

## BAB II

### TINJAUAN PUSTAKA

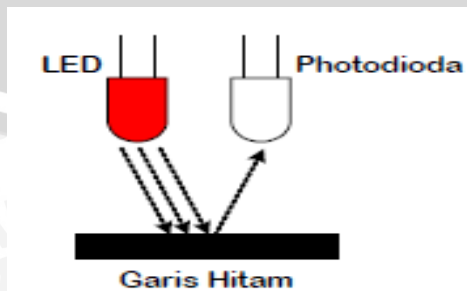
Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan alat ini, maka perlu adanya penjelasan dan uraian teori penunjang yang digunakan dalam penulisan ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Sensor
- Motor DC
- Mikrokontroler AVR ATmega 8
- Mikrokontroler AVR ATmega 16
- Kontroler
- PWM
- Program CodeVision AVR

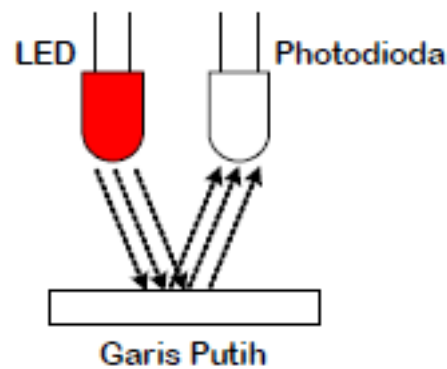
#### 2.1 Sensor

##### 2.1.1 Sensor Cahaya

Sensor *proximity* bisa kita buat sendiri. Prinsip kerjanya sederhana, hanya memanfaatkan sifat cahaya yang akan dipantulkan jika mengenai benda berwarna terang dan akan diserap jika mengenai benda berwarna gelap. Sebagai sumber cahaya kita gunakan LED (*Light Emitting Diode*) yang akan memancarkan cahaya merah. Dan untuk menangkap pantulan cahaya LED, kita gunakan photodiode. Jika sensor berada diatas garis hitam maka photodiode akan menerima sedikit sekali cahaya pantulan. Tetapi jika sensor berada diatas garis putih maka photodiode akan menerima banyak cahaya pantulan. Berikut adalah ilustrasinya :



**Gambar 2.1.** Cahaya Pantulan Sensor Cahaya (sedikit)  
Sumber : Hendawan Soebhakti, 2007



**Gambar 2.2.** Cahaya Pantulan Sensor Cahaya (banyak)  
Sumber : Hendawan Soebhakti, 2007

Sifat dari photodiode adalah jika semakin banyak cahaya yang diterima, maka nilai resistansi diodanya semakin kecil. Dengan melakukan sedikit modifikasi, maka besaran resistansi tersebut dapat diubah menjadi tegangan. Sehingga jika sensor berada diatas garis hitam, maka tegangan keluaran sensor akan kecil, demikian pula sebaliknya.

### 2.1.2 Sensor Ultrasonik

Sensor jarak ultrasonik (PING))) adalah sensor 40 KHz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas untuk mendeteksi jarak suatu objek.

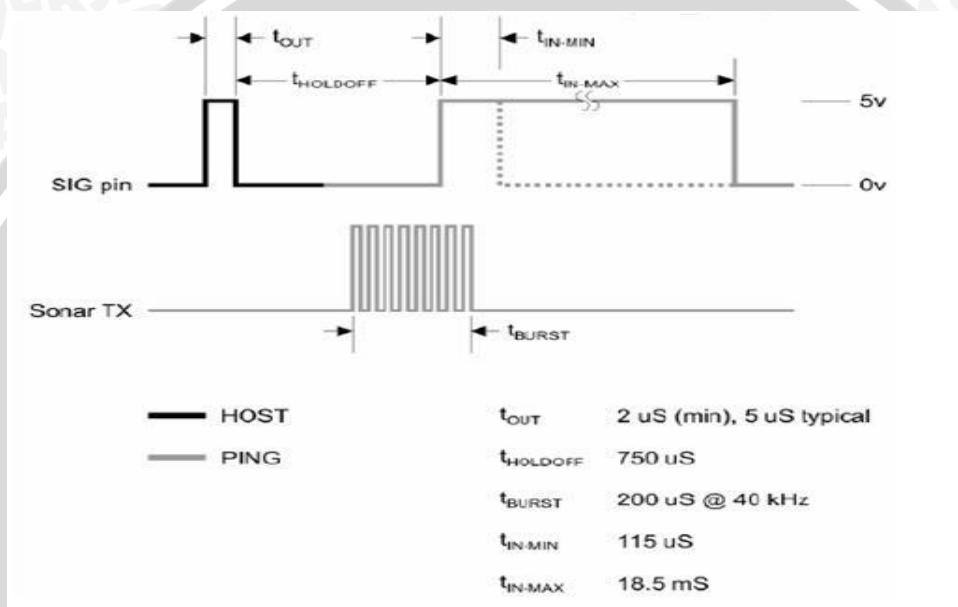


**Gambar 2.3.** Sensor Jarak Ultrasonik (PING)))  
Sumber: Parallax, 2008

Sensor (PING))) mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama  $t = 200 \mu s$  kemudian mendeteksi pantulannya. Sensor (PING))) memancarkan gelombang ultrasonik sesuai dengan kontrol dari

