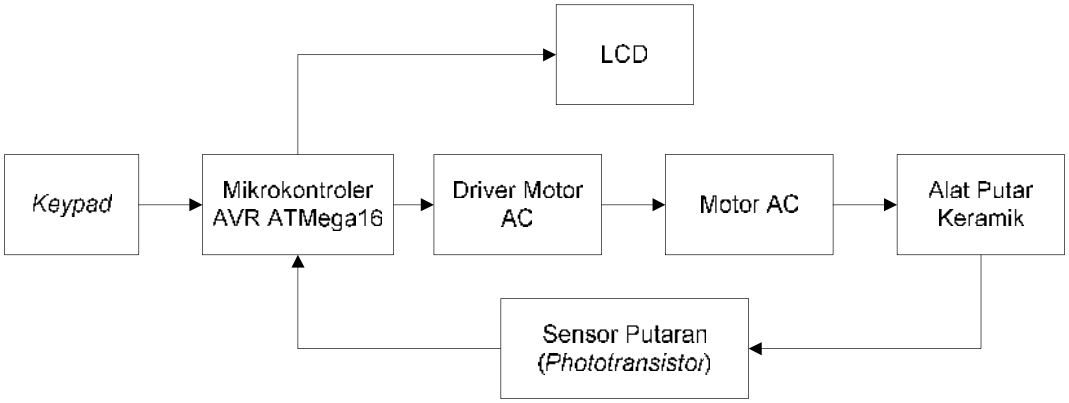
BAB III

PERANCANGAN ALAT DAN PROGRAM MIKROKONTROLER

Pada tahap perancangan ini dibagi menjadi 2 tahap perancangan. Tahap pertama adalah perancangan perangkat keras (*hardware*), yang meliputi rangkaian – rangkaian elektronika dan alat putar keramik. Tahap kedua adalah perancangan algoritma, *listing* program pada *software Code Vision* AVR dan penanaman *listing* program pada mikrokontroler AVR ATMega16 dengan menggunakan *software* ISP *Programmer*.

3.1 Perancangan Perangkat Keras

Seluruh perangkat atau komponen yang digunakan dalam perancangan pengaturan kecepatan pada alat putar keramik menggunakan motor AC ini, tersusun seperti pada blok diagram di bawah ini.



Gambar 3.1 Blok Diagram Pengaturan Kecepatan Motor AC Pada Alat Putar Keramik

Blok diagram diatas, secara garis besar dapat dijelaskan sebagai berikut.

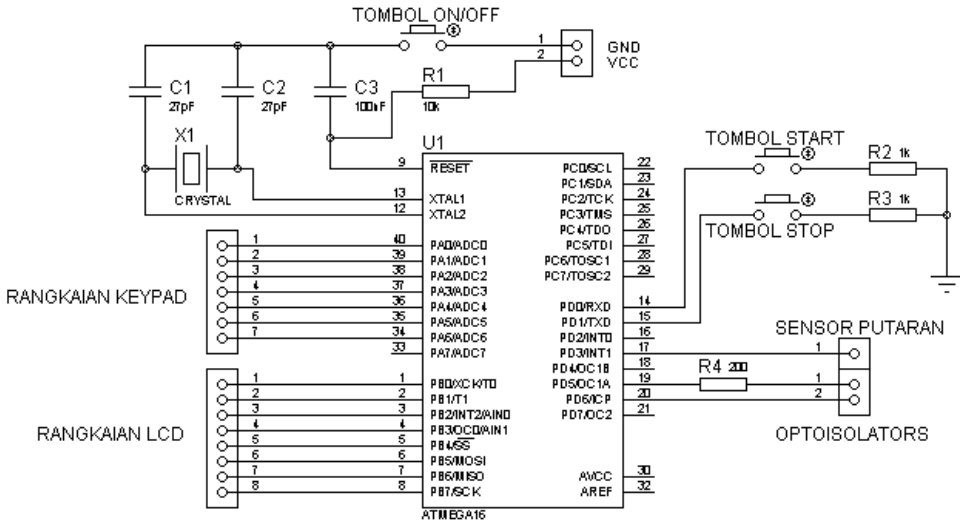
1. *Keypad* digunakan sebagai alat untuk memasukkan nilai kecepatan putaran yang diinginkan (nilai *setpoint*). *Keypