;
// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 62.500 kHz
// SPI Clock Phase: Cycle Half
// SPI Clock Polarity: High
// SPI Data Order: LSB First
SPCR=0x7F;
SPSR=0x00;

lcd_init(16);

while (1)
{
// Place your code here
lcd_clear();
PORTB.2 = 0;
spi(0x01);
ID = spi(0x42);
SS = spi(0);
byte4 = spi(0);
byte5 = spi(0);
byte6 = spi(0);
byte7 = spi(0);
byte8 = spi(0);
byte9 = spi(0);
PORTB.2 = 1;
```

```
lcd_clear();
itoa(ID, analog);
lcd_gotoxy(0,0);
lcd_puts(analog);

if (ID == 65)
{
    lcd_gotoxy(5,0);
    lcd_puts("analog off");
    delay_ms(500);
}

else if (ID == 115)
{
    lcd_gotoxy(5,0);
    lcd_puts("analog on");
    delay_ms(500);
}

lcd_clear();
if (byte4 == 0b01111111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("left");
    delay_ms(700);
}

else if (byte4 == 0b10111111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("down");
    delay_ms(700);
}

else if (byte4 == 0b11011111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("right");
    delay_ms(700);
}

else if (byte4 == 0b11101111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("up");
    delay_ms(700);
}
```

```
else if (byte4 == 0b11110111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("start");
    delay_ms(700);
}

else if (byte4 == 0b11111110)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("select");
    delay_ms(700);
}

else if (byte5 == 0b01111111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("kotak");
    delay_ms(700);
}

else if (byte5 == 0b10111111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("X");
    delay_ms(700);
}

else if (byte5 == 0b11011111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("bulat");
    delay_ms(700);
}

else if (byte5 == 0b11101111)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("segitiga");
    delay_ms(700);
}

else if (byte5 == 0b11110111)
{
    lcd_clear();
```

```

    lcd_gotoxy(0,1);
    lcd_puts("R1");
    delay_ms(700);
}

else if (byte5 == 0b1111011)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("L1");
    delay_ms(700);
}

else if (byte5 == 0b1111101)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("R2");
    delay_ms(700);
}

else if (byte5 == 0b1111110)
{
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_puts("L2");
    delay_ms(700);
}

else if ( (ID == 115) &&
(byte6 == 0b0000000) || (byte6 == 0b1111111) )
{itoa(byte6, analog6);
lcd_clear();
lcd_gotoxy(0,1);
lcd_puts(analog6);
lcd_gotoxy(0,5);
lcd_puts(" analog ON");
delay_ms(700);}

else if ( (ID == 115) &&
(byte7 == 0b0000000) || (byte7 == 0b1111111) )
{itoa(byte7, analog7);
lcd_clear();
lcd_gotoxy(0,1);
lcd_puts(analog7);
lcd_gotoxy(0,5);
lcd_puts(" analog ON");
delay_ms(700);}

else if ((ID == 115) &&
(byte8 == 0b0000000) || (byte8 == 0b1111111) )

```

```

{itoa(byte8, analog8);
lcd_clear();
lcd_gotoxy(0,1);
lcd_puts(analog8);
lcd_gotoxy(0,5);
lcd_puts(" analog ON");
delay_ms(700);}

else if ((ID == 115) &&
(byte9 == 0b00000000) || (byte9 == 0b11111111) )
{itoa(byte9, analog9);
lcd_clear();
lcd_gotoxy(0,1);
lcd_puts(analog9);
lcd_gotoxy(0,5);
lcd_puts(" analog ON");
delay_ms(700);}

lcd_clear();
}
}

```

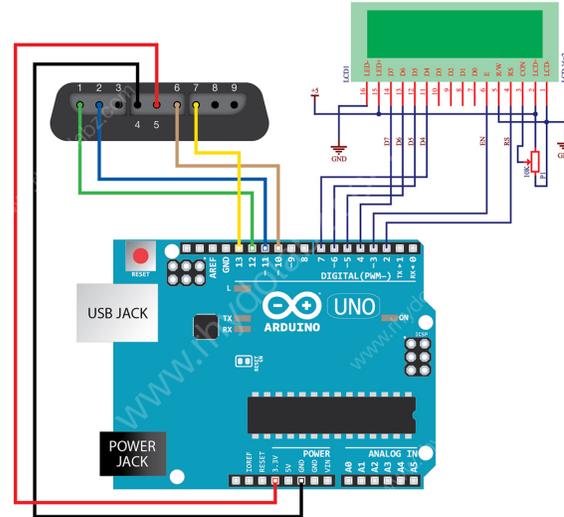
## 8.4 Interface Serial pada Arduino

Sama seperti mikrokontroler AVR, Interface SPI pada mikrokontroler adalah komunikasi yang menggunakan 3 data yaitu mosi, miso, dan sck yang harus ada pada kedua perangkat yang akan berkomunikasi.

Salah satu contoh perangkat yang menggunakan komunikasi antarmuka SPI adalah joystick. Pengaplikasian antarmuka joystick pada arduino terbilang sangatlah mudah dibandingkan menggunakan mikrokontroler AVR. Dalam menggunakan komunikasi SPI, Arduino telah mengembangkan dan menyediakan port yang dapat langsung dihubungkan dengan perangkat antarmuka begtupan dalam hal pemrograman telah tersedia banyak library yang siap digunakan tergantung dari jenis perangkat tersebut. Lebih jelasnya, berikut adalah salah satu contoh rangkaian dan pemrogramna menggunakan interface joystick.

### 8.4.1 Rangkaian SPI pada JoyStick

Dapat diamati pada gambar 8.10 bahwa interface untuk arduino tidak lagi menggunakan rangkaian tambahan seperti yang digunakan pada mikrokontroler jenis AVR.



Gambar 8.10: Interface Joystick PS2 pada Arduino

### 8.4.2 Pemrograman SPI pada JoyStick

Beikut adalah contoh pemrograman SPI, pada contoh berikut menggunakan joystick PS2 sebagai interface ke Arduino. Pemrograman Joystick PS2 menggunakan library "PS2X.lib.h", library tersebut telah menyediakan semua fitur yang berhubungan dengan kontrol joystick PS2 yang berkomunikasi lewat interface serial SPI. Program dibawah ini adalah program default atau program standar bawaan dari library "PS2X.lib.h". Program seperti dibawah dapat diperoleh di bagian *example* yang telah disediakan oleh aplikasi Arduino.IDE, jika contohnya program belum ada maka pengguna arduino harus mengunduh librarynya terlebih dahulu.

Listing Program:

```
PS2X_Example $
#include <PS2X_lib.h> //for v1.6

#define PS2_DAT      13 //14
#define PS2_CMD      11 //15
#define PS2_SEL      10 //16
#define PS2_CLK      12 //17
#define pressures    false
#define rumble        false

PS2X ps2x; // create PS2 Controller Class
```

```

//right now, the library does NOT support hot pluggable controllers, meaning
//you must always either restart your Arduino after you connect the controller,
//or call config_gamepad(pins) again after connecting the controller.

int error = 0;
byte type = 0;
byte vibrate = 0;

void setup(){

  Serial.begin(57600);

  delay(300);
  //CHANGES for v1.6 HERE!!! *****PAY ATTENTION*****

  error = ps2x.config_gamepad(PS2_CLK, PS2_CMD, PS2_SEL, PS2_DAT, pressures, rumble);

  if(error == 0){
    Serial.print("Found Controller, configured successful ");
    Serial.print("pressures = ");
    if (pressures)
      Serial.println("true ");
    else
      Serial.println("false");
    Serial.print("rumble = ");
    if (rumble)
      Serial.println("true");
    else
      Serial.println("false");
    Serial.println("Try out all the buttons,");
    Serial.println("holding L1 or R1 will print out the analog stick values.");
    Serial.println("Note: Go to www.billporter.info for updates and to report bugs.");
  }
  else if(error == 1)
    Serial.println("No controller found, check wiring");

  else if(error == 2)
    Serial.println("Controller found but not accepting commands.");

  else if(error == 3)
    Serial.println("Controller refusing to enter Pressures mode, may not support it. ");

  // Serial.print(ps2x.Analog(1), HEX);

  type = ps2x.readType();
  switch(type) {
    case 0:
      Serial.print("Unknown Controller type found ");
      break;
    case 1:
      Serial.print("DualShock Controller found ");
      break;
    case 2:
      Serial.print("GuitarHero Controller found ");
      break;
    case 3:
      Serial.print("Wireless Sony DualShock Controller found ");
      break;
  }
}

```

```

void loop() {
  /* You must Read Gamepad to get new values and set vibration values
   ps2x.read_gamepad(small motor on/off, larger motor strenght from 0-255)
   if you don't enable the rumble, use ps2x.read_gamepad(); with no values
   You should call this at least once a second
  */
  if(error == 1) //skip loop if no controller found
    return;

  if(type == 2){ //Guitar Hero Controller
    ps2x.read_gamepad(); //read controller

    if(ps2x.ButtonPressed(GREEN_FRET))
      Serial.println("Green Fret Pressed");
    if(ps2x.ButtonPressed(RED_FRET))
      Serial.println("Red Fret Pressed");
    if(ps2x.ButtonPressed(YELLOW_FRET))
      Serial.println("Yellow Fret Pressed");
    if(ps2x.ButtonPressed(BLUE_FRET))
      Serial.println("Blue Fret Pressed");
    if(ps2x.ButtonPressed(ORANGE_FRET))
      Serial.println("Orange Fret Pressed");

    if(ps2x.ButtonPressed(STAR_POWER))
      Serial.println("Star Power Command");

    if(ps2x.Button(UP_STRUM)) //will be TRUE as long as button is pressed
      Serial.println("Up Strum");
    if(ps2x.Button(DOWN_STRUM))
      Serial.println("DOWN Strum");

    if(ps2x.Button(PSB_START)) //will be TRUE as long as button is pressed
      Serial.println("Start is being held");
    if(ps2x.Button(PSB_SELECT))
      Serial.println("Select is being held");

    if(ps2x.Button(ORANGE_FRET)) { // print stick value IF TRUE
      Serial.print("Wammy Bar Position:");
      Serial.println(ps2x.Analog(WHAMMY_BAR), DEC);
    }
  }
  else { //DualShock Controller
    ps2x.read_gamepad(false, vibrate);

    if(ps2x.Button(PSB_START)) //will be TRUE as long as button is pressed
      Serial.println("Start is being held");
    if(ps2x.Button(PSB_SELECT))
      Serial.println("Select is being held");

    if(ps2x.Button(PSB_PAD_UP)) { //will be TRUE as long as button is pressed
      Serial.print("Up held this hard: ");
      Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
    }
    if(ps2x.Button(PSB_PAD_RIGHT)){
      Serial.print("Right held this hard: ");
      Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
    }
    if(ps2x.Button(PSB_PAD_LEFT)){
      Serial.print("LEFT held this hard: ");
      Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
    }
  }
}

```

```

if(ps2x.Button(PSB_PAD_DOWN)){
  Serial.print("DOWN held this hard: ");
  Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
}

vibrate = ps2x.Analog(PSAB_CROSS);
if (ps2x.NewButtonState()) {
  if(ps2x.Button(PSB_L3))
    Serial.println("L3 pressed");
  if(ps2x.Button(PSB_R3))
    Serial.println("R3 pressed");
  if(ps2x.Button(PSB_L2))
    Serial.println("L2 pressed");
  if(ps2x.Button(PSB_R2))
    Serial.println("R2 pressed");
  if(ps2x.Button(PSB_TRIANGLE))
    Serial.println("Triangle pressed");
}

if(ps2x.ButtonPressed(PSB_CIRCLE))
  Serial.println("Circle just pressed");
if(ps2x.NewButtonState(PSB_CROSS))
  Serial.println("X just changed");
if(ps2x.ButtonReleased(PSB_SQUARE))
  Serial.println("Square just released");

if(ps2x.Button(PSB_L1) || ps2x.Button(PSB_R1)) {
  Serial.print("Stick Values:");
  Serial.print(ps2x.Analog(PSS_LY), DEC);
  Serial.print(",");
  Serial.print(ps2x.Analog(PSS_LX), DEC);
  Serial.print(",");
  Serial.print(ps2x.Analog(PSS_RY), DEC);
  Serial.print(",");
  Serial.println(ps2x.Analog(PSS_RX), DEC);
}
}
delay(50);
}

```

---

## Rangkuman

---

1. Serial Peripheral Interface ( SPI ) adalah sebuah antarmuka bus yang biasa digunakan untuk mengirim data antara mikrokontroler dan perangkat kecil seperti register geser , sensor , dan kartu SD .
2. Contoh perangkat yang berkomunikasi lewat interface ISP adalah downloader. Downloader adalah sebuah alat yang digunakan untuk memasukkan program ke dalam mikrokontroler, baik itu yang berjenis MCS ataupun AVR semuanya butuh downloader.
3. Komunikasi SPI menggunakan 3 pin data yaitu mosi, miso, dan sck.
4. Pemrograman Komunikasi Serial SPI pada mikrokontroler AVR memerlukan registrasi setiap bit untuk mendapatkan data sedangkan pada pemrograman Arduino menggunakan library bawaan tergantung dari perangkat yang akan dikontrol.

## Soal Formatif

---

1. Jelaskan pengertian komunikasi serial SPI?
2. Carilah 5 Komponen atau Modul yang menggunakan komunikasi SPI?
3. Jelaskan kegunaan masing-masing pin mosi, miso dan sck yang ada pada komunikasi serial SPI?

addcontentslinetocsectionTugas Latihan

## **Tugas Latihan**

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 1 buah komponen motor DC dan 1 buah stick PS. Buatlah dengan algoritma, jika salah satu tombol pada stick ditekan maka motor DC Berputar

## Bab 9

# Interface Serial I2C

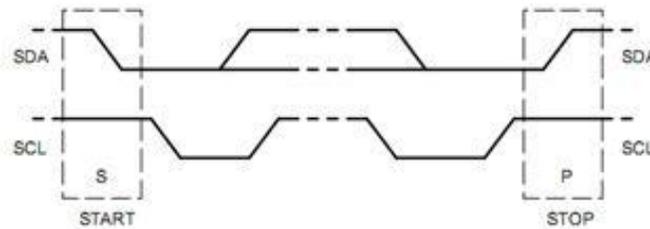
### Tujuan Pembelajaran

---

1. Peserta didik mampu menjelaskan Komunikasi Serial I2C.
2. Peserta Didik mampu membuat rangkaian dan pemrograman innterface serial I2C pada mikrokontroler AVR dan mikrokontroler

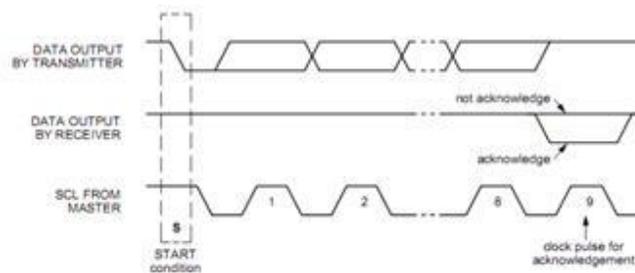
## 9.1 Pengenalan I2C

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master.



Gambar 9.1: Kondisi sinyal start dan stop

Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 1 menjadi 0 pada saat SCL 1. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 0 menjadi 1 pada saat SCL 1. Kondisi sinyal Start dan sinyal Stop seperti tampak pada Gambar 9.1



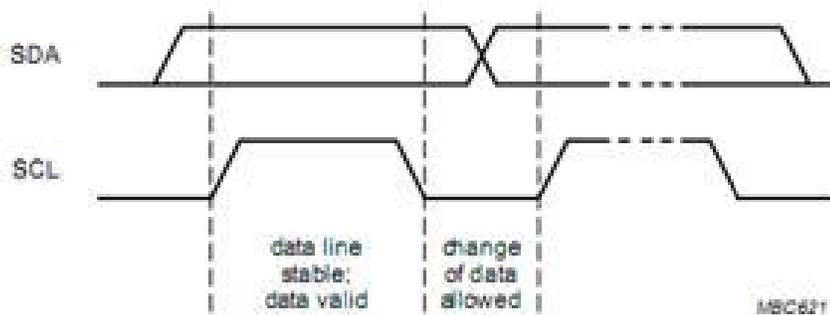
Gambar 9.2: Sinyal ACK dan NACK

Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK. Setelah transfer data oleh master berhasil diterima slave, slave akan menjawabnya dengan mengirim sinyal acknowledge,

yaitu dengan membuat SDA menjadi 0 selama siklus clock ke 9. Ini menunjukkan bahwa Slave telah menerima 8 bit data dari Master. Kondisi sinyal acknowledge seperti tampak pada Gambar 9.2

Dalam melakukan transfer data pada I2C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- Transfer data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- Selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan 1 atau 0 pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal Start atau sinyal Stop.



Gambar 9.3: Trasfer Bit pada I2C bus

## 9.2 RTC sebagai Interface I2C

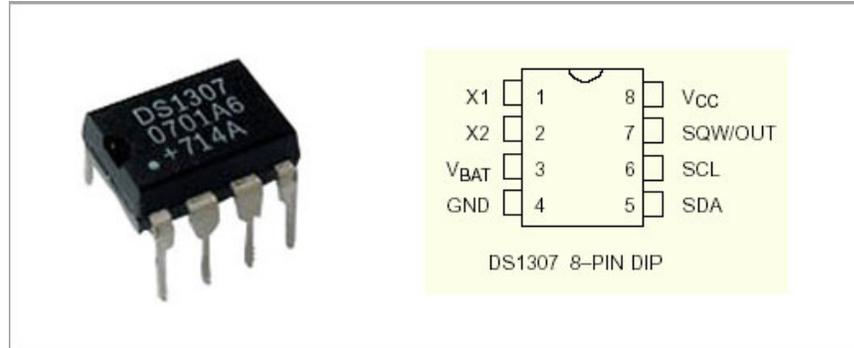
Real Time Clock merupakan suatu IC yang memiliki fungsi sebagai penyimpan waktu dan tanggal. RTC (Real time clock) adalah jam elektronik berupa chip yang dapat menghitung waktu dan menyimpan data waktu tersebut secara real time. RTC dilengkapi dengan baterai sebagai penyedia daya pada IC. RTC cukup akurat sebagai pewaktu (timer) karena menggunakan osilator kristal Beberapa. contoh IC RTC yang ada di pasaran seperti DS12C887, DS1307, DS1302, DS3234. Salah satu IC RTC yang banyak digunakan yaitu adalah DS1307.

DS1307 merupakan Real-time clock (RTC) yang memiliki antarmuka serial Two-wire I2C(Inter Integrated Circuit). Beberapa keistimewaan RTC



Gambar 9.4: Modul Real Time Clock (RTC)

DS1307 yaitu sinyal keluaran berupa gelombangkotak terprogram (Programmable squarewave), deteksi otomatis kegagalan-daya (power-fail), konsumsi daya kurang dari 500mA menggunakan mode baterai cadangan dengan operasional osilator. Tersedia untuk industri dengan ketahanan suhu: -40C hingga +85C. Tersedia dalam kemasan 8 pin DIP atau SOIC.



Gambar 9.5: IC RTC DS1307

Penjelasan dari masing-masing kaki adalah sebagai berikut:

- X1 dan X2 adalah pin yang dihubungkan dengan kristal 32.768 KHz.
- VBAT adalah pin yang dihubungkan masukan baterai +3V.
- GND adalah pin yang dihubungkan Ground.
- SDA adalah pin yang difungsikan sebagai jalur data.

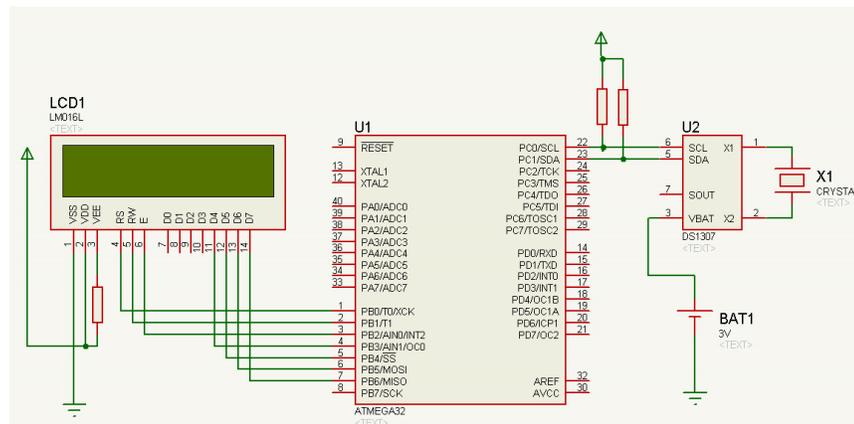
- SCL adalah pin yang berfungsi sebagai jalur clock.
- SQW/OUT adalah pin yang digunakan sebagai keluaran sinyal kotak.
- VCC adalah pin untuk mencatu tegangan 5V.

DS1307 memiliki akurasi hingga tahun 2100. Sistem RTC D81307 memerlukan baterai eksternal 3 volt yang terhubung ke pin Vbat dan ground. Pin X1 dan X2 dihubungkan dengan kristal osilator 32,768 KHz. Sedangkan pin SCL, SDA, dan SQW/OUT di-pull-up dengan resistor (nilainya 1k s.d 10k) ke Vcc. Untuk membaca data tanggal dan waktu yang tersimpan di memori RTC D81307 dapat dilakukan melalui komunikasi serial I2C.

### 9.3 Interface I2C pada Mikrokontroler AVR

Mikrokontroler AVR sudah menyediakan pin dengan fitur SDA dan SCL, pada mikrokontroler jenis ATmega16/32 pin SDA/SCL berada pada PORTC.0 dan PORTC.1. Berikut adalah contoh pengaplikasian interface I2C pada perangkat RTC.

#### 9.3.1 Rangkaian



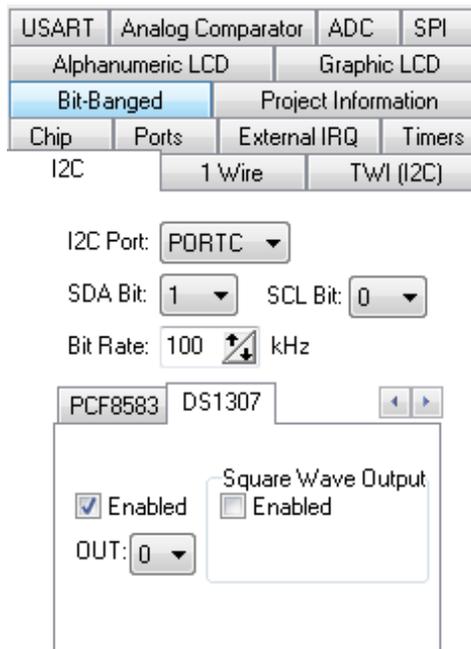
Gambar 9.6: Rangkaian interface RTC

Pada Gambar 9.6, Mikrokontroler ATmega32 diberikan interface berupa RTC dan interface LCD untuk menampilkan data berupa waktu. Prinsip kerjanya adalah mikrokontroler akan melakukan request data setiap waktu

ke RTC dan akan mengumpukan ke LCD untuk menampilkan waktu yang telah diterima dari RTC.

### 9.3.2 Pemrograman

Pada pemrograman mikrokontroler AVR, fitur I2C terlebih dahulu harus diatur PORT dan jenis IC RTC yang digunakan pada codevisionAVR seperti gambar 9.7. Konfigurasi pin disesuaikan dengan gambar rangkaian pada gambar 9.6



Gambar 9.7: Konfigurasi Pin untuk mengaktifkan I2C

Pada pembuatan program RTC, dibutuhkan sebuah tipe data untuk menampung data jam, menit, dan detik. Data yang ditampung tersebut terlebih dahulu dilakukan request ke RTC untuk memperoleh data. Berikut adalah contoh listing program RTC yang menampilkan jam, menit dan detik pada LCD.

Listing Program:

```
/*
Chip type           : ATmega16
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*/

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

// I2C Bus functions
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// Alphanumeric LCD functions
#include <alcd.h>

unsigned char jam, menit, detik, buff[17];

void main(void)
{
    i2c_init();
    rtc_init(0,0,0);
    lcd_init(16);

    lcd_gotoxy(0,0);
    lcd_putsf("Real Time Clock");
    delay_ms(2000);
    lcd_clear();

    //mengatur waktu
    rtc_set_time(23,11,0);

    while (1)
    {
        rtc_get_time(&jam, &menit, &detik);
        lcd_gotoxy(0,0);
        sprintf(buff, "Jam: %d:%d:%d  ", jam, menit, detik);
        lcd_puts(buff);
    };
}
```



## Listing Program:

```
ds1307 $
// Date and time functions using a DS1307 RTC
// and connected via I2C and Wire lib
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 rtc;

char daysOfTheWeek[7][12] = {
  "Sunday",
  "Monday",
  "Tuesday",
  "Wednesday",
  "Thursday",
  "Friday",
  "Saturday"
};

void setup () {
  while (!Serial); // for Leonardo/Micro/Zero

  Serial.begin(57600);
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }

  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }
}

void loop () {
  DateTime now = rtc.now();

  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(" (");
  Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial.print(" ");
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();

Serial.print(" since midnight 1/1/1970 = ");
Serial.print(now.unixtime());
Serial.print("s = ");
Serial.print(now.unixtime() / 86400L);
Serial.println("d");

// calculate a date which is 7 days and 30 seconds into the future
DateTime future (now + TimeSpan(7,12,30,6));

Serial.print(" now + 7d + 30s: ");
Serial.print(future.year(), DEC);
Serial.print('/');
Serial.print(future.month(), DEC);
Serial.print('/');
Serial.print(future.day(), DEC);
Serial.print(' ');
Serial.print(future.hour(), DEC);
Serial.print(':');
Serial.print(future.minute(), DEC);
Serial.print(':');
Serial.print(future.second(), DEC);
Serial.println();

Serial.println();
delay(3000);
}
```

---

## Rangkuman

---

1. Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data.
2. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya.
3. Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 1 menjadi 0 pada saat SCL 1. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 0 menjadi 1 pada saat SCL 1.
4. Contoh Komponen atau alat yang menggunakan komunikasi serial I2C adalah RTC. Real Time Clock merupakan suatu IC yang memiliki fungsi sebagai penyimpan waktu dan tanggal. RTC (Real time clock) adalah jam elektronik berupa chip yang dapat menghitung waktu dan menyimpan data waktu tersebut secara real time. DS1307 merupakan Real-time clock (RTC) yang memiliki antarmuka serial Two-wire I2C(Inter Integrated Circuit).
5. Pemrograman Komunikasi Serial I2C pada mikrokontroler AVR memerlukan konfigurasi awal berupa penentuan PORT sebagai jalur data, sedangkan pada Arduino menggunakan sebuah fitur library "Wire.h".

## Soal Formatif

---

1. Jelaskan pengertian komunikasi serial I2C?
2. Jelaskan apa itu RTC?
3. Jelaskan prinsip kerja RTC?

## **Tugas Latihan**

---

Buatlah rangkaian dan program menggunakan mikrokontroler AVR dan Arduino menggunakan interface 1 buah module LCD dan 1 buah module RTC, dan satu buah buzzer. Buatlah dengan algoritma, menampilkan waktu jam, menit, detik, tanggal, bulan, dan tahun, pada RTC, dan jika masuk waktu jam 12 maka Buzzer akan berbunyi sebagai alarm.

# Daftar Pustaka

- Adrianto, H.: 2013, *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVisionAVR)*, Informatika bandung, Bandung.
- Adrianto, H.: 2015, *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVisionAVR)*, Informatika bandung, Bandung.
- E. Knuth, D.: 2011, *The Art of Computer Programing*, Addison Wesley, United States.
- Fahmizal: 2011, Jenis dan Tipe Motor Servo.  
**URL:** <https://fahmizaleits.wordpress.com/>
- Fauzi, F. A.: 2015, Downloader).  
**URL:** <https://fajarahmadfauzi.wordpress.com/>
- Grupta, M.: 2012, Perkenalan Seputar Mikrokontroler.  
**URL:** <https://guptayp.wordpress.com/>
- Iswanto: 2011, *Belajar Mikrokontroler AT89S51 dengan Bahasa C*, CV Andi Offset, Yogyakarta.
- Kadir, A.: 2015, *Buku Pintar Pemrograman Arduino*, MediaKom, Yogyakarta.
- Marta Dinata, Yuwono: 2015, *Arduino Itu Mudah*, PT Alex Media Komputindo, Jakarta.
- Rachmat, O.: 2012, *Panduan Praktis Membuat Robotik dengan Pemrograman C++*, CV Andi Offset, Yogyakarta.
- Rangkuti, S.: 2011, *Mikrokontroler Atmel AVR Simulasi dan Praktik Menggunakan ISIS Proteus dan CodeVisionAVR*, Informatika Bandung, Bandung.

Saputra, E.: 2015, Materi dasar Algoritma dan Pemrograman.

**URL:** <https://suaebiebi40.wordpress.com/>

Spurianto: 2015, Pengertian Push Button Switch (Saklar Tombol Tekan)).

**URL:** <http://blog.unnes.ac.id/>

Syahrul: 2012, *Mikrokontroler AVR ATmega8535*, Informatika Bandung, Bandung.

syarif, m.: 2014, Serial Pheripheral Interface (SPI) dan Universal Synchronous Asynchronous Receiver Transmitter (USART).

**URL:** <http://muhammadsyarif.ilearning.me/>

Wangready: 2012, Kendali Motor Servo Menggunakan PWM dari Timer).

**URL:** <https://wangready.wordpress.com/>

zoniaelektro.net: 2014, ADC (Analog To Digital Converter).

**URL:** <http://zoniaelektro.net/adc-analog-to-digital-converter/>

# Kunci Jawaban Soal Formatif

## BAB 1. Pengenalan & Sejarah Mikrokontroler

---

1. Mikrokontroler adalah sebuah komputer kecil di dalam satu IC yang berisi CPU, memori, timer, saluran komunikasi serial dan paralel, port input output, ADC, dan mikrokontroler bekerja dengan instruksi program yang telah terinput.
2. Awala dari mikrokontroler dimulai dari pembuatan sebuah sistem yang dapat melakukan perintah alogaritma dasar seperti melakukan penjumlahan, pengurangan, pembagian dan perkalian pada tahun 1671, dan terus dikembangkan mulai dari pembuatan gerbang logika 1850, pembuatan memori 1949, ditemukannya transistor 1948, pertama kali diperkenalkan IC 1959, mikroprocessor pertama 1971, dan pada tahun 1972 dipaketlah mikroprocessor ini sehingga tercipta mikrokontroler. dan terus berkembang sampai sekarang dan melahirkan mikrokontroler yang semakin hari semakin beragam fiturnya.
3. Mikropropesor adalah Central Professing Unit (CPU) didalam single chip, mikro-Komputer adalah mikropropesor yang dihubungkan rangkaian pendukung, jika komponen yang menyusun sebuah mikro-komputer diletakkan bersama didalam single chip silicon maka disebut dengan mikrokontroler
4. Arsitektur RISC merupakan jenis mikrokontroler yang memiliki jumlah instruksi yang terbatas (sedikit) sedangkan arsitektur CISC merupakan jenis mikrokontroler yang memiliki instruksi yang kompleks dan lengkap.
5. (a) Mikrokontroller TinyAVR (ATTiny) adalah mikrokontroller 8 bit.

ATTiny merupakan mikrokontroler avr kecil dan memiliki peripheral yang terbatas.

- (b) Mikrokontroler AT90S adalah mikrokontroler 8 bit jenis lama, merupakan mikrokontroler avr klasik.
- (c) Mikrokontroler Atmega adalah mikrokontroler 8 bit. Atmega memiliki peripheral lebih banyak dibandingkan dengan seri AT-Tiny.
- (d) Mikrokontroler Xmega adalah mikrokontroler 8/16 bit. Xmega memiliki peripheral baru dan canggih dengan untuk kerja, sistem monitoring event dan DMA yang ditingkatkan, serta merupakan pengembangan keluarga AVR untuk pasar low power dan high performance. Dengan adanya fitur DMA (direct memory access) dapat mengurangi kemungkinan terjadinya kemacetan pada saat transfer data. Xmega mendukung kriptografi AES dan DES.
- (e) Mikrokontroler AVR32 adalah mikrokontroler 32 bit, mikrokontroler ini pertama kali dibuat oleh atmel pada tahun 2006. AVR32 menggunakan arsitektur RISC 32 bit, mikrokontroler ini ditujukan untuk bersaing dengan mikrokontroler yang berbasis prosesor ARM mikrokontroler AVR32 tidak memiliki EEPROM internal, sebagai pengganti EEPROM, AVR32 dapat menggunakan SD Card dan MMC.

## **BAB 2. Hardware dan Software Mikrokontroler AVR dan Arduino**

---

1. Mikrokontroler AVR adalah salah satu mikrokontroler dari Atmel yang sangat populer digunakan, dimana pada mikrokontroler ini adalah IC dengan fitur lengkap sebagai otak pengontrol berbagai macam alat mekanik.
2. Arduino adalah salah satu pengembangan dari mikrokontroler AVR yang berbentuk alat pengendali mikro single chip yang open source yang disajikan dalam satu board lengkap dengan bootloader sehingga penggunaannya sangat mudah.
3. Berikut Adalah jenis-jenis bahasa pemrograman untuk mikrokontroler

- (a) Bahasa mesin (machine language/machine code) adalah bahasa yang dipahami oleh komputer. Kode yang terdapat pada bahasa ini berupa kode bilangan biner yang dapat diproses oleh mikrokontroler sehingga sulit dipahami oleh manusia. Bahasa mesin umumnya bersifat spesifik, yaitu tidak lintas jenis mikrokontroler sehingga suatu jenis mikrokontroler memiliki bahasa pemrograman yang berbeda dengan jenis mikrokontroler lainnya. File bahasa mesin memiliki ekstensi .hex, karena untuk kepentingan manusia yang memprogram kode biner dikonversikan menjadi kode bilangan heksadesimal.
- (b) Bahasa Assembly terdiri dari instruksi berupa representasi mnemonic dari instruksi berupa kode bilangan biner dari bahasa mesin. Umumnya mnemonic berupa singkatan tiga atau empat huruf dari kata yang mewakili suatu instruksi. Contohnya instruksi Assembly adalah sebagai berikut :
- SUB adalah kode Assembly untuk instruksi subtract, yaitu mengurangi suatu angka dari angka lain.
  - CBI adalah kode Assembly untuk instruksi Clear Bit I/O, yaitu memberi logika nol pada suatu pin I/O digital.

Instruksi pada Bahasa Assembly diterjemahkan menjadi bahasa mesin oleh perangkat lunak Assembler. Seperti juga bahasa mesin bahasa Assembly juga spesifik untuk suatu jenis mikrokontroler tertentu.

- (c) Bahasa pemrograman tingkat tinggi (high level programming language) memiliki sintaks yang lebih mendekati bahasa manusia. Hal ini menyebabkan bahasa tingkat tinggi lebih mudah untuk dipelajari. Meski demikian bahasa tingkat tinggi umumnya menghasilkan ukuran kode yang lebih besar dibandingkan bahasa Assembly. Bahasa tingkat tinggi memerlukan perangkat lunak compiler (compiler) untuk menerjemahkan kode menjadi bahasa mesin.

4. contoh program bahasa C:

```
1+1=x;
x=2
if(x==2){print="benar"} else {print="salah"}
```

## BAB 3. Interface Data Digital

---

1. Data digital adalah data yang hanya memproses 2 keadaan yaitu ada (1) atau tidak ada (0)
2. Data digital hanya mengenal 2 data saja sedangkan data analog adalah data yang tidak dapat diukur menggunakan dua data saja atau data analog memproses lebih dari dua data.
3. PushButton, karena komponen ini adalah komponen yang dapat memberikan atau mengelolah data pada mikrokontroler, dengan cara ditekan maka komponen ini akan megirimkan sinyal tertentu ke mikrokontroler, dan akan mengelolah data sesuai instruksi program.
4.
  - Calculator Standar, menggunakan sebuah seven segment untuk menampilkan angka dan menggunakan sebuah keypad atau push button untuk mengatur data pada kalkulator tersebut.
  - Tombol Cerdas Cermat, menggunakan sebuah tombol dan led sebagai indikator, jika tombol ditekan maka akan memberikan umpan sinyal ke mikrokontroler untuk menyalakan sebuah LED.

## BAB 4. Interface dengan LCD

---

1. LCD adalah suatu display dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. LCD banyak digunakan sebagai display dari alat-alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya.
2. LCD teks digunakan untuk menampilkan teks atau simbol-simbol tertentu sedangkan LCD grafik memungkinkan untuk menampilkan gambar.
3. Fungsi Pin-pin pada LCD 16x2:  
**Pin 1 dan 2** Merupakan sambungan catu daya, Vss, dan Vdd. Pin Vdd dihubungkan dengan tegangan positif catu daya, sedangkan Vss pada 0 volt atau ground. Meskipun demikian, data menentukan catu 5 Vdc (hanya beberapa mA), menyediakan 6V dan

4,5V yang keduanya bekerja dengan baik, bahkan 3V cukup untuk beberapa modul.

**Pin 3** Merupakan pin kontrol Vcc yang digunakan untuk mengatur kontras display. Idealnya pin ini dihubungkan dengan tegangan yang bisa diubah untuk memungkinkan pengaturan tingkatan kontras display sesuai kebutuhan.

**Pin 4** Merupakan register select (RS), masukan yang pertama dari 3 command control input. Dengan membuat RS menjadi high, data karakter dapat ditransfer dari dan menuju modulnya.

**Pin 5** Merupakan Read/Write (R/W). Cara memfungsikan perintah Write adalah R/W low atau menulis karakter ke modul. R/W high untuk membaca data karakter atau informasi status registernya.

**Pin 6** Merupakan Enable (E). Input ini digunakan untuk transfer aktual perintah-perintah atau karakter antara modul dengan hubungan data. Ketika menulis ke display, data ditransfer hanya pada perpindahan high/low. Namun, ketika membaca dari display, data akan menjadi lebih cepat tersediasetelah perpindahan dari low ke high dan tetap tersedia hingga sinyal low kembali.

**Pin 7 sampai 14** Pin 7 sampai 14 adalah jalur 8 jalur data (D0-D7) di mana data dapat ditransfer ke dan dari display.

**Pin 15 dan 16** Pin 15 atau A (+) mempunyai level DC +5V dan berfungsi sebagai LED backlight +, sedangkan pin 16 atau K (-) memiliki level 0V dan berfungsi sebagai LED backlight.

## BAB 5. Interface Input Analog (ADC)

---

1. Data analog adalah sinyal data dalam bentuk gelombang yang kontinyu, yang membawa informasi dengan mengubah karakteristik gelombang.
2. ADC (Analog To Digital Converter) adalah perangkat elektronika yang berfungsi untuk mengubah sinyal analog (sinyal kontinyu) menjadi sinyal digital.
