

# APLIKASI MOTOR DC-SHUNT UNTUK LABORATORY SHAKER MENGGUNAKAN METODE PWM (PULSE WIDTH MODULATION) BERBASIS MIKROKONTROLER ATMEGA 32

Royan<sup>1</sup>\*, Luqman A<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Elektro, Fakultas Teknik Universitas Muhammadiyah Semarang  
Jl. Kasipah No.12 Semarang

\*Email : [royan.belix@gmail.com](mailto:royan.belix@gmail.com)

## ABSTRACT

*Motor DC (Direct Current) banyak digunakan sebagai penggerak dalam berbagai peralatan, baik kecil maupun besar, lambat maupun cepat. Metode PWM (Pulse Width Modulation) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC (Direct Current) namun diperlukan rangkaian driver motor, diantaranya yang paling sederhana berupa transistor yang disusun secara Darlington maupun driver motor DC (Direct Current) dalam kemasan IC seperti L293D yang dapat mengalirkan arus dan tegangan yang lebih besar serta mudah dikendalikan dengan mikrokontroler untuk arah putar dan kecepatannya.*

*Tujuan Penelitian ini adalah merancang sebuah perangkat pengendalian motor DC shunt dengan menggunakan metode PWM ATmega 32 yang dapat bekerja otomatis dengan timer dan menggunakan bahasa pemrograman yang mudah dikembangkan yaitu bascom AVR serta mengaplikasikannya pada laboratory shaker. Sehingga penelitian ini penulis beri judul "Aplikasi Motor DC Shunt Untuk Laboratory Shaker dengan menggunakan metode PWM berbasis mikrokontroler ATmega32". Alat ini mampu mengendalikan kecepatan motor DC Shunt sesuai kecepatan yang diatur, serta bekerja secara otomatis menggunakan timer.*

**Kata Kunci :** Motor DC Shunt, L293D, PWM, Mikrokontroler ATmega 32, Laboratory shaker, Bascom AVR

## 1. PENDAHULUAN

Dua masalah umum dalam mengendalikan motor DC (*direct curent*) adalah mengendalikan arah putar dan mengendalikan kecepatan putar, untuk mengendalikan arah putar dapat menggunakan saklar untuk mengubah arah arus. Sedangkan untuk mengendalikan kecepatan ada beberapa metode yang bisa di gunakan diantaranya, pengaturan fluks medan ( ), pengaturan tahanan jangkar

(Ra), dan pengaturan tegangan masukan (Eko, 2008).

Metode PWM (*Pulse Width Modulation*) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC (*direct curent*). PWM (*Pulse Width Modulation*) ini bekerja dengan cara membuat perbandingan pulsa *high* terhadap pulsa *low* . Perbandingan pulsa *high* terhadap *low* ini akan menentukan besarnya tegangan yang diberikan ke

motor DC (*direct curent*), sehingga kecepatan dapat di atur (Nasrul, 2009).

Sinyal PWM (*Pulse Width Modulation*) 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%) (Andri, 2014).

Mikrokontroler adalah salah satu komponen yang mampu membangkitkan PWM (*Pulse Width Modulation*) sehingga dapat dijadikan kontrol motor DC (*direct curent*) dengan mengatur lebar pulsa maka tegangan pada terminal motor DC (*direct curent*) dapat diatur (Prasetyo, 2010).

2. Sistem kendali berbasis mikrokontroler sangat fleksibel, karena mempunyai kemampuan untuk mengubah metode atau sifat pengendali suatu proses tanpa mengubah perangkat kerasnya. Perubahan algoritma tersebut cukup dengan mengubah perangkat lunaknya, hal ini berarti sangat menghemat biaya dan juga tingkat efisiensinya cukup tinggi dibanding menggunakan sistem kendali analog (Muchlas, 2006).

## 2. LANDASAN TEORI

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 current*) 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 current*) 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)

### **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, atau naik turun, di atas plat inilah beberapa larutan di letakan sehingga larutan dapat dicampur.. Gambar 2.1 merupakan contoh *Laboratory shaker*.



Gambar 2.1. *Laboratory shaker*

### **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:

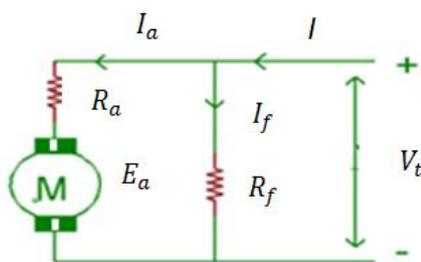
1. Kutub medan.

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

**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:

$$E_a = V_t - I_a R_a \tag{2.1}$$

$$V_t = I_f R_f \tag{2.2}$$

$$I = I_a + I_f \tag{2.3}$$

$$n = \frac{E_a}{CW} \tag{2.4}$$

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

$w$  : Fluks magnet/kutub

$n$  : Kecepatan putar (rpm)

$C$  : Konstanta motor

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

**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

### 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

### 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

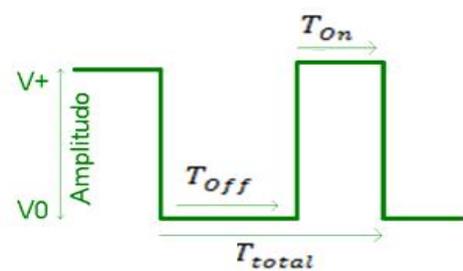
### Pengendalian Tegangan Jangkar

Mengubah tegangan yang diberikan pada rangkaian jangkar merupakan model pengendalian Tegangan Jangkar, salah

satu pengendalian ini adalah dengan metode pwm (*Pulse Width Modulation*)

### 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

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

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

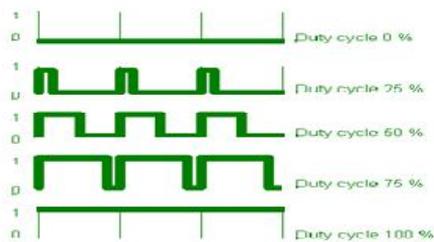
$T_{Off}$  : Waktu pulsa Off / Low

$T_{On}$  : 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

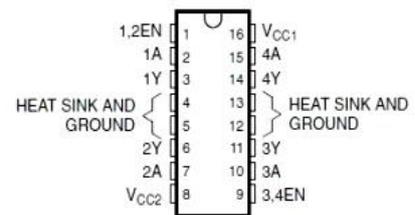


Gambar 2.7. *Duty cycle* pwm 8 bit

**Driver Motor L293D**

IC L293D adalah IC yang didesain khusus sebagai *driver* motor DC (*Direct Current*) dan dapat dikendalikan dengan rangkaian TTL maupun mikrokontroler. Aplikasi Motor DC *Shunt*.....

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

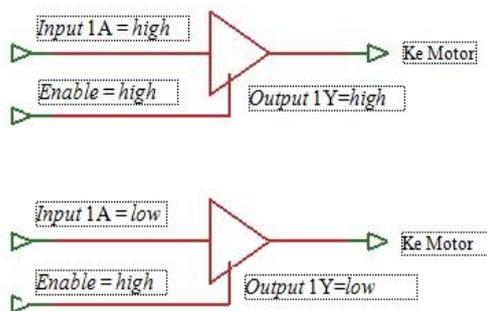
Keterangan masing-masing pin L293D

1. Pin EN (*Enable*, EN1.2, EN3.4) berfungsi untuk mengijinkan *driver* menerima perintah untuk menggerakan 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.

### 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

### Mikrokontroler ATmega 32

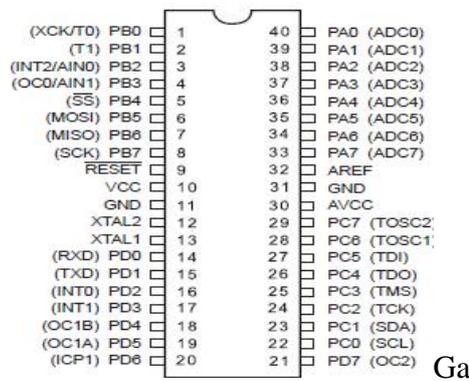
Seperti mikroprosesor pada umumnya, secara *internal* mikrokontroler ATmega32 terdiri atas unit-unit fungsionalnya *Arithmetic and Logical Unit* (ALU), himpunan *register* kerja, *register* dan *dekoder instruksi*, dan pewaktu beserta komponen kendali lainnya. Berbeda dengan mikroprosesor, mikrokontroler 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*

**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

**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

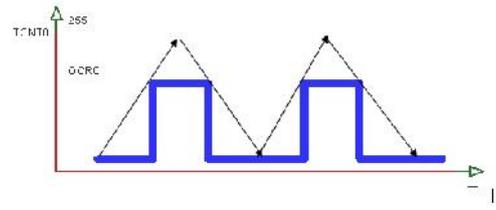
- atau sumber osilator luar dapat digunakan.
6. Pin 13 (XTAL 1) adalah pin keluaran ke rangkaian osilator *internal*. Pin ini dipakai bila menggunakan osilator kristal.
  7. Pin 14 sampai 21 (Port D) adalah 8-bit dua arah (*bi-directional I/O*) port dengan *internal pull-up resistors*) digunakan untuk *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*

### 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 kecepatannya dengan tegangan yang di atur pada terminal.

### **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 konfigurasi 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. Sedangkan *pwm1A=xxxx* adalah untuk mengatur kecepatan.

### **3. METODE PENELITIAN**

#### **Bahan-Bahan yang Digunakan**

Bahan-Bahan Untuk Perangkat Keras

Bahan dan komponen elektronika yang digunakan dalam tugas akhir ini adalah:

1. Motor DC shunt dengan spesifikasi sebagai berikut:
  - 1) *Voltage* : 6 VDC (7, 5 max)
  - 2) *Shaft / output rpm* :1800 rpm
  - 3) *Torque* : 0.4 Kg.cm
  - 4) *No load current* :450 mA
  - 5) *Stall current* : 2,2 A
2. Mikrokontroler ATmega 32
3. LCD 16x2
4. *Driver* motor L293D
5. Saklar *Phus On*
6. *Crystal* 16 Mhz
7. Kapasitor polar
8. Resistor
9. Led
10. PCB polos dan pelarutnya
11. *Power supply*

#### **Bahan-Bahan Untuk Perangkat Lunak**

Perangkat lunak yang diperlukan dalam pemrograman ini adalah sebagai berikut :

1. *Bascom* AVR
2. Windows 7 atau Windows XP

#### **Alat-Alat yang Digunakan**

Peralatan yang diperlukan dalam merancang Perangkat Keras dalam Tugas Akhir ini antara lain adalah :

1. Multimeter
2. Solder dan timah

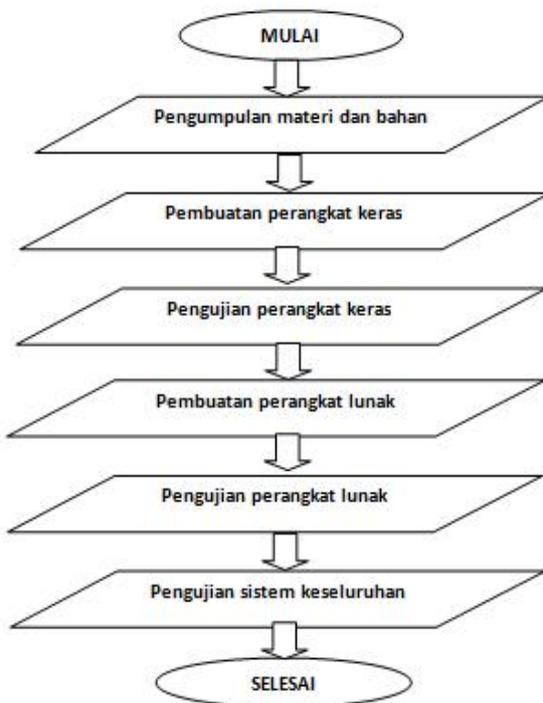
3. Obeng
4. Tang
5. Bor PCB
6. Alat Potong berupa Gergaji atau *Cutter*
7. *Downloder* ISP untuk memasukan program ke mikrokontroler AVR

Alat yang digunakan untuk Pembuatan Perangkat Lunak

1. Komputer yang mendukung Pemrograman *Bascom* AVR
2. Proteus sebagai simulasi awal program .

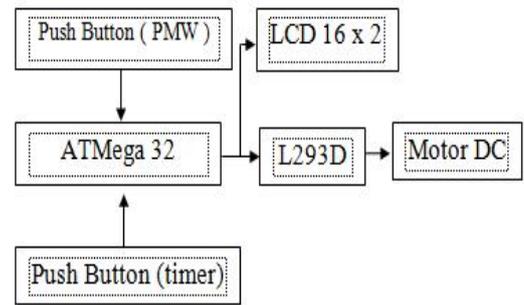
**Tahapan Penelitian**

Tahapan-tahapan yang dilakukan dalam Penelitian ini seperti *Flow Chart* pada gambar 3.1 di bawah ini :



Gambar 3.1. *Flow Chart*

**Blok diagram perangkat keras**

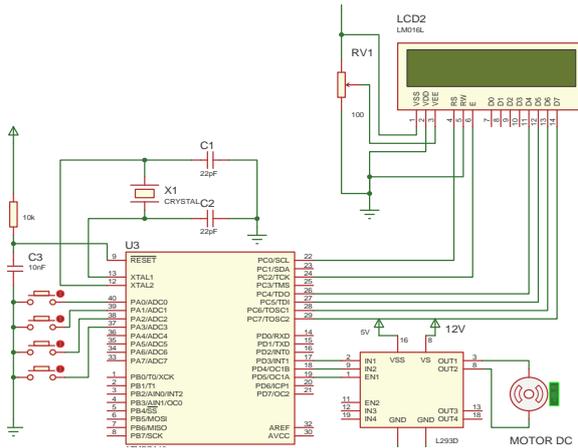


Gambar 3.2. Blok diagram perangkat keras

Push button sebagai interaksi/perintah awal kemikrokontroler ATmega 32 berupa perintah untuk mengatur *timer* yang akan di gunakan untuk proses *shaker* berlangsung, selain itu *push button* juga di jadikan perintah untuk menaikkan atau menurunkan kecepatan putaran motor DC (*direct current*), kemudian perintah ini di tampilkan di LCD 16x 2 berupa *timer Count down* seiring dengan ATmega 32 memberikan perintah otomatis ke L293D sebagai *driver* motor DC (*direct current*). Dan lamanya proses tergantung dari lamanya waktu yang di *setting*

**Rancangan Gambar Keseluruhan**

Rancangan gambar sistem keseluruhan dapat di lihat pada gambar 3.2.



Gambar 3.2. Rangkaian keseluruhan

Cara Kerja Rangkaian:

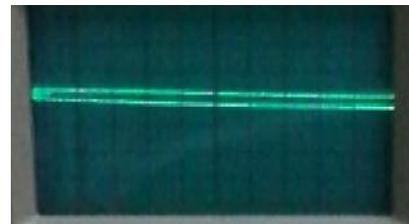
1. Saklar (*push button* 1) akan memberikan interuksi kepada ATmega 32 melalui PortA.0, sebagai perintah untuk *setting timer (time up)*
2. Saklar (*push button* 2) akan memberikan interuksi kepada ATmega 32 melalui PortA.1, sebagai perintah untuk *setting timer (time down)*
3. Saklar (*push button* 3) akan memberikan interuksi kepada ATmega 32 melalui PortA.2, sebagai perintah untuk *setting kecepatan (speed up)*
4. Saklar (*push button* 4) akan memberikan interuksi kepada ATmega 32 melalui PortA.3, sebagai perintah untuk *setting kecepatan (speed down)*
5. Lamanya proses *shaker* dan *duty cycle* pwm di tampilkan ke LCD

6. Gelombang pwm akan di bangkitkan di ATmega 32 dan di keluarkan melalui portd.5 (OC1A) yang selanjutnya akan menjadi inputan IC L293D untuk mengatur tegangan yang di berikan ke motor DC.

#### 4. PENGUJIAN DAN HASIL

Pada penelitian ini diperoleh beberapa data, diantaranya data pengujian dengan osiloskop, pengujian dengan AVO meter, serta pengujian dengan tachometer. berikut adalah hasil dan data pengujianya:

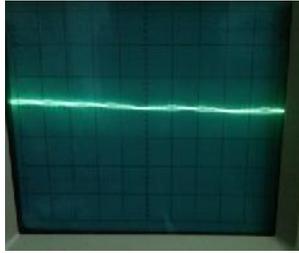
**Pengujian Dengan Osiloskop Tegangan masukan pada *driver* motor ( $V_{in}$ ) 7,5 V.**



Gambar 4.1. Bentuk pulsa tegangan masukan pada *driver* motor ( $V_{in}$ )

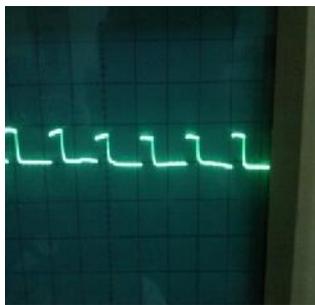
**Pengujian *output* pada *driver* motor ( $V_{out}$ ) dengan variasi *duty cycle* menggunakan osiloskop.**

1. Pengujian dengan *duty cycle* 0 % dapat dilihat pada gambar 4.2



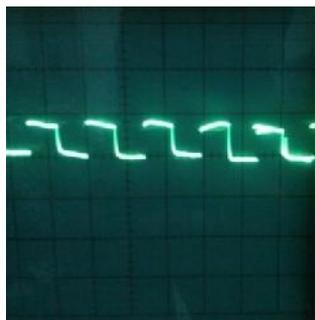
Gambar 4.2. *Output driver motor* dengan PWM *duty cycle* 0%

2. Pengujian dengan *duty cycle* 25 % dapat dilihat pada gambar 4.3



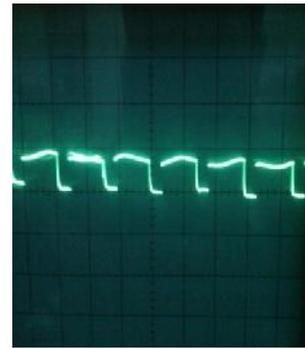
Gambar 4.3. *Output driver motor* dengan PWM *duty cycle* 25%

3. Pengujian dengan *duty cycle* 50 % dapat dilihat pada gambar 4.4



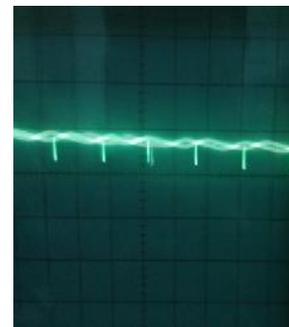
Gambar 4.4. *Output driver motor* dengan PWM *duty cycle* 50%

4. Pengujian dengan *duty cycle* 75 % dapat dilihat pada gambar 4.5



Gambar 4.5. *Output driver motor* dengan PWM *duty cycle* 75%

5. Pengujian dengan *duty cycle* 100% dapat dilihat pada gambar 4.6



Gambar 4.6. *Output driver motor* dengan PWM *duty cycle* 100%

**Pengujian kecepatan putar motor dengan tachometer.**

1. Pengujian dengan *duty cycle* 25% dapat dilihat pada gambar 4.7



Gambar 4.7. RPM motor dengan PWM *duty cycle* 25%

- Pengujian dengan *duty cycle* 50% dapat dilihat pada gambar 4.8



Gambar 4.8. RPM motor dengan PWM *duty cycle* 50%

- Pengujian dengan *duty cycle* 75% dapat dilihat pada gambar 4.9



Gambar 4.9. RPM motor dengan PWM *duty cycle* 75%

- Pengujian dengan *duty cycle* 100% dapat dilihat pada gambar 4.10



Gambar 4.10. RPM motor dengan PWM *duty cycle* 100%

### Analisa Hardware

Analisa Tegangan Keluaran pada *driver* motor ( $V_{out}$ ) Terhadap *Duty cycle* yang dibangkitkan oleh mikrokontroler ATmega 32.

Pada pengujian tegangan keluaran *driver* motor  $V_{out}$  terhadap perubahan *duty cycle*, tegangan masukan pada *driver* motor ( $V_{in}$ ) yang penulis gunakan adalah tegangan sebesar 7,5 volt, hal ini karena motor DC yang penulis gunakan pada penelitian ini adalah motor DC dengan tegangan operasional 6 volt dan maksimal tegangan 7,5 volt. Hasil pengujian dapat dilihat pada tabel.

Tabel 4.1 Hasil pengujian  $V_{out}$  terhadap *duty cycle*

<i>Duty cycle</i> (%)	Lebar pulsa dengan Time/div 5ms		Tegangan keluaran <i>driver</i> motor ( $V_{out}$ )
	High	Low	
0	0	8	0
25	2,5	5,5	2,23
50	4	4	3,80
75	5,5	2,5	5,26
100	8	0	7,47

Jika hasil pengujian pada tabel 4.1 diatas kita jabarkan dengan perhitungan rumus:

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{in} \tag{4.1}$$

Keterangan :

$V_{out}$  : Tegangan keluaran

$T_{on}$  : Waktu pulsa ON / High

$V_{in}$  : Tegangan masukan

$$T_{total} = T_{off} + T_{on} \tag{4.2}$$

Keterangan :

$T_{off}$  : Waktu pulsa OFF / Low

$T_{on}$  : Waktu pulsa ON / High

Dengan melihat persamaan rumus 4.1 dan 4.2, Maka diperoleh data perhitungan sebagai berikut:

1. Tegangan keluaran  $V_{out}$  dengan *duty cycle* 0 %

$$V_{out} = \frac{0}{0 + 7} \times 7,5 = 0$$

2. Tegangan keluaran  $V_{out}$  dengan *duty cycle* 25 %

$$V_{out} = \frac{2,5}{2,5 + 5,2} \times 7,5 = 2,34$$

3. Tegangan keluaran  $V_{out}$  dengan *duty cycle* 50 %

$$V_{out} = \frac{4}{4 + 4} \times 7,5 = 3,75$$

4. Tegangan keluaran  $V_{out}$  dengan *duty cycle* 75%

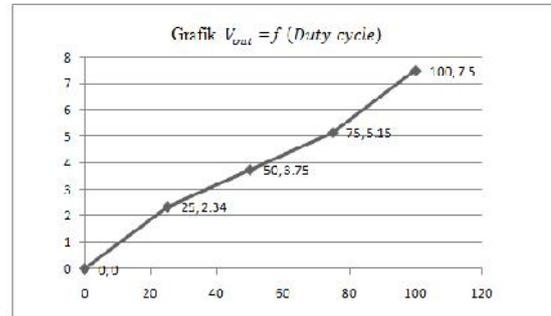
$$V_{out} = \frac{5,5}{5,5 + 2,5} \times 7,5 = 5,15$$

5. Tegangan keluaran  $V_{out}$  dengan *duty cycle* 100 %

$$V_{out} = \frac{8}{8 + 0} \times 7,5 = 7,5$$

Dari data-data diatas dapat dibuat grafik hubungan antara tegangan keluaran *driver* motor ( $V_{out}$ ) terhadap perubahan

*duty cycle* 0 % sampai dengan 100 % seperti gambar 4.12.



Gambar 4.12. Hubungan antara ( $V_{out}$ )

terhadap perubahan *duty cycle*

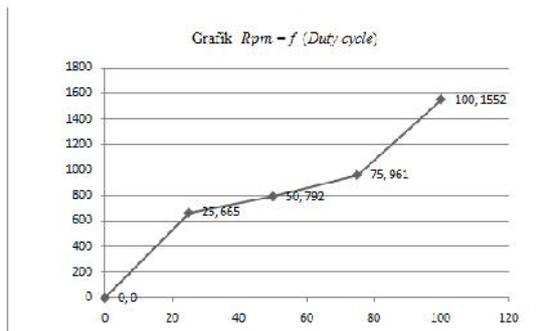
### Analisa Kecepatan Motor (RPM) Terhadap *Duty Cycle*

Pada pengujian kecepatan putar (RPM) terhadap *duty cycle*, tegangan masukan ( $V_{in}$ ) yang penulis gunakan adalah tegangan sebesar 7,5 volt, hal ini karena motor DC yang penulis gunakan pada penelitian ini adalah motor DC dengan tegangan operasional 6 volt dan maksimal tegangan 7,5 volt. Hasil pengujian dapat dilihat pada tabel 4.2

Tabel 4.2. Pengujian RPM motor dengan *duty cycle* yang di atur

<i>Duty cycle</i> (%)	Lebar pulsa Time/div 5ms		Kecepatan putar (rpm)
	High	Low	
0	0	8	0
25	2,5	5,5	665,0
50	4	4	792,6

Dari data-data diatas dapat dibuat grafik hubungan antara kecepatan putar motor dengan *duty cycle* antara 0 % sampai dengan 100 % seperti pada gambar 4.14



Gambar 4.14. Hubungan RPM motor terhadap *duty cycle*

Dari data-data penelitian diatas dapat diambil kesimpulan bahwa dengan merubah tegangan masukan pada *driver* motor ( $V_{in}$ ) dapat merubah tegangan keluaran ( $V_{out}$ ) sehingga secara otomatis kecepatan juga dapat dirubah, demikian halnya dengan merubah *duty cycle* PWM, maka tegangan keluaran ( $V_{out}$ ) dan kecepatan motor dapat dikendalikan.

#### Analisa software

Pada perancangan sistem pengendalian motor DC Shunt ini PWM yang di gunakan adalah PWM1 pada PORTD.5(OC1) ATmega32 yang mempunyai kapasitas 10bit sehingga memiliki variasi perubahan nilai sebanyak 1023, variasi mulai dari 0 – 1023 perubahan nilai yang mewakili *duty cycle* 0 – 100%, berikut adalah cuplikan *software* dari ATmega32 untuk pengendalian PWM1 dan penjelasanya

```
$regfile = "m32def.dat"
```

```
$crystal = 16000000
```

Aplikasi Motor DC Shunt.....

Jenis mikrokontroler ATmega 32 dan *crystal* yang di gunakan 16Mhz

**Config Int0 = Low Level**

**Config Int1 = Low Level**

**On Int0 Time\_up**

**On Int1 Time\_down**

**Enable Int0**

**Enable Int1**

**Enable Interrupts**

**Portd.2 = 1**

**Portd.3 = 1**

Konfigurasi untuk interuksi penambahan atau pengurangan *timer* PWM

**Config Lcdpin = Pin , Rs = Portc.0 , E = Portc.2 , Db4 = Portc.4 ,**

**Config Lcdpin = Pin , Db5 = Portc.5 ,**

**Db6 = Portc.6 , Db7 = Portc.7**

**Config Lcd = 16 \* 2**

**Cursor Off**

**Cls**

Konfigurasi dan pengalamatan LCD ukuran 16x2 pada portc ATmega32

**Config Timer1 = Pwm , Pwm = 10 ,**

**Compare A Pwm = Clear Up**

*Timer* 1 digunakan untuk membangkitkan pwm 10bit, hitungan naik *non inverting*

**Dim Waktu As Integer , Dely As Byte**

**Config Porta = Input**

**Config Portd = Output**

**Sw\_up Alias Pinb.2**

**Sw\_down Alias Pinb.3**

Konfigurasi *Input* dan *Output*

*Dim Pwm As Integer , Set\_kecepatan As Integer , Persen As Integer*

*Dim Pwm\_eprom As Eram Integer , Persenya As Integer , Hasil As Integer*

Ukuran-ukuran data pada folder yang disiapkan

Program utama :

**Do**

**If Waktu <= 0 Then**

**Gosub Selesai**

Jika waktu/timer =0 maka lompat ke label “selesai”

**End If**

**If Sw\_up = 0 Then**

**Incr PWM**

Jika tombol Up di tekan PWM bertambah/kecepatan motor dinaikan

**End If**

**If Sw\_down = 0 Then**

**Decr PWM**

Jika tombol down ditekan maka PWM dikurangi/kecepatan motor diturunkan

**End If**

**Pwm1a = Pwm**

Output PWM pada PWM1A yaitu pada portd.5 ATmega32 (lihat data sheet)

**Locate 1 , 1**

**Lcd "Duty Sycle:" ; Hasil ; "% "**

LCD lokasi baris 1 dan mulai dari kolom 1 untuk menampilkan Duty cycle dalam %

**Locate 2 , 1**

**Lcd "Duration : " ; Waktu ; ".s "**

**Waitms 10**

Locasi LCD baris 2 dan mulai dari kolom 1 untuk menampilkan *timer*

**Loop**

**End**

**Time\_up:**

**Waktu = Waktu + 30**

**Return**

pengaturan waktu naik per30 detik ketika tombol *Up* ditekan, dan setelah tombol di lepas maka kembali ke program utama di atas.

**Time\_down:**

**Waktu = Waktu - 30**

pengaturan waktu turun per30 detik ketika tombol *down* ditekan, dan setelah tombol di lepas maka kembali ke program utama di atas.

**Return**

**Selesai:**

**Portd.6 = 0**

**Locate 1 , 1**

**Lcd "-----Finish-----"**

**Portd.7 = 0 ‘ buzzer bunyi**

Proses selesai jika *timer* telah selesai dan di sertai bunyi buzzer

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Setelah penulis menyelesaikan perancangan dan pembuatan peralatan ini, maka dapat disimpulkan:

1. *Microcontroller* ATmega 32 dapat membangkitkan gelombang PWM yang dapat diaplikasikan pada sistem pengendalian motor DC shunt yang cukup efektif
2. Sistem pengendalian ini dapat bekerja secara otomatis dengan *timer* yang di setting.
3. Sistem pengendalian ini dapat diprogram dengan mudah menggunakan bahasa yang sangat sederhana yaitu bascom-AVR

### 5.2. Saran

Setelah penulis selesai membuat alat dan tulisan ini, penulis menyadari bahwa masih banyak kekurangan yang penulis rasakan, sehingga penulis ingin memberikan saran-saran yang dapat dijadikan motivasi penulis dan pembaca tulisan ini, antara lain:

1. Sistem pengendalian motor DC shunt ini masih butuh penyempurnaan sistem mekanik jika ingin diaplikasikan pada *laboratory shaker* yang memiliki unjuk kerja yang lebih halus.
2. Alat ini dapat digunakan pada laboratorium kesehatan namun perlu

Aplikasi Motor DC *Shunt*.....

diperhatikan lebih teliti konstruksi dan *layout* untuk penempatan *tube samples*.

## DAFTAR PUSTAKA

- Azwardi .(2012); *Perancangan Kontrol dan Monitoring Kecepatan Motor DC Melalui Jaringan Intranet*, Teknik Komputer, Politeknik Negeri Sriwijaya
- Data Sheet ATmega32  
<http://www.atmel.com/images/doc2503.pdf>
- Data sheet L293D  
[http://www.datasheetcatalog.com/datasheets\\_pdf/L/2/9/3/L293D.shtml](http://www.datasheetcatalog.com/datasheets_pdf/L/2/9/3/L293D.shtml)
- Hamdani, M.(2010); *Pengendalian Kecepatan Putar Motor DC Terhadap Perubahan Temperatur Dengan Sistem Modulasi Lebar Pulsa*, teknik elektro, Universitas Indonesia
- Ihsanto, E., Triwisaksana. Dan Suryana, N. (2008); *Sistem Pengendalian Motor DC Dengan PWM*, Teknik Elektro, universitas Mercu Buana
- Laboratory shaker*  
<http://www.medicalexpo.com/prod/stuart-equipment/laboratory-shaker-3d>

Muchlas. Sutikno,T. dan Hastoro,C.  
(2006); *Pengendali Kecepatan Motor DC Dengan PWM Secara Remote Berbasis Mikrokontroler M68hc11*, Teknik Elektro, Universitas Ahmad Dahlan

Nasrul.(2009); *Pengendalian Kecepatan Motor DC dengan Metoda Fuzzy Logic Berbasis Mikrokontroler*, Teknik Elektro,Politeknik Negeri Padang