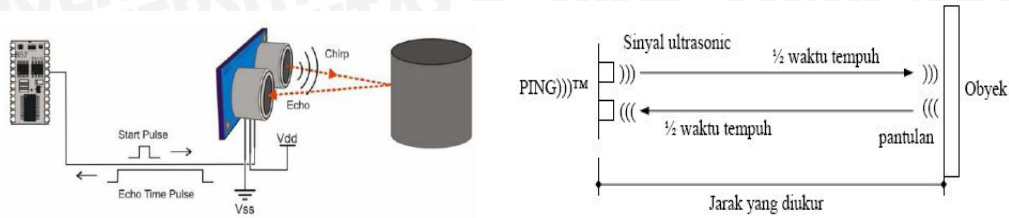
mikrokontroler pengendali ( pulsa trigger dengan tout min 2 us ). Spesifikasi sensor ini :

- Kisaran pengukuran 3cm-3m.
- Input trigger –positive TTL pulse, 2uS min., 5uS tipikal.
- Echo hold off 750uS dari fall of trigger pulse.
- Delay before next measurement 200uS.
- Burst indicator LED menampilkan aktifitas sensor.



**Gambar 2.4.** Diagram Waktu Sensor PING  
Sumber: Parallax, 2008

Dengan diketahui spesifikasi pada sensor PING))), selanjutnya diperoleh prinsip kerja dari sensor tersebut sebagai berikut. Pada dasarnya Sensor PING))) terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah *speaker* ultrasonik dan sebuah mikropon ultrasonik. *Speaker* ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Sensor PING))) mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama tBURST (200  $\mu$ s) kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan tOUT min. 2  $\mu$ s).



**Gambar 2.5.** Prinsip Kerja Sensor PING  
 Sumber: Parallax, 2008

Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per detik, mengenai obyek dan memantul kembali ke sensor. PING))) mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi PING akan membuat *output low* pada pin SIG. Lebar pulsa High (tIN) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah :

$$S = (tIN \times V) \div 2$$

Dimana :

- S = Jarak antara sensor ultrasonik dengan objek yang dideteksi
- V = Cepat rambat gelombang ultrasonik di udara (344 m/s)
- tIN = Selisih waktu pemancaran dan penerimaan pantulan gelombang.

## 2.2 Motor DC

### 2.2.1 Prinsip kerja motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri. Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan(Soemarwanto,1999):

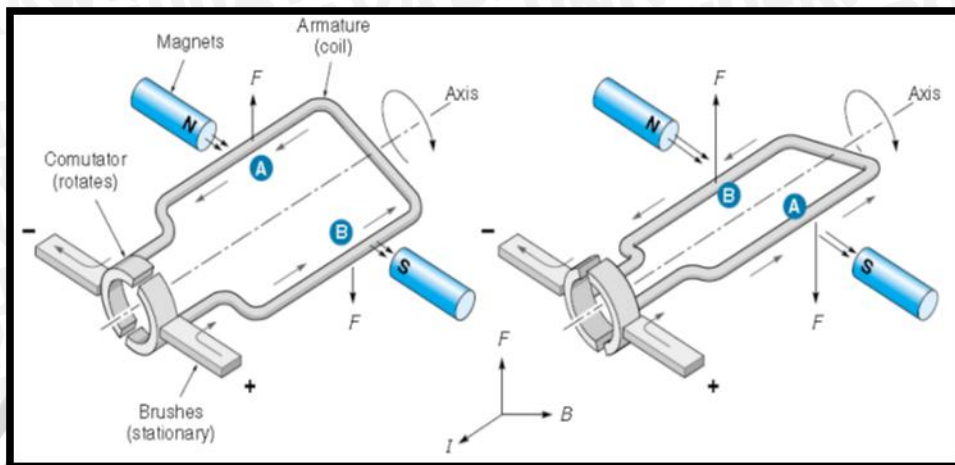
$$F = B.I.L \text{ (Newton)}$$

Dimana :

$B$  = kerapatan fluks magnet (weber)

$L$  = panjang konduktor (meter)

$I$  = arus listrik (ampere)