3. Komponen interface yang menggunakan data ADC pada umumnya berupa sensor, seperti, sensor suhu (LM35), sensor cahaya (LCD/Photodiode), sensor gas, dan sensor-sensor lainnya.

4. Pengukur Suhu Ruangan menggunakan sensor suhu (LM35), alat ini dapat mendeteksi suhu ruangan dan memberikan umpan pada alat lain seperti pendingin ruangan untuk dinyalakan atau dimatikan.

## BAB 6. Interface Output PWM

---

1. Timer/Counter adalah Seperangkat pencacah biner (binary counter) yang terhubung langsung ke saluran data mikrokontroler, sehingga mikrokontroler bisa membaca kondisi pencacah dan bila diperlukan mikrokontroler dapat pula merubah kondisi pencacah tersebut. Saat sinyal clock yang diberikan sudah melebihi kapasitas pencacah, maka pencacah akan memberikan sinyal overflow/limpahan. Limpahan pencacah ini dicatat dalam suatu register.
2. PWM ( Pulse Width Modulation) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (duty cycle) dengan nilai amplitudo dan frekuensi yang tetap.
3. Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo.
4. Pada dasarnya DAC dengan PWM sama, untuk membuat konversi data digital menjadi tegangan analog diperlukan sebuah output pwm, besar tegangan keluaranya akan ditentukan oleh besar dutysycle pada output pwm.

## BAB 7. Interface Serial USART

---

1. Komunikasi Serial merupakan salah satu alternatif yang relatif lebih murah untuk menggantikan komunikasi paralel, karena transfer data paralel menggunakan 8 jalur konduktor/kawat untuk mentransfer 8bit sekaligus.
2. Device yang melakukan konversi paralel ke serial dan konversi serial ke paralel salah satunya disebut universal asynchronous receiver-transmitter (UART).

3. Secara singkat kedua pin pada komunikasi serial USART berfungsi sebagai pin penerima/receiver(RX) dan sebagai pin pengirim/transmitter(TX).
4. Menyalakan Lampu listrik rumah tanpa menggunakan kabel maupun saklar dan dikontrol menggunakan smartphone, hanya dengan menekan subah tombol pada aplikasi smartphone akan langsung memberikan sinyal pada relay yang sudah terkontrol menggunakan mikrokontroler.

## BAB 8. Interface Serial SPI

---

1. Serial Peripheral Interface ( SPI ) adalah sebuah antarmuka bus yang biasa digunakan untuk mengirim data antara mikrokontroler dan perangkat kecil seperti register geser , sensor , dan kartu SD .
2. (a) Downloader SPI  
(b) Stick PS2  
(c) Camera dengan SPI  
(d) Sensor Serial dengan data SPI  
(e) GSM Module
3. Dalam SPI , hanya satu sisi menghasilkan sinyal clock (biasanya disebut CLK atau SCK untuk Serial Clock) . Sisi yang menghasilkan jam disebut master , dan sisi lainnya disebut budak . Selalu hanya satu master ( yang hampir selalu mikrokontroler Anda ) , tetapi bisa ada beberapa budak ( lebih lanjut tentang ini dalam sedikit ) .Ketika data dikirim dari master ke slave , itu dikirim pada baris data yang disebut MOSI , untuk Guru Out / Slave In . Jika slave harus mengirim respon kembali ke master , master akan terus menghasilkan sejumlah diatur sebelumnya clock cycle , dan budak akan menempatkan data ke jalur data ketiga yang disebut MISO , untuk Guru In / Out Slave .

## BAB 9. Interface Serial I2C

---

1. Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data.

2. Real Time Clock merupakan suatu IC yang memiliki fungsi sebagai penyimpan waktu dan tanggal. RTC (Real time clock) adalah jam elektronik berupa chip yang dapat menghitung waktu dan menyimpan data waktu tersebut secara real time. DS1307 merupakan Real-time clock (RTC) yang memiliki antarmuka serial Two-wire 12C(Inter Integrated Circuit).
3. Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 1 menjadi 0 pada saat SCL 1. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari 0 menjadi 1 pada saat SCL 1.

# Glosarium

**AC** Tegangan dengan arus bolak balik.

**Akusisi** masukan, perolehan (tentang data komputer).

**Data** keterangan nyata.

**Embedded** benam bagian yang tidak dapat berdiri sendiri

**Real** nyata

**Amplitudo** ukuran tinggi tegangan dari sinyal.

**System** Tersusun

**AVCC** Tegangan referensi untuk pin analog.

**Baud Rate** jumlah atau banyak data yang dikirimkan dalam waktu yang ditentukan pada komunikasi serial.

**Biner** Jenis bilangan yang memiliki dua rate atau dua keadaan yaout 1 dan 0.

**Bit** banyaknya data 0 atau 1 yang dipaket dalam satu group.

**CPU** Central Processing Unit, pusat proses pengontroloan pada mikrokon-  
troller.

**Counter** data penghitung naik atau turun tergantung penggunaannya.

**DC** tegangan dengan arus searah.

**Duty Sycle** Panjang nilai high pada gelombang PWM.

**Frequensi** jumlah gelombang dalam satuan detik.

**Gerbang Logika (Logic Gate)** aplikasi pengatur data logika berdasarkan input yang diterima dan menghasilkan keluaran yang diinginkan.

**GND (Ground)** pentanahan dalam ilmu kelistrikan.

**IC Integrated Circuit**, komponen elektronika yang memiliki fungsi logika yang beragam.

**Hardware (H/D)** perangkat keras seperti papan rangkaian elektronika.

**I/O Input Output** pada mikrokontroler.

**Memori** Tempat penyimpanan data.

**Open Source** penggunaan perangkat yang bebas dapat dikelola dan dirubah.

**Paralel** Jenis pengiriman data yang memiliki banyak jalur pengiriman.

**Pin** Satu kaki pada mikrokontroler.

**Port** Delapan pin kaki mikrokontroler yang dipaket.

**Serial** Jenis pengiriman data yang berturut turut dalam satu jalur.

**Single Board** Rangkaian elektronika siap digunakan yang terangkai dalam satu papan.

**Software** Perangkat lunak seperti aplikasi pada komputer.

**Listing Program (Source Code)** kode program untuk dikonversi kemudian diupload ke mikrokontroler.

**Sourcing** Arus keluaran pada pin mikrokontroler.

**Timer** Pewaktu dalam mikrokontroler.

**Path** Sebuah titik, percabangan pada line Followe

**Planning** Perencanaan path

**VCC** Tegangan kerja / tegangan referensi pada aplikasi mikrokontroler.

**V<sub>ref</sub>** Tegangan referensi pada rangkaian.

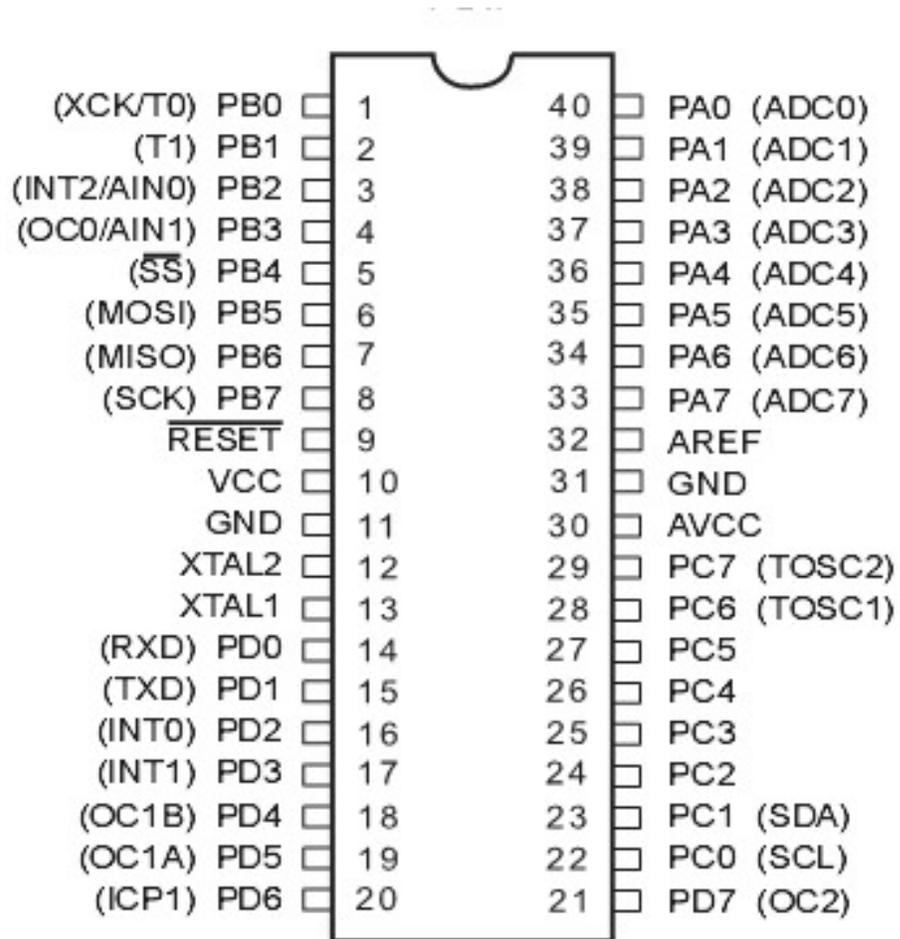
# Kumpulan Data Seet Mikrokontroler

Buku ini menyediakan daftar data seet mikrokontroler yang umum digunakan.

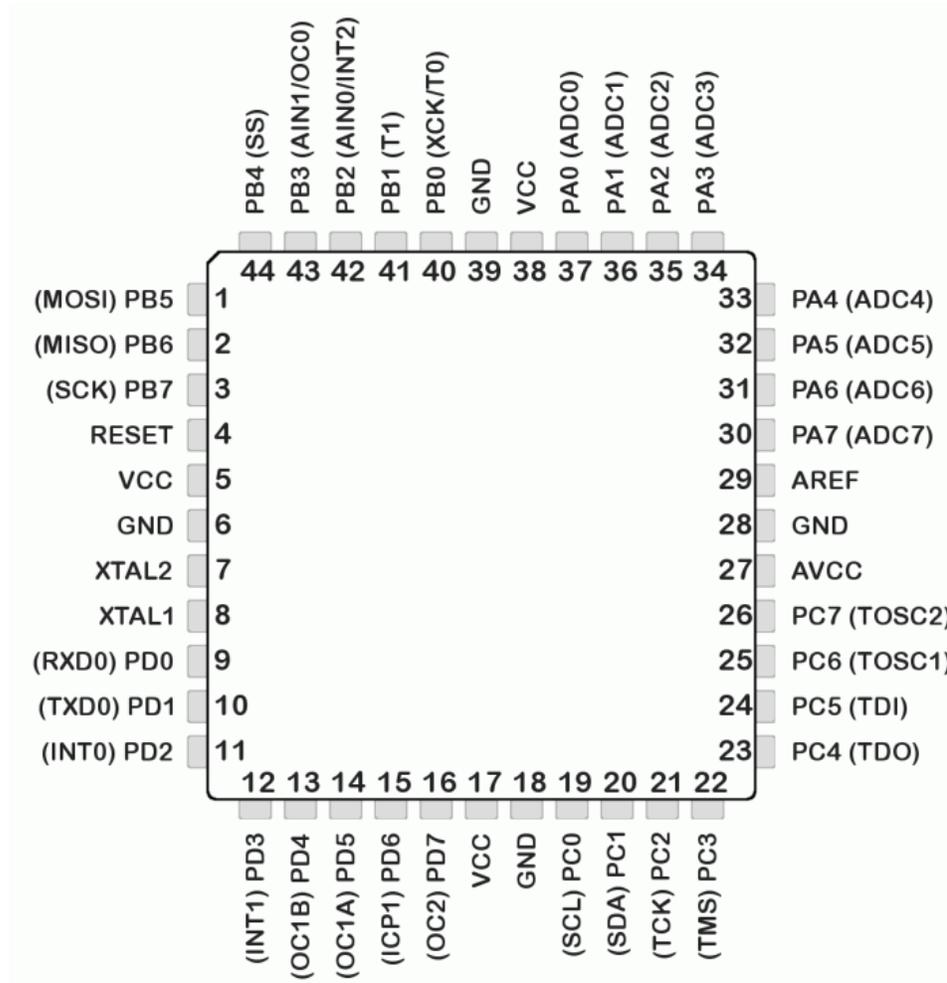
- ATmega8535
- ATmega8535 SMD
- ATmega8
- ATmega8 SMD
- ATmega16 / ATmega32
- ATmega16 SMD
- ArduinoUNO
- ArduinoNANO
- ArduinoMICRO
- ArduinoPROMINI
- ArduinoMEGA
- ArduinoDUE

Memperhatikan sebuah data seet adalah keutamaan dan dapat meminimalisir kesalahan dalam membuat sebuah rangkaian dan membuat sebuah listing program.

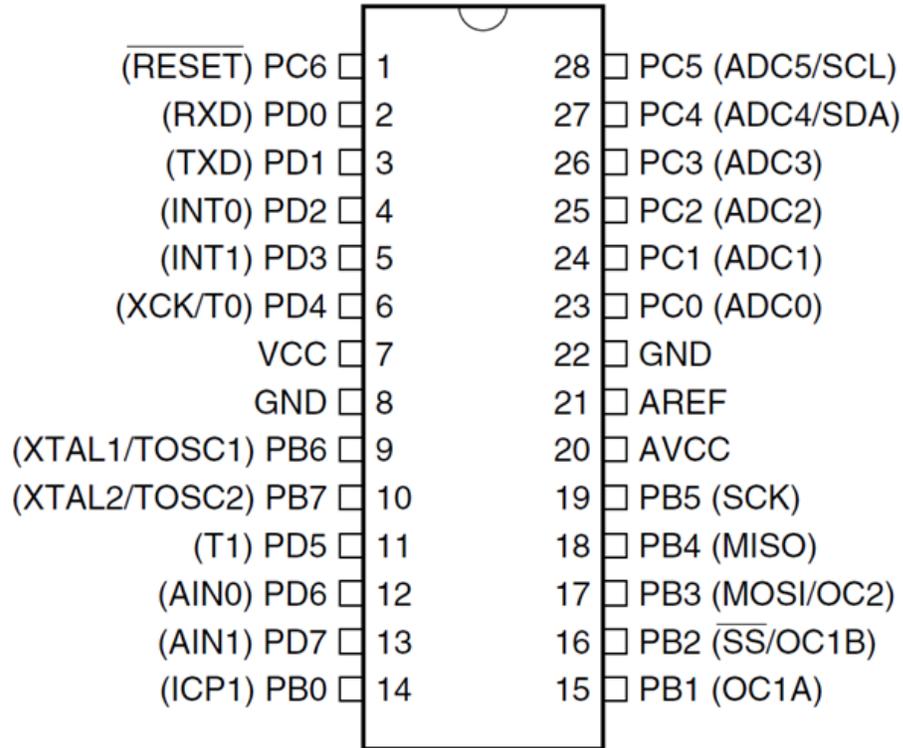
## ATmega8535



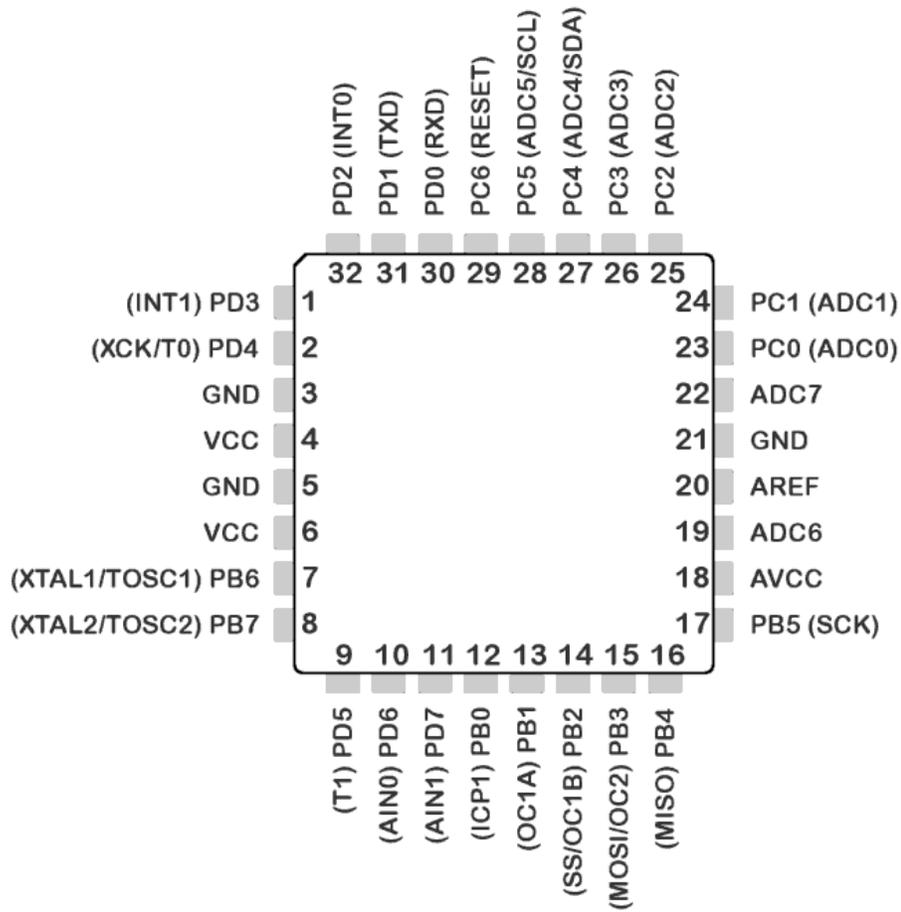
## ATmega8535 TQFP SMD



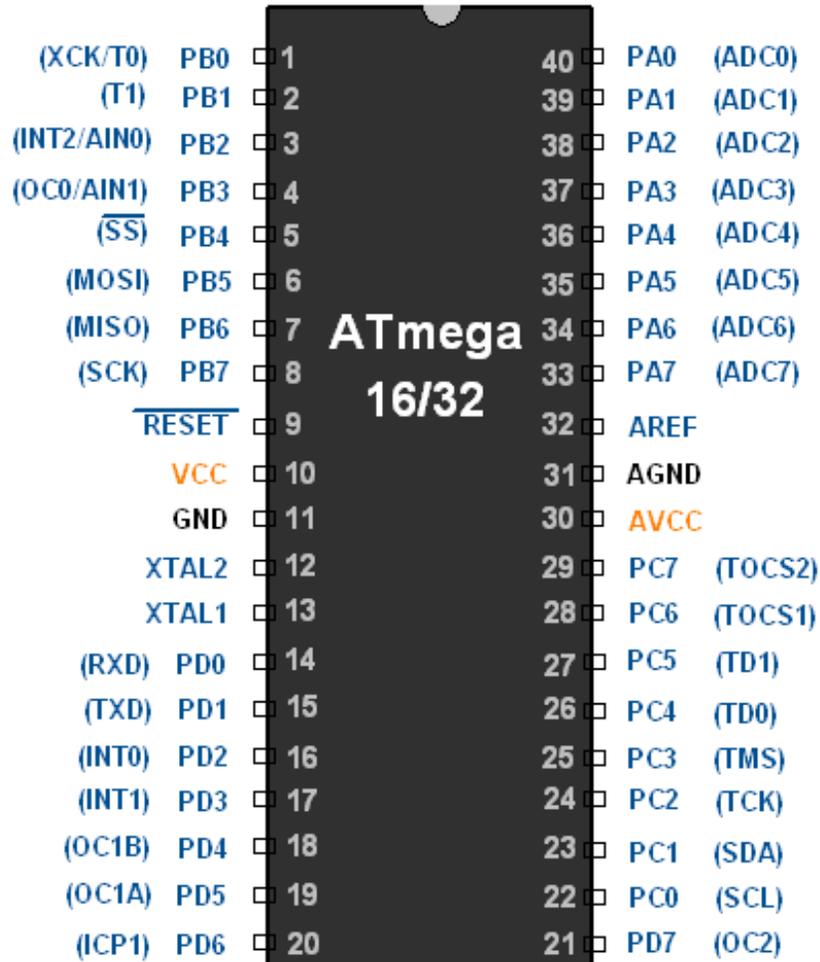
## ATmega8



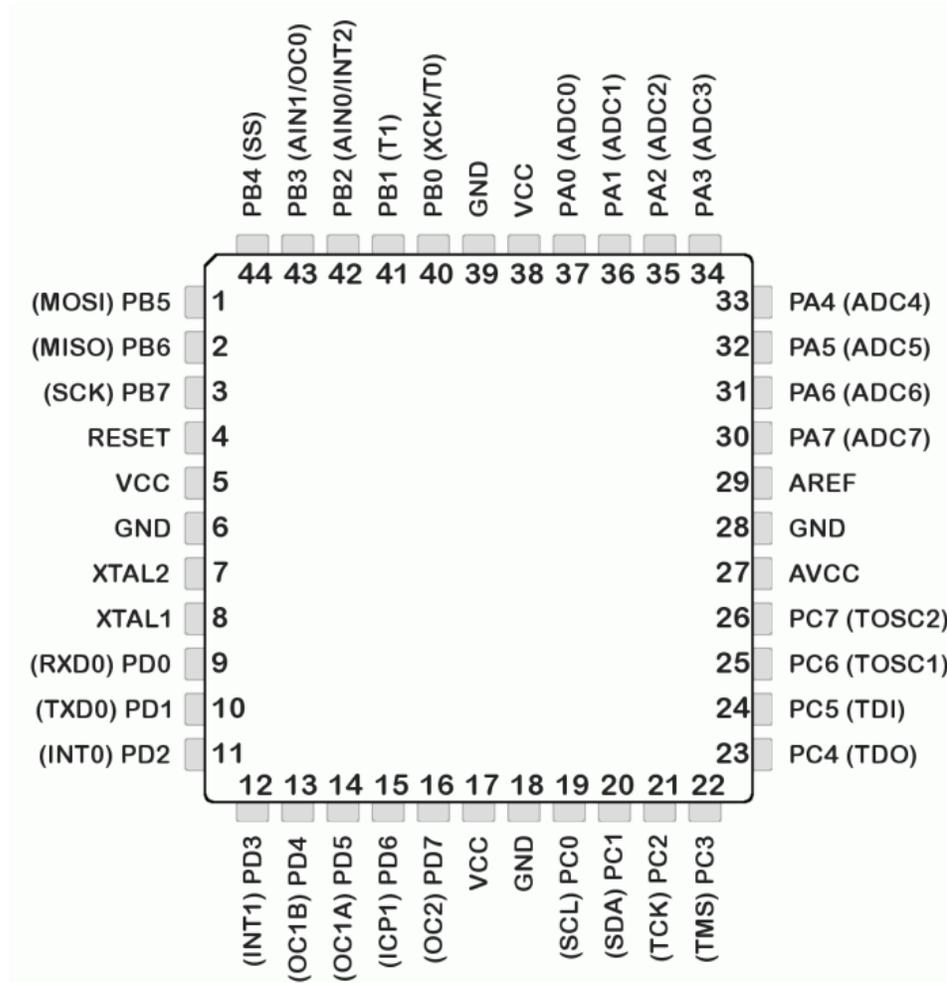
## ATmega8 TQFP SMD



## ATmega16



## ATmega16 TQFP SMD





# Arduino NANO

