

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Pengendalian kecepatan motor DC (*direct curent*) dapat dilakukan dengan tiga metode, diantaranya pengendalian fluks medan, pengendalian tahanan jangkar, pengendalian tegangan jangkar.

Pengendalian motor DC (*direct curent*) dapat dilakukan dengan metode PWM (*Pulse Width Modulation*) dengan menaikkan atau menurunkan *duty cycle* 5 % hingga naik sampai 95 % dengan periode gelombang 1 ms (Eko, 2008).

Pengendalian kecepatan motor DC (*direct curent*) dengan metode PWM (*Pulse Width Modulation*) juga dapat dikombinasikan dengan remote kontrol yang dapat menjadi masukan pada mikrokontroler sebagai pembangkit PWM, dimana remote dapat diatur menjadi 8 (delapan) tingkat kecepatan yang masing-masing mewakili nilai *duty cycle* yang dibangkitkan oleh mikrokontroler (Muklas dkk, 2006).

Pengendalian kecepatan putar motor DC (*direct curent*) dengan metode pengaturan lebar pulsa atau PWM (*Pulse Width Modulation*) juga dapat dibangkitkan melalui perubahan nilai temperatur sebagai referensi terhadap perubahan *duty cycle* yang harus dibangkitkan untuk mempercepat putaran motor ataupun memperlambat putaran motor tergantung dari besarnya temperatur (Hamdani, 2010).

Pengendalian motor DC (*direct curent*) shunt dengan mikrokontroler M68HC11 pada pengendalian motor kalang tertutup dengan metode arus jangkar terkontrol menyebabkan motor lebih cepat mencapai kondisi beban penuh dibandingkan dengan sistem kalang terbuka (Agus, 2006).

Pengendalian motor DC (*direct curent*) juga dapat dikendalikan dengan metode *fuzzy logic* berbasis mikrokontroler, dimana *fuzzy logic* dapat digunakan sebagai masukan PWM yang dapat mengendalikan kecepatan motor DC (Nasrul, 2009)

Perbedaan penelitian penulis dengan penelitian sebelumnya adalah pada metode pembangkitan PWM, penulis menggunakan mikrokontroler ATmega 32 sebagai pembangkit PWM sekaligus sebagai kontrol otomatis PWM berupa kontrol *timer*, dimana *timer* berfungsi sebagai pewaktu untuk lamanya motor bekerja pada kecepatan

tertentu serta mengaplikasikannya pada *Laboratory Shaker*. Sehingga pada penelitian ini penulis beri judul “Aplikasi Motor DC Shunt pada *Laboratory Shaker* Menggunakan Metode PWM (*Pulse Width Modulation*) Berbasis Mikrokontroler ATmega 32”.

2.2. *Laboratory Shaker*

Laboratory shaker adalah alat yang digunakan untuk mengaduk atau mencampur suatu larutan dengan larutan yang lain sehingga bersifat homogen. Alat ini sangat penting mengingat didalam laboratorium sering melakukan kegiatan pencampuran larutan, dan jika Pencampuran larutan dilakukan secara manual akan kurang efisien dalam waktu maupun tenaga. Disamping itu ada beberapa larutan yang berbahaya untuk disentuh. Maka dari itu, alat ini menambah *safety* dari pengguna di laboratorium.

Prinsip kerja *Laboratory shaker* adalah motor berputar dengan arah tertentu dan dengan bantuan beberapa sistem mekanik maka sebuah plat akan bergerak maju mundur, di atas plat inilah beberapa larutan di letakan sehingga larutan dapat dicampur dengan gerakan maju mundur. Gambar 2.1 merupakan contoh *Laboratory shaker*.



Gambar 2.1. *Laboratory shaker*

2.3. Motor DC (*Direct Current*)

Motor DC (*direct current*) adalah motor listrik yang memerlukan tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor DC (*direct current*) disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Bagian Atau Komponen Utama Motor DC diantaranya:

1. Kutub medan.

Motor DC (*direct current*) sederhana memiliki dua kutub medan yaitu kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet

2. *Current Elektromagnet* atau Dinamo.

Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC (*direct current*) yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi

3. Commutator.

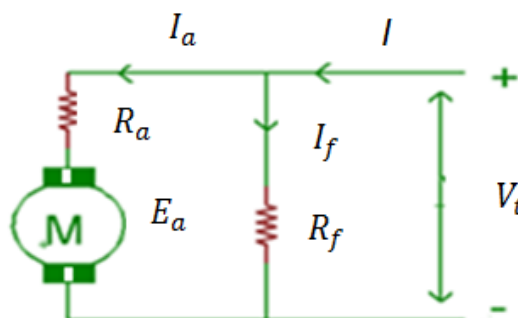
Komponen ini untuk transmisi arus antara dinamo dan sumber daya

2.3.1. Jenis-jenis motor DC (*Direct Current*)

Berdasarkan hubungan lilitan penguat magnet terhadap lilitan jangkar, motor DC (*Direct Current*) dibedakan menjadi :

1) Motor DC Shunt

Motor DC Shunt mempunyai kecepatan hampir konstan. Pada tegangan jepit konstan, motor ini mempunyai putaran yang hampir konstan walaupun terjadi perubahan beban. gambar 2.2 merupakan rangkaian ekuivalen motor DC shunt.



Gambar 2.2. Rangkaian ekuivalen motor DC Shunt

Pada motor DC Shunt, kumparan medan dihubungkan paralel dengan jangkar, seperti terlihat pada gambar 2.2 sehingga dapat dirumuskan sebagai berikut:

$$E_a = V_t - I_a R_a \quad (2.1)$$

$$V_t = I_f R_f \quad (2.2)$$

$$I = I_a + I_f \quad (2.3)$$

$$n = \frac{E_a}{C\phi} \quad (2.4)$$

Keterangan :

E_a : Tegangan induksi (Volt)

V_t : Tegangan terminal motor DC

R_a : Tahanan jangkar

I_a : Arus jangkar

R_f : Tahanan medan shunt

I_f : Arus medan shunt

ϕ : Fluks magnet/kutub

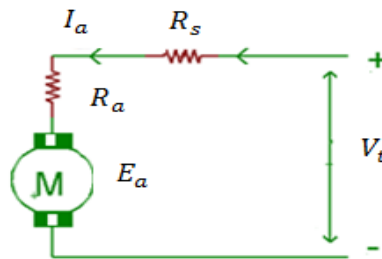
n : Kecepatan putar (rpm)

C : Konstanta motor

Putaran berbanding lurus dengan tegangan sumber (tegangan terminal motor) dan arus jangkar.

2) Motor DC Seri

Merupakan motor DC yang mempunyai kecepatan putar yang tidak konstan jika beban tinggi maka putaran akan lemah. Gambar 2.3 merupakan rangkaian ekuivalen motor DC seri



Gambar 2.3. Rangkaian ekuivalen motor DC seri

Pada motor DC seri, kumparan medan dihubungkan seri dengan jangkar, seperti terlihat pada gambar 2.3 sehingga dapat dirumuskan sebagai berikut:

$$E_a = V_t - I_a (R_a + R_s) \quad (2.5)$$

$$I = I_a = I_s \quad (2.6)$$

$$n = \frac{E_a}{C\phi} \quad (2.7)$$

Keterangan :

E_a : Tegangan induksi (Volt)

V_t : Tegangan terminal motor DC

R_a : Tahanan jangkar

I_a : Arus jangkar

R_s : Tahanan medan seri

I_s : Arus medan seri

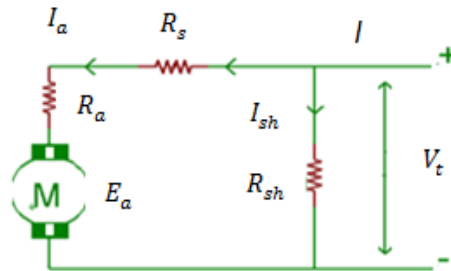
ϕ : Fluks magnet/kutub

n : Kecepatan putar (rpm)

C : Konstanta motor

3) Motor DC Kompon

Motor DC jenis ini mempunyai sifat seperti motor DC Shunt dan Seri tergantung lilitan mana yang kuat. Seperti terlihat pada gambar 2.4 dan 2.5



Gambar 2.4. Rangkaian ekuivalen motor DC kompon panjang

$$E_a = V_t - I_a(R_a + R_s) \quad (2.8)$$

$$V_t = I_{sh}R_{sh} \quad (2.9)$$

$$I = I_a + I_{sh} \quad (2.10)$$

$$n = \frac{E_a}{C\phi} \quad (2.11)$$

Keterangan :

E_a : Tegangan induksi (Volt)

V_t : Tegangan terminal motor DC

R_a : Tahanan jangkar

I_a : Arus jangkar

R_s : Tahanan medan seri

I_f : Arus medan shunt

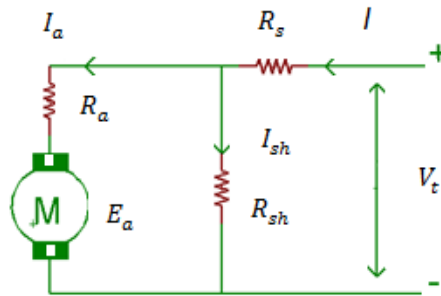
R_{sh} : Tahanan medan shunt

I_{sh} : Arus medan shunt

ϕ : Fluks magnet/kutub

n : Kecepatan putar (rpm)

C : Konstanta motor



Gambar 2.5. Rangkaian ekuivalen motor DC kompon pendek

$$E_a = V_t - IR_s - I_a R_a \quad (2.12)$$

$$V_t = I_{sh} R_{sh} + IR_s \quad (2.13)$$

$$I = I_a + I_f \quad (2.14)$$

Keterangan :

E_a : Tegangan induksi (Volt)

V_t : Tegangan terminal motor DC

R_a : Tahanan jangkar

I_a : Arus jangkar

R_s : Tahanan medan seri

I_f : Arus medan shunt

R_{sh} : Tahanan medan shunt

I_{sh} : Arus medan shunt

ϕ : Fluks magnet/kutub

n : Kecepatan putar (rpm)

C : Konstanta motor

2.4. Pengendalian Motor DC (*Direct Current*)

Pengendalian kecepatan motor dapat dilakukan dengan memanipulasi tiga variabel secara manual maupun otomatis. Arus jangkar (I_a) ditentukan oleh besarnya beban yang sedang dicatu oleh jangkar motor, oleh sebab itu tidak digunakan untuk pengendalian kecepatan motor. Terdapat 3 metode dasar pengendalian kecepatan yaitu :

1. Pengendalian Fluks medan
2. Pengendalian Tahanan Jangkar
3. Pengendalian Tegangan Jangkar

2.4.1. Pengendalian Fluks Medan

Arus Medan dan juga fluks medan dalam motor shunt telah siap diubah dengan mengatur tahanan geser medan yang dihubungkan secara seri dengan medan shunt. Dengan demikian menaikkan tahanan dalam rangkaian medan akan menyebabkan penurunan dalam fluksi medan yang pada akhirnya akan menaikkan kecepatan, begitu pula sebaliknya jika menurunkan tahanan dalam menyebabkan berkurangnya kecepatan motor

2.4.2. Pengendalian Tahanan jangkar

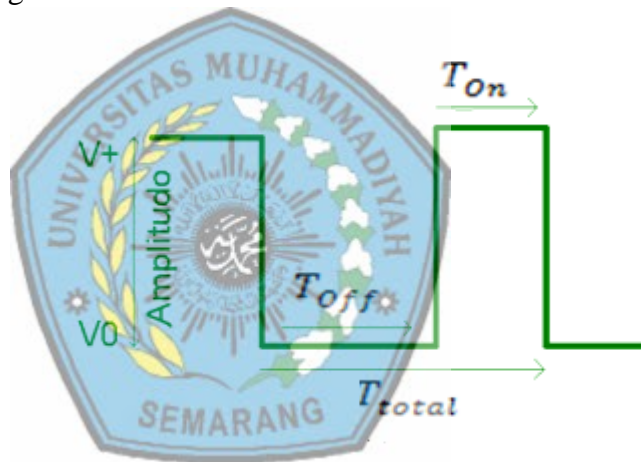
Tahanan jangkar motor dapat diubah dengan menambahkan tahanan variabel yang dihubungkan seri dengan jangkar . Bila tahanan seri dinaikkan, maka tegangan jangkar akan berkurang sehingga menyebabkan turunnya kecepatan motor. Demikian halnya jika bila tahanan seri diturunkan maka kecepatan motor akan bertambah. Ini biasanya digunakan untuk pengendalian kecepatan motor seri

2.4.3. Pengendalian Tegangan Jangkar

Mengubah tegangan yang diberikan pada rangkaian jangkar merupakan model pengendalian Tegangan Jangkar, salah satu pengendalian ini adalah dengan metode pwm (*Pulse Width Modulation*)

2.5. PWM (Pulse Width Modulation)

PWM (Pulse Width Modulation) secara umum adalah sebuah cara memanipulasi lebar pulsa dalam suatu periode untuk mendapatkan tegangan rata-rata yang berbeda. Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar Pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%), bentuk pulsa pwm seperti pada gambar 2.6



Gambar 2.6. Bentuk Pulsa PWM

$$T_{total} = T_{On} + T_{Off}$$

$$V_{Out} = \frac{T_{On}}{T_{total}} \times V_{in}$$

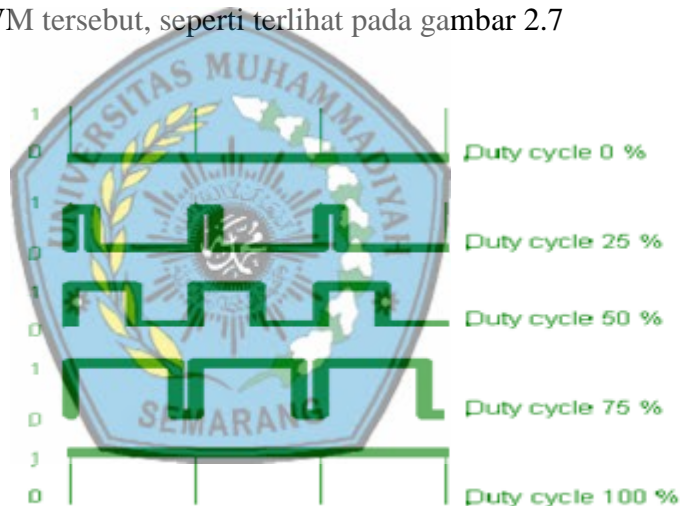
T_{Off} : Waktu pulsa off / low

T_{On} : Waktu pulsa on / high

V_{Out} : Tegangan keluaran

V_{in} : Tegangan masukan

PWM (*Pulse Width Modulation*) merupakan salah satu teknik untuk mendapatkan signal analog dari sebuah piranti digital. Sebenarnya Sinyal PWM dapat dibangkitkan dengan banyak cara, dapat menggunakan metode analog dengan menggunakan rangkaian op-amp atau dengan menggunakan metode digital. Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Contoh, suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak 256 variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0 – 100% dari keluaran PWM tersebut, seperti terlihat pada gambar 2.7

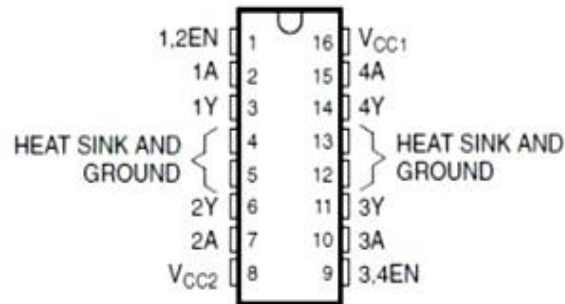


Gambar 2.7. *Duty cycle* pwm 8 bit

2.6. *Driver* Motor L293D

IC L293D adalah IC yang didesain khusus sebagai *driver* motor DC (*Direct Current*) dan dapat dikendalikan dengan rangkaian TTL maupun mikrokontroler. Motor DC (*Direct Current*) yang dikontrol dengan *driver* IC L293D dapat dihubungkan ke ground maupun ke sumber tegangan positif karena di dalam *driver* L293D sistem *driver* yang digunakan adalah *totem pool* yaitu dalam 1 unit chip IC L293D terdiri dari 4 buah *driver* motor DC yang berdiri sendiri sendiri dengan kemampuan mengalirkan arus 1 Ampere tiap drivernya. Sehingga dapat digunakan untuk membuat *driver H-bridge*

untuk 2 buah motor DC (*Direct Current*). Konstruksi pin *driver* motor DC IC L293D terlihat pada gambar 2.8.



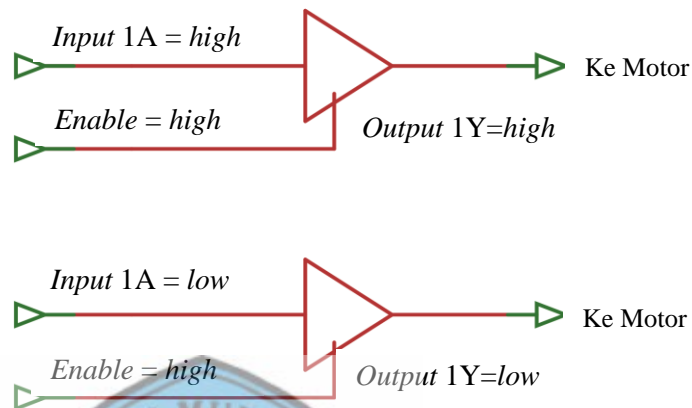
Gambar 2.8. Konfigurasi pin L293D

Keterangan masing-masing pin L293D

1. Pin EN (*Enable*, EN1.2, EN3.4) berfungsi untuk memungkinkan *driver* menerima perintah untuk menggerakkan motor DC (*Direct Current*) .
2. Pin In (*Input*, 1A, 2A, 3A, 4A) adalah pin *input* sinyal kendali motor DC (*Direct Current*)
3. Pin Out (*Output*, 1Y, 2Y, 3Y, 4Y) adalah jalur *output* masing-masing *driver* yang dihubungkan ke motor DC (*Direct Current*)
4. Pin VCC (VCC1, VCC2) adalah jalur *input* tegangan sumber *driver* motor DC (*Direct Current*), VCC1 adalah jalur *input* sumber tegangan rangkaian kontrol *dirver* dan VCC2 adalah jalur *input* sumber tegangan untuk motor DC yang dikendalikan.
5. Pin GND (Ground) adalah jalu yang harus dihubungkan ke ground, pin GND ini ada buah yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil.

2.6.1. Prinsip kerja L293D

Prinsip kerja L293D dapat dilihat pada gambar 2.9 dan di bawah ini



Gambar 2.9. Prinsip kerja L293D

Keterangan gambar:

1. *Output 1Y* L293D akan *high* ketika *input 1A=high* dan *enable=high*, sehingga motor DC mendapatkan *supply* tegangan.
2. *Output 1Y* L293D akan *low* ketika *input 1A=low* meskipun *enable=high*, sehingga motor DC tidak mendapatkan *supply* tegangan.

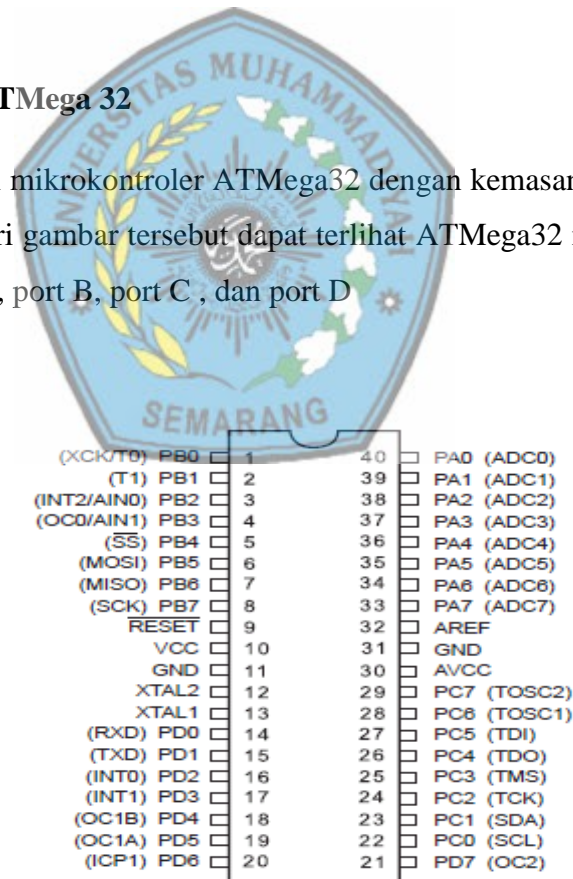
2.7. Mikrokontroler ATmega 32

Seperti mikroprosesor pada umumnya, secara *internal* mikrokontroler ATmega32 terdiri atas unit-unit fungsionalnya *Arithmetic and Logical Unit* (ALU), himpunan *register* kerja, *register* dan *dekoder instruksi*, dan pewaktu beserta komponen kendali lainnya. Berbeda dengan mikroprosesor, mikrokontroler menyediakan memori dalam serpih yang sama dengan prosesornya (*in chip*), Mikrokontroler ini menggunakan arsitektur *Harvard* yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan. Secara garis besar mikrokontroler ATmega32 terdiri dari :

1. 32 kb ISP *flash program memory*
2. 2 kb SRAM
3. 1 kb EEPROM
4. Frekuensi osilator maksimal 16 MHz
5. 32 pin *input/ output*
6. 8 *channel* 10 bit ADC, analog comparator
7. Satu 16 bit *timer/ counter* dan dua 8 bit *timer/ counter*
8. *Watchdog timer*, RTC, 4 *channel* PWM, *master/ slave* SPI, TWI
9. *Programmable* USART
10. *Package* 40 PDIP

2.7.1. Konfigurasi ATmega 32

Konfigurasi pin mikrokontroler ATmega32 dengan kemasan 40-pin dapat dilihat pada Gambar 2.10. Dari gambar tersebut dapat terlihat ATmega32 memiliki 8 pin untuk masing-masing port A, port B, port C, dan port D



Gambar 2.10. Konfigurasi pin ATmega32

2.7.2. Deskripsi ATmega32

1. Pin 1 sampai 8 (PortB) merupakan port parallel 8 bit dua arah yang dapat digunakan untuk *general purpose* dan *special featur*.
2. Pin 9 (*Reset*)
3. Pin 10 (VCC) dihubungkan ke Vcc (2,7 – 5,5 Volt)
4. Pin 11 dan 31 (GND) dihubungkan ke Vss atau Ground
5. Pin 12 (XTAL 2) adalah pin masukkan ke rangkaian osilator *internal* Sebuah osilator kristal atau sumber osilator luar dapat digunakan.
6. Pin 13 (XTAL 1) adalah pin keluaran ke rangkaian osilator *internal*. Pin ini dipakai bila menggunakan osilator kristal.
7. Pin 14 sampai 21 (Port D) adalah 8-bit dua arah (*bi-directional I/O*) port dengan *internal pull-up resistors* digunakan untuk *general purpose* dan *special feature*
8. Pin 22 sampai 29 (Port C) adalah 8-bit dua arah (*bi-directional I/O*) port dengan *internal pull-up resistors* digunakan untuk *general purpose* dan *special feature*
9. Pin 30 adalah Avcc. pin penyuplai daya untuk portA dan A/D converter dan dihubungkan ke Vcc. Jika ADC digunakan maka pin ini dihubun
10. Pin 32 adalah A REF, pin yang berfungsi sebagai referensi untuk pin analog jika A/D Converter digunakan
11. Pin 33 sampai 40 (PortA) adalah 8-bit dua arah (*bi-directional I/O*) port dengan *internal pull-up resistors* digunakan untuk *general purpose*

2.7.3. Fungsi khusus port ATmega32

1. Port A (pin 33 sampai 40) dapat di gunakan sebagai:

Tabel 2.1.Fungsi khusus port A ATmega32

Pin/kaki	Port ATmega 32	Fungsi
33	PA7	<i>Input ADC chanel 7</i>
34	PA6	<i>Input ADC chanel 6</i>
35	PA5	<i>Input ADC chanel 5</i>
36	PA4	<i>Input ADC chanel 4</i>
37	PA3	<i>Input ADC chanel 3</i>
38	PA2	<i>Input ADC chanel 2</i>
39	PA1	<i>Input ADC chanel 1</i>
40	PA0	<i>Input ADC chanel 0</i>

2. Port B (pin 1 sampai 8) dapat di gunakan sebagai:

Tabel 2.2.Fungsi khusus port B ATmega32

Pin/kaki	Port ATmega 32	Fungsi
1	PB0	<i>T0 (timer/counter0, external counter input)</i>
2	PB1	<i>T1 (timer/counter1,external counter input)</i>
3	PB2	<i>INT2 (exsternal interups2 input)</i>
4	PB3	<i>OC0 (timer/counter output compare)</i>
5	PB4	<i>SSI (slave select input)</i>
6	PB5	<i>MOSI (master output/slave input)</i>
7	PB6	<i>MISO (master input/slave output)</i>
8	PB7	<i>SCK (serial clock)</i>