**Gambar 2.6.** Cara Kerja Motor DC  
Sumber : Kilian, 1996

Gambar 2.6 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a-b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.6 arah fluks magnetic adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel

yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

### 2.3 Mikrokontroler AVR ATmega16

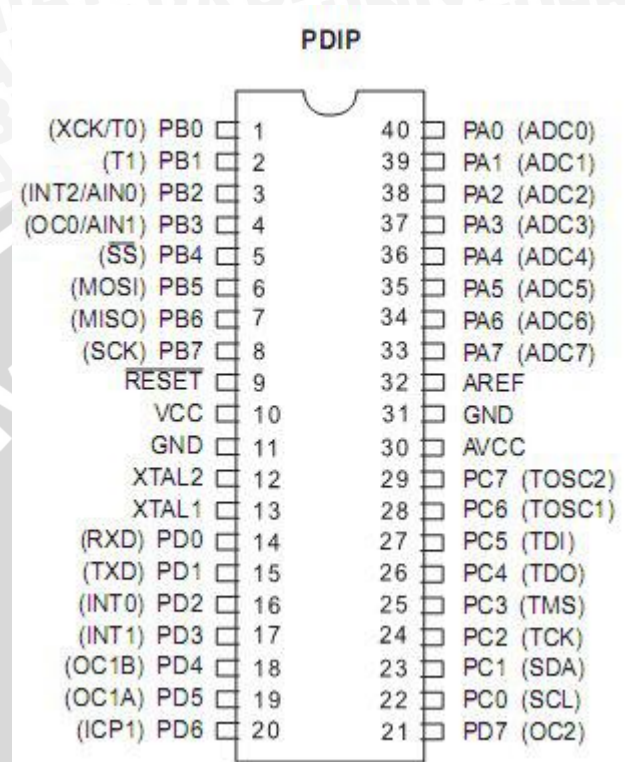
AVR merupakan seri mikrokontroler *Complementary Metal Oxide Semiconductor* (CMOS) 8-bit buatan Atmel berbasis arsitektur RISC (*Reduced Instruction Set Computer*).

Fitur AVR ATMEGA16 antara lain:

1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi dengan konsumsi daya rendah
2. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz
3. Memiliki kapasitas *Flash* memori 16 Kbyte, EEPROM 512 Byte dan SRAM 1 Kbyte
4. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*
5. CPU yang terdiri dari 32 buah *register*
6. Unit interupsi dan eksternal
7. *Port* USART untuk komunikasi serial
8. Fitur *peripheral*
  - Tiga buah *Timer/Counter* dengan kemampuan perbandingan (*compare*)
  - Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*
  - Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare* dan *Mode Capture*
  - Real Time Counter* dengan *Oscillator* tersendiri
  - Empat kanal PWM
  - 8 kanal ADC
  - 8 *Single-ended Channel* dengan keluaran hasil konversi 8 dan 10 resolusi (register ADCH dan ADCL)
  - 7 *Diferrential Channel* hanya pada kemasan *Thin Quad Flat Pack* (TQFP)
  - 2 *Differential Channel* dengan *Programmable Gain*
  - Antarmuka *Serial Peripheral Interface* (SPI) Bus

- Watchdog Timer dengan *Oscillator Internal*
- On-chip Analog Comparator

### 9. Non-volatile program memory



**Gambar 2.7.** Konfigurasi Pin ATmega16  
Sumber: ATMEL, 2010

Konfigurasi pin mikrokontroler ATmega 16 meliputi :

- VCC : Suplai tegangan.
- GND : Ground.
- Port A (PA7..PA0)

Bandar A berfungsi sebagai *input* analog pada konverter A/D. Bandar A juga sebagai suatu bandar I/O 8-bit dua arah, jika A/D konverter tidak digunakan. Pena

- Pena Bandar dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). Bandar A *output* buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Ketika pena PA0 ke PA7 digunakan sebagai input dan secara eksternal ditarik rendah, pena-pena akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pena

Bandar A adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port B (PB7..PB0)

Bandar B adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar B output buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pena Bandar B yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena Bandar B adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port C (PC7..PC0)

Bandar C adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar C output buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pena bandar C yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena bandar C adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port D (PD7..PD0)

Bandar D adalah suatu bandar I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Bandar D output buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pena bandar D yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pena Bandar D adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- RESET (*Reset input*)
- XTAL1 (*Input Oscillator*)
- XTAL2 (*Output Oscillator*)
- AVCC adalah pena penyedia tegangan untuk bandar A dan Konverter A/D.
- AREF adalah pena referensi analog untuk konverter A/D.

#### 2.4 Mikrokontroler AVR ATMega 8

Secara umum, Mikrokontroler adalah suatu IC (*integrated circuit*) yang memiliki kemampuan untuk membuat keputusan berdasarkan sinyal dari luar



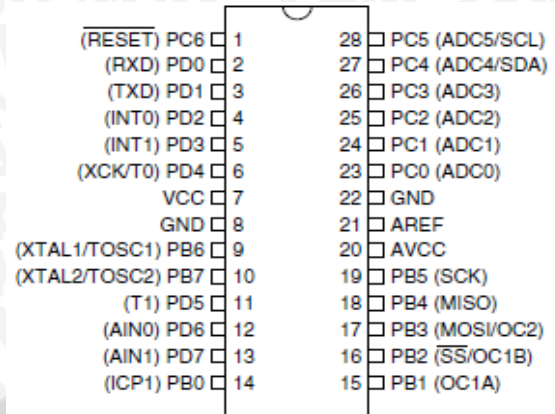
(input), dan berdasarkan algoritma yang dibentuk menjadi suatu program tertentu. Pada dasarnya mikrokontroler memiliki mikroprosesor, *timer*, *counter*, perangkat I/O dan internal memori.

Sedangkan menurut *datasheet* ATmega 8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K Bytes *ISP Flash*. Dengan mengeksekusi instruksi dalam *clock* yang sangat cepat, ATmega 8 memiliki *throughput* mendekati 1 MIPS per MHz, yang memungkinkan perancang sistem mengoptimalkan konsumsi daya terhadap kecepatan pemrosesan.

Sebagai suatu sistem kontrol mikrokontroler ATmega 8 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. Secara umum konfigurasi yang dimiliki mikrokontroler ATmega 8 adalah sebagai berikut :

- Sebuah CPU 8 bit dengan menggunakan teknologi dari Atmel.
- Jalur dua arah (*bidirectional*) yang digunakan sebagai saluran masukan atau keluaran yang dikontrol oleh *register* DDR.
- Dua buah *timer/counter* 8 bit dan sebuah *timer/counter* 16 bit.
- 23 jalur *programmable* I/O (input/output).
- *Analog to Digital Converter (ADC)* 10-bit dan *Analog comparator* di dalam chip.
- Osilator internal dan rangkaian pewaktu.
- Sebuah komunikasi serial USART yang dapat diprogram.
- Sebuah *master/slave* serial SPI yang dapat diprogram.
- Sebuah *Two Wire Serial Interface*.

Masing-masing pin mikrokontroler ATmega 8 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing pin mikrokontroler ATmega 8, perancangan aplikasi mikrokontroler ATmega8 akan lebih mudah dan maksimal. ATmega 8 mempunyai 28 pin, susunan masing-masing pin ditunjukkan dalam Gambar 2.8.



**Gambar 2.8.** Konfigurasi Pin ATmega8  
Sumber: ATMEL, 2010

Fungsi kaki-kaki PIN dalam ATmega8 sesuai dengan *data sheet* ATmega 8 antara lain sebagai berikut:

- *Port B* (Pin B0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port B* diantaranya adalah : *Port B0* {ICP (*Timer/counter input capture pin*)}, *Port B1* {OC1A (*Timer/counter 1 output compare A match output*)}, *Port B2* yang bisa digunakan sebagai SS (*SPI slave select input*) atau OC1B (*Timer/counter 1 output compare B match output*), *Port B3* yang bisa digunakan sebagai MOSI (*SPI busmaster output/slave input*) atau OC2 (*timer/counter 2 compare match output*), *Port B4* {MISO (*SPI bus master input/slave output*)}, *Port B7* {SCK (*SPI bus serialclock*)}, *Port B5* {SCK (*SPI bus serialclock*)}, *Port B6* (XTAL1 & TOSC1 (*Timer Oscilator pin1*)), dan *Port B7* (XTAL2 & TOSC2 (*Timer Oscilator pin2*)).
- *Port C* (Pin C0..5), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port C* adalah sebagai ADC (*input ADC channel 0..5*), selain itu pada beberapa *Port* memiliki fungsi lain antara lain *Port C4* (SDA (*Two-Wire serial bus data input/output line*)), dan *Port C5* (SCL (*Two-Wire serial bus clock line*)).
- *Pin C6* memiliki fungsi *RESET*, merupakan saluran dua masukan untuk mereset mikrokontroler dengan cara memberi masukan logika rendah.
- *Port D* (Pin D0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus dari *Port D* diantaranya adalah : *Port D0* {RXD (*USART input pin*)}, *Port D1* {TXD (*USART output pin*)}, *Port D2*

{INT0 (*Eksternalinterrupt 0 input*)}, Port D3 {INT1 (*Eksternalinterrupt 1 input*)}, Port D4 (T0 (*timer/counter0eksternalcounter input*) & XCK (*USART eksternalclock input/output*), Port D5 (T1 (*timer/counter eksternal counter input*)), Port D6 (AIN0 (*Analog comparator positive input*)) dan Port D7 (AIN1 (*Analog comparator negative input*)).

- Pin 7 VCC, merupakan masukan untuk catu daya positif DC sebesar 5 volt.
- Pin 8 GND, merupakan *ground* dari seluruh rangkaian.
- Pin B6 dan Pin B7 (XTAL2 dan XTAL1), merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan internal maupun eksternal.
- Pin 21 AREF, merupakan pin referensi analog untuk masukan ADC.
- Pin 22 GND, merupakan *ground* dari ADC.
- Pin 20 AVCC, merupakan catu untuk perangkat ADC.

## 2.5 Kotroler

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Disamping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata K.,1997).

Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respons transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

Dalam sistem pengendalian terdapat 2 macam *loop*:

1. Pengendalian dengan *loop* terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpam balikan untuk dibandingkan dengan masukan.

2. Pengendalian dengan *loop* tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

**2.5.1 Kontroler Proporsional**

Kontroler proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah:

$$m(t) = K_p \cdot e(t) \dots \dots \dots (2-1)$$

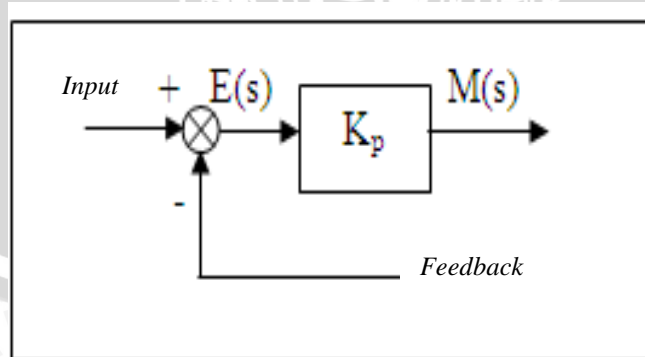
atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p \dots \dots \dots (2-2)$$

Di mana  $K_p$  adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K.,1997).

Diagram blok kontroler proporsional ditunjukkan dalam Gambar 2.9.



**Gambar 2.9.** Diagram Blok Kontroler Proporsional  
 Sumber: Ogata K., 1997



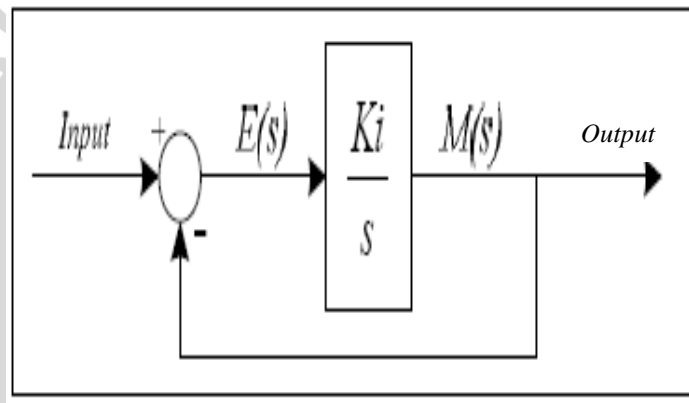
### 2.5.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap pada kontroler proporsional sebelumnya. Pada kontroler dengan aksi integral, harga keluaran kontroler  $m(t)$  diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak  $e(t)$ .

Jadi,

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots \dots \dots (2-3)$$

Jika harga  $e(t)$  diduakalikan, maka harga  $m(t)$  berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga  $m(t)$  tetap stasioner. Aksi kontrol integral seringkali disebut *control reset* (Ogata K.,1997). Gambar 2.10 menunjukkan diagram blok kontroler integral.



Gambar 2.10. Diagram Blok Kontroler Integral  
Sumber: Ogata K., 1997

### 2.5.3 Kontroler Diferensial

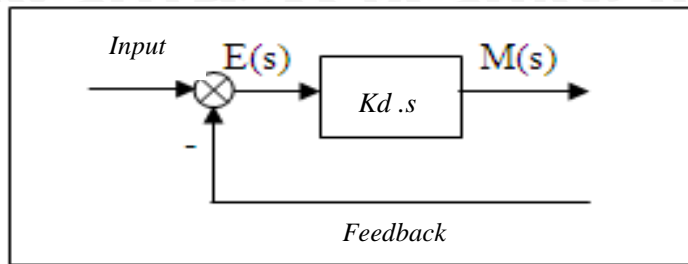
Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata K.,1997).

Kontroler diferensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah :

$$\frac{M(s)}{E(s)} = K_d \cdot s \dots \dots \dots (2-4)$$



Gambar 2.11 menunjukkan diagram blok kontroler diferensial.



**Gambar 2.11.** Diagram Blok Kontroler Diferensial  
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

**2.5.4 Kontroler Proporsional Integral Diferensial (PID)**

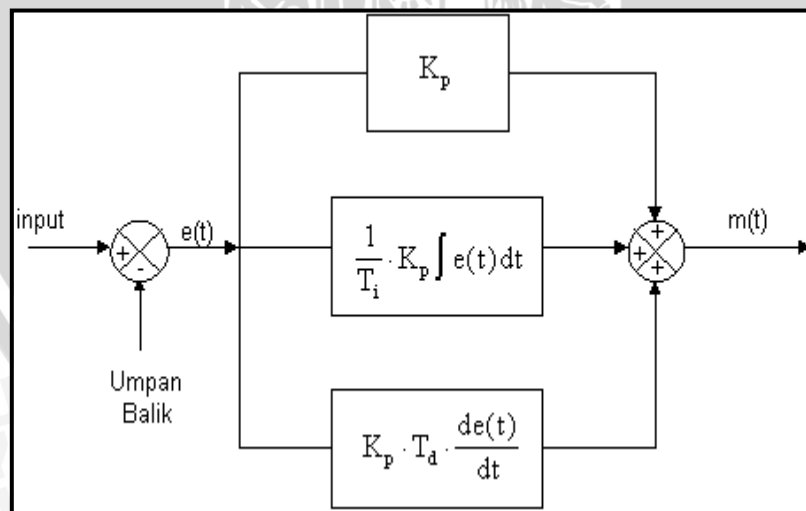
Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan kelebihan dari masing-masing kontroler. Elemen-elemen P, I, dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut :

$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \dots\dots\dots(2-5)$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\frac{M(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \dots\dots\dots(2-6)$$

Gambar 2.12 menunjukkan diagram blok kontroler PID



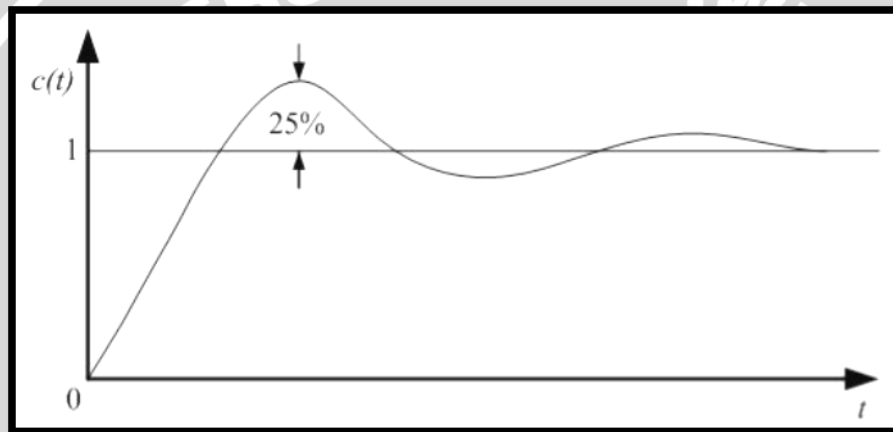
**Gambar 2.12.** Diagram Blok Kontroler PID  
 Sumber: Ogata K., 1997



### 2.5.5 Metode Kontroler Proporsional Integral Diferensial (PID)

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional  $K_p$ , waktu integral  $T_i$ , dan waktu derivatif  $T_d$  berdasarkan karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan. (Ogata, K., 1997)

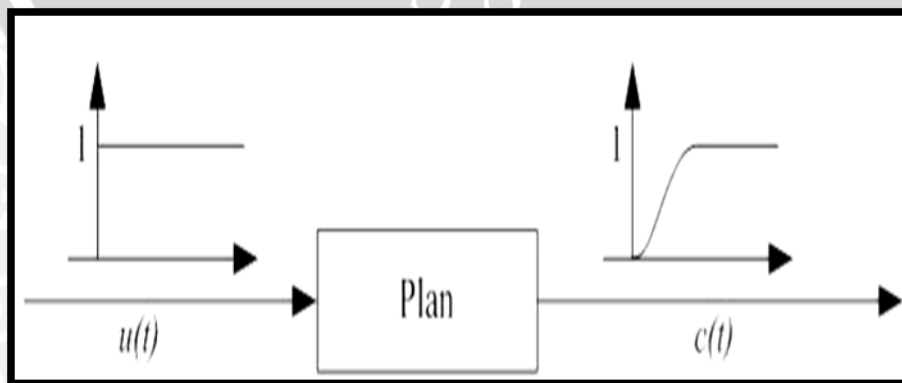
Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, ditunjukkan dengan Gambar 2.13



**Gambar 2.13.** Kurva Respon Unit Step yang Menunjukkan 25% *Maximum Overshoot*  
Sumber: Ogata, K., 1997

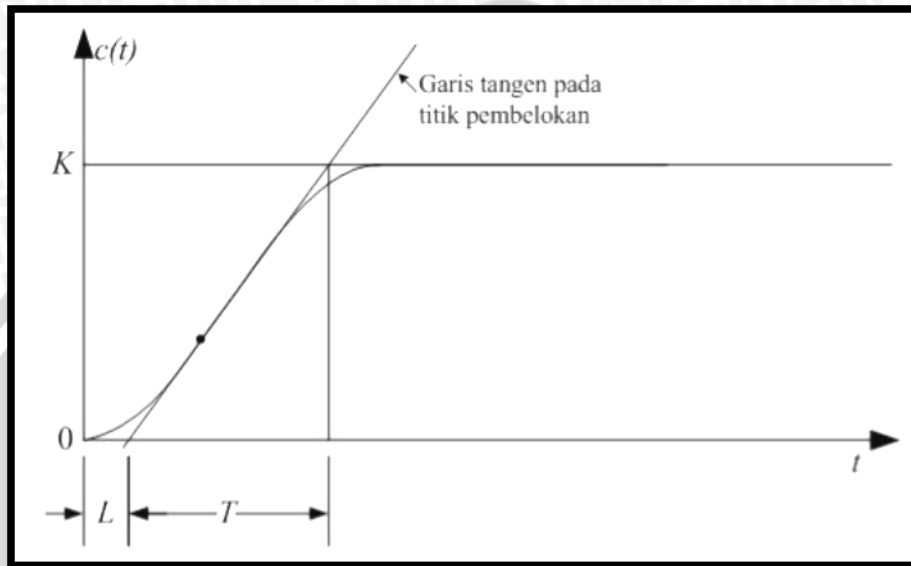
#### 1). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dengan Gambar 2.14



**Gambar 2.14.** Respon Plant Terhadap Masukan Berupa Unit Step  
Sumber: Ogata, K. 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti ditunjukkan dengan Gambar 2.15 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



Gambar 2.15. Kurva Respon yang Berbentuk S  
 Sumber: Ogata, K. 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda  $L$  dan konstanta waktu  $T$ . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu  $t$  dan sumbu  $c(t) = K$ , seperti yang telah ditunjukkan dalam Persamaan (2.7). Fungsi alih  $C(s)/U(s)$  dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \dots\dots\dots(2.7)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari  $K_p$ ,  $T_i$  dan  $T_d$  berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata, K., 1997)





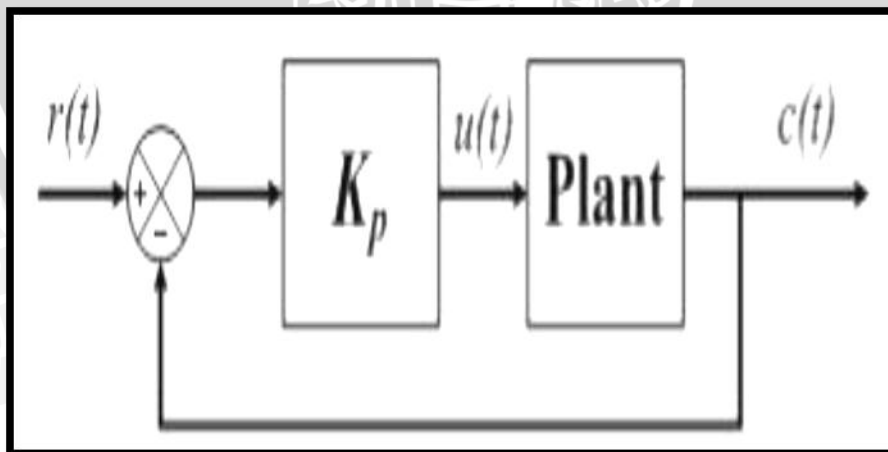
**Tabel 2.1** Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step dari Plan  
Sumber: Ogata, K. 1997

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5 L$

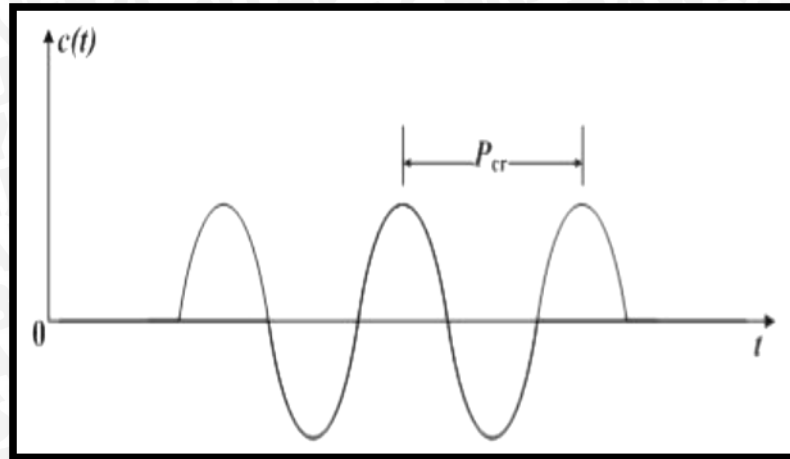
2). Metode Kedua

Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat  $T_i = \infty$  dan  $T_d = 0$ . Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis  $K_{cr}$ , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai  $K_p$  manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis  $K_{cr}$  dan periode  $P_{cr}$  dapat ditentukan.

Diagram blok sistem *loop* tertutup dengan kontroler proporsional ditunjukkan dengan Gambar 2.16 dan untuk osilasi berkesinambungan dengan periode  $P_{cr}$  ditunjukkan dengan Gambar 2.17.



**Gambar 2.16.** Sistem Loop Tertutup dengan Kontroler Proporsional  
Sumber: Ogata, K., 1997



**Gambar 2.17.** Osilasi Berkesinambungan dengan Periode  $P_{cr}$   
 Sumber : Ogata, K., 1997

Ziegler dan Nichols menyarankan penyetelan nilai parameter  $K_p, T_i, T_d$  dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2. (Ogata, K., 1997)

**Tabel 2.2** Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain  $K_{cr}$  dan Critical Period  $P_{cr}$

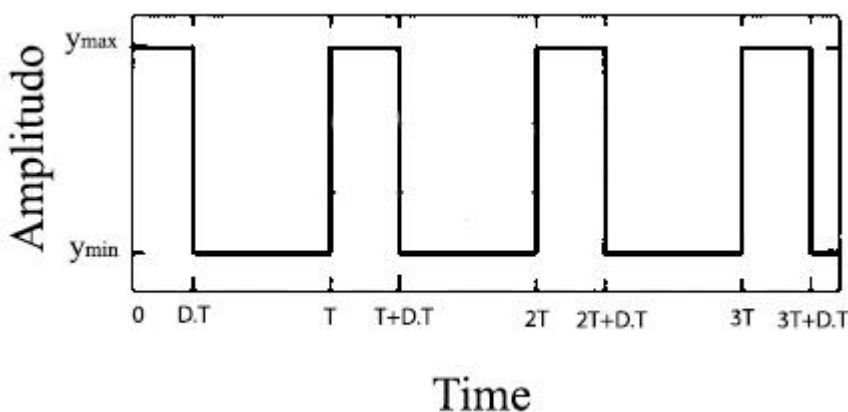
Sumber: Ogata, K., 1997

Tipe Kontroler	$K_p$	$T_i$	$T_d$
P	$0.5 K_{cr}$	$\infty$	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

## 2.6 PWM (*Pulse Width Modulation*)

PWM (*Pulse Width Modulation*) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0% - 100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam gambar 2.18 berikut:



**Gambar 2.18.** Gelombang kotak yang memiliki  $y_{max}$ ,  $y_{min}$  dan  $D$   
 Sumber: Heri Andrianto, 2008 : 137

$$DutyCycle = \frac{T_{on}}{T} \times 100(\%) \dots\dots\dots(2-8)$$

Dengan:

- Ton = Periode logika tinggi
- T = Periode keseluruhan

$$Vdc = DutyCycle \times Vcc(V) \dots\dots\dots(2-9)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$fOn = \frac{f_{clock}}{N.256} (Hz) \dots\dots\dots(2-10)$$

**2.7 Program CodeVision AVR**

CodeVisionAVR merupakan sebuah cross-compiler C, *Integrated Development Environment* (IDE), dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, dan XP.

Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*.

*File object* COFF hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan C, dengan pengamatan variabel, menggunakan *debugger* Atmel AVR Studio.

IDE mempunyai fasilitas internal berupa software AVR Chip In-System Programmer yang memungkinkan pengguna untuk melakukan transfer program kedalam chip mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 *programmers/development boards*.

Untuk keperluan *debugging* sistem *embedded*, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal.

Selain *library* standar C, CodeVisionAVR juga mempunyai *library* tertentu untuk:

- a) Modul LCD *alphanumeric*
- b) Bus I2C dari Philips
- c) Sensor Suhu LM75 dari *National Semiconductor*
- d) *Real-Time Clock*: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas *Semiconductor*
- e) Protokol 1-Wire dari Maxim/Dallas *Semiconductor*
- f) Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas *Semiconductor*
- g) Termometer/Termostat DS1621 dari Maxim/Dallas *Semiconductor*
- h) EEPROM DS2430 dan DS2433 dari Maxim/Dallas *Semiconductor*
- i) SPI
- j) *Power Management*
- k) *Delay*
- l) Konversi ke Kode Gray

CodeVisionAVR juga mempunyai *Automatic Program Generator* bernama CodeWizardAVR, yang mengizinkan pengguna untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- a) *Set-up* akses memori eksternal
- b) Identifikasi sumber *reset* untuk *chip*
- c) Inisialisasi port *input/output*

- d) Inisialisasi interupsi eksternal
- e) Inisialisasi *Timer/Counter*
- f) Inisialisasi *Watchdog-Timer*
- g) Inisialisasi UART (USART) dan komunikasi *serial* berbasis *buffer* yang digerakkan oleh interupsi
- h) Inisialisasi Pembanding Analog
- i) Inisialisasi ADC
- j) Inisialisasi Antarmuka SPI
- k) Inisialisasi Antarmuka Two-Wire
- l) Inisialisasi Antarmuka CAN
- m) Inisialisasi Bus I2C, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307
- n) Inisialisasi *Bus 1-Wire* dan Sensor Suhu DS1820, DS18S20
- o) Inisialisasi modul LCD