3. Port D (pin 14 sampai 21) dapat di gunakan sebagai:

Tabel 2.3.Fungsi khusus port D ATmega32

Pin/kaki	Port ATmega 32	Fungsi
14	PD0	RXD (USART <i>input pin</i>)
15	PD1	TXD (USART <i>output pin</i>)
16	PD2	INT0 (<i>external interups 0 input</i>)
17	PD3	INT1 (<i>external interups 1 input</i>)
18	PD4	OC1B (<i>timer/counter1output compare B</i>)
19	PD5	OC1A (<i>timer/counter1output compareA</i>)
20	PD6	ICP1 (<i>timer/counter 1input capture pin</i>)
21	PD7	OC2 (<i>timer/counter 2 output compare 2</i>)

4. Port C (pin 22 sampai 29) dapat di gunakan sebagai:

Tabel 2.4.Fungsi khusus port C ATmega32

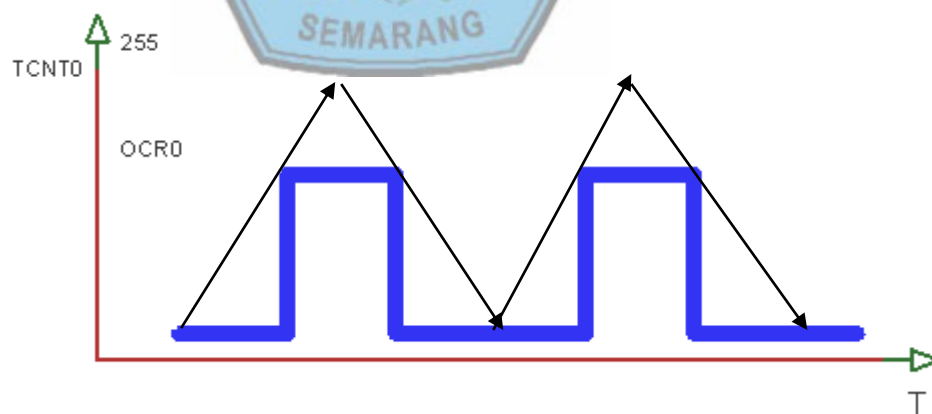
Pin/kaki	Port ATmega 32	Fungsi
22	PC0	SCL (<i>two-wire serial bus clock line</i>)
23	PC1	SDA (<i>serial bus data input/output</i>)
24	PC2	TCK (JTAG <i>test clock</i>)
25	PC3	TMS (JTAG <i>test mode select</i>)
26	PC4	TDO (JTAG <i>test data output</i>)
27	PC5	TDI (JTAG <i>test data input</i>)
28	PC6	TOSC1 (<i>timer oscilator timer pin1</i>)
29	PC7	TOSC2 (<i>timer oscilator timer pin2</i>)

2.8. Sistem Input dan Output

Kelebihan mikrokontroler AVR ATmega32 dibandingkan dengan generasi sebelumnya 8051 (misal AT89C51/52 atau AT89S51/52), yaitu *timer/counter* dapat di fungsikan sebagai pembangkit gelombang (*wave generator*) seperti PWM (*Pulse Width Modulation*). Karena pulsa ini dibangkitkan oleh mikrokontroler, tentu saja harus ada pin khusus untuk *output* untuk mengeluarkan pulsa tersebut, yaitu:

1. OC0 / pin 4 pada port B (pwm dari *timer/counter* 0)
2. OC1A / pin 19 pada port D (pwm dari *timer/counter* 1)
3. OC1B / pin 18 pada port D (pwm dari *timer/counter* 1)
4. OC2 / pin 21 pada port D (pwm dari *timer/counter* 2)

Jadi ATmega 32 mempunyai 4 buah PWM (*Pulse Width Modulation*) yang dapat di aplikasikan untuk mengatur kecepatan motor, mengatur sudut putar motor servo, mengatur terang redup lampu, dan beberapa aplikasi lainnya, namun dalam hal ini penulis ingin mengaplikasikanya untuk pengaturan kecepatan motor DC shunt, Prinsip kerja dari PWM (*Pulse Width Modulation*) dapat di lihat pada gambar 2.11.



Gambar 2.11. Prinsip kerja PWM (*Pulse Width Modulation*)

Dari gambar 2.11 di atas menggunakan PWM0 pada portB.4 (OC0) ATmega32, register TCNT0 akan mencacah mulai dari 0, 1, 2, sampai 255 (*counting up*), kemudian akan mencacah turun dari 255, 254, 253 sampai 0 (*counting down*), demikian seterusnya sehingga tampak pada grafik garis tipis pada gambar 2.11. Nilai register OCR0 akan

dibandingkan dengan nilai TCNT0, sehingga akan muncul waktu dimana nilai OCR0 = TCNT0, kondisi seperti ini disebut *match*, pada saat inilah logika pin OC0 akan di ubah, jika sebelumnya berlogika *high* (1) maka akan di ubah menjadi *low* (0), atau sebaliknya dari *low* (0) menjadi *high* (1). Sehingga terjadilah gelombang kotak seperti pada gambar 2.11. yang bergaris tebal. Dengan mengatur nilai OCR0 maka lebar pulsa dapat kita atur.

Untuk mendapatkan referensi tegangan yang lebih besar, maka digunakan *driver* motor DC berupa IC L293D yang dapat mengalirkan tegangan 7,5 VDC dan arus 1 A, dalam hal ini mikrokontroler ATmega32 sebagai inputan pada IC L293D dan sekaligus mengatur *dyty cycle* pada pin Enable L293D sehingga tegangan keluaran pada IC L293D dapat di atur dari 0 V-7,5 V, sehingga motor DC dapat di di kendalikan kecepatanya dengan tegangan yang di atur pada terminal.

2.9. Pemrograman dengan BASCOM-AVR

Untuk membuat pwm (*Pulse Width Modulation*) dengan bahasa BASCOM - AVR kita dapat langsung mengakses register *timer / counter* dengan konfigurasi BASCOM-AVR. Misal kita akan membuat pwm pada PWM1A pada port D.5 (OC0A) maka konfigurasinya seperti berikut ini:

```
Config timer1=pwm, pwm=10, CompareA pwm=Clear Up, prescale 1
Pwm1A= xxxx
```

Config timer 1 maksudnya *timer 1* difungsikan sebagai pembangkit gelombang pwm, *pwm=10* maksudnya pwm 10 bit, *Compare A pwm= Clear Up* untuk membuat pwm dengan *type non inverting*, jika menghendaki pwm dengan *type inverting* maka *Compare A PWM=Clear Down*, *prescale* dapat menggunakan 1, 8, 64, 256, 1024. Dan untuk menghitung frekuensi pwm digunakan rumus :

$$f_{pwm} = \frac{clock I / 0}{2.N.TOP}$$

Keterangan :

Clock I/0 : Frekuensi osilator

N : *Prescale*

TOP : Nilai maksimal dari *timer* yang di gunakan sesuai dengan jumlah bitnya, untuk 8 bit nilai TOP = 255, 9 bit nilai TOP=511, dan 10 bit nilai TOP=1023

$$f_{pwm} = \frac{16\text{Mhz}}{2 \cdot (64) \cdot 1024} = 122,1\text{kHz}$$

f_{pwm} : Frekuensi PWM

$Clock\ I/O$: Frekuensi osilator sebesar 16Mhz

N : *Prescale* menggunakan 64

TOP : 1024 karena menggunakan 10 bit

Sedangkan $pwmIA=xxxx$ adalah untuk mengatur kecepatan motor DC, hal ini akan di bahas di bab IV.

2.9.1. Tutorial penggunaan BASCOM-AVR

1. Buka aplikasi BASCOM-AVR setelah sukses di *instal*
2. Kemudian pilih *File* → *New* sehingga akan tampil seperti gambar 2.12



Gambar 2.12. Tampilan BASCOM-AVR

3. Tulis program PWM pada lokasi “ tempat menulis *source code*” seperti pada gambar 2.13.

```

$regfile = "m32_def.dat"
$crystal = 16000000

Config TIMER1 = Pwm , Pwm = 10 , Prescale = 64 , Compare A Pwm = Clear Up
Config PORTE = Input
PORTB 0 = 1
PORTB 1 = 1
PORTB 2 = 1
PORTB 3 = 1
Config PORTD = Output
Do
Pwm = Pwm_eprom
If Sw_up = 0 Then : Incr Pwm : Waitms 10
End If
If Sw_down = 0 Then : Decr Pwm : Waitms 10
End If
If Pwm > 1023 Then : Pwm = 1023 : Waitms 10
End If
If Pwm < 0 Then : Pwm = 0 : Waitms 10
End If
Pwm_eprom = Pwm
Home
Lcd "LABORATORIUM SHAKER"
Lowerline
Pwma = Pwm
Pwalb = Pwm

```

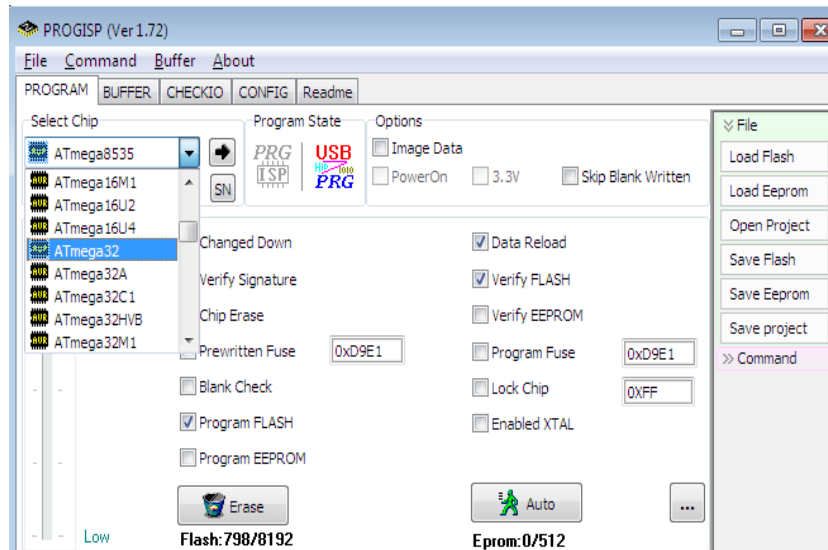
Gambar 2.13. Cara penulisan *source code* pada BASCOM-AVR

4. Kemudian simpan dengan nama *file* yang diinginkan, contoh “pwm”
5. Setelah disimpan kemudian *file* di *compile* untuk mendapatkan *file hexa* dengan menekan F7 seperti gambar 2.14



Gambar 2.14. Cara *compile* program ke *file hexa*

6. Setelah program di *compile* sempurna tanpa ada *error*, maka sudah di dapat *file hexa* yang siap di download ke ATmega 32, dengan cara membuka progISP seperti gambar 2.15



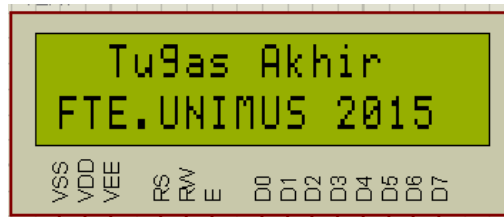
Gambar 2.15. Cara download ke ATMega 32 dengan progISP

7. Kemudian yang terakhir ambil *file hexa* dengan cara klik *load flash* kemudian pastikan ATMega32 sudah terhubung dengan PC/laptop melalui perantara *downloader*, setelah terhubung klik *Auto* seperti gambar 2.15 diatas, dengan hitungan kurang lebih 3 detik program sudah masuk ke ATMega 32 dan siap untuk beroperasi sebagai kontrol PWM.

2.10. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) merupakan suatu media penampilan data yang sangat efektif dan efisien dalam penggunaannya. Untuk menampilkan sebuah karakter pada layar LCD (*Liquid Crystal Display*) diperlukan beberapa rangkaian tambahan. Untuk lebih memudahkan para pengguna, maka beberapa perusahaan elektronik menciptakan modul LCD (*Liquid Crystal Display*). Adapun bentuk fisik LCD (*Liquid Crystal Display*) seperti pada Gambar 2.16.

LCD (*Liquid Crystal Display*) dibagi menjadi dua bagian yaitu bagian depan panel LCD (*Liquid Crystal Display*) yang terdiri dari banyak dot atau titik LCD (*Liquid Crystal Display*) dan mikrokontroler yang menempel pada bagian belakang panel LCD (*Liquid Crystal Display*) yang berfungsi untuk mengatur titik-titik LCD (*Liquid Crystal Display*) sehingga dapat menampilkan huruf, angka, dan simbol khusus yang dapat terbaca



Gambar 2.16. LCD 16 x 2

LCD (*Liquid Crystal Display*) berukuran 16 karakter x 2 baris dengan fasilitas *back lighting* memiliki 16 pin yang terdiri dari 8 jalur data, 3 jalur kontrol dan jalur-jalur catu daya, dengan fasilitas pin yang tersedia maka lcd 16 x 2 dapat digunakan secara maksimal untuk menampilkan data yang dikeluarkan oleh mikrokontroler, secara ringkas fungsi pin-pin pada LCD (*Liquid Crystal Display*) dituliskan pada Tabel 2.5.

Tabel 2.5. Fungsi pin LCD 16x2

PIN	SIMBOL	FUNGSI
1	Vss	Sambungan catu daya tegangan positif
2	Vdd	Sambungan catu daya ground / 0 V
3	Vo	mengatur kontras LCD
4	RS	kontrol input (logika 1) untuk transfer data
5	Rw	menulis dan membaca karakter/informasi
6	E	transfer aktual data dengan modul
7	DB0	jalur data/bus transfer data ke display
8	DB1	jalur data/bus transfer data ke display
9	DB2	jalur data/bus transfer data ke display
10	DB3	jalur data/bus transfer data ke display
11	DB4	jalur data/bus transfer data ke display
12	DB5	jalur data/bus transfer data ke display
13	DB6	jalur data/bus transfer data ke display
14	DB7	jalur data/bus transfer data ke display
15	A	tegangan positif untuk backlight
16	K	ground untuk backlight

2.11. *Push Button*

Push Button adalah saklar tekan yang berfungsi sebagai pemutus atau penyambung arus listrik, Prinsip kerja *Push Button* adalah apabila dalam keadaan normal tidak ditekan maka kontak tidak berubah, apabila ditekan maka kontak NC akan berfungsi sebagai *stop* (memberhentikan) dan kontak NO akan berfungsi sebagai *start* (menjalankan). Bentuk fisik *push button* seperti terlihat pada gambar 2.17



Gambar 2.17. *Push button*

Push button ada dua tipe diantaranya:

1. *Tipe Normally Open (NO)*

Tombol ini disebut juga dengan tombol start karena kontak akan menutup bila ditekan dan kembali terbuka bila dilepaskan. Bila tombol ditekan maka kontak bergerak akan menyentuh kontak tetap sehingga arus listrik akan mengalir.

2. *Tipe Normally Close (NC)*

Tombol ini disebut juga dengan tombol stop karena kontak akan membuka bila ditekan dan kembali tertutup bila dilepaskan. Kontak bergerak akan lepas dari kontak tetap sehingga arus listrik akan terputus

Pada penelitian ini saklar *push button* yang penulis pakai adalah tipe *normally open*, hal ini bertujuan untuk memberikan interuksi kepada mikrokontroler ketika saklar ditekan, sehingga mikrokontroler mendapat inputan berupa perintah saklar *push button* yang mempresentasikan kecepatan PWM (*Pulse Width Modulation*) dan *timer*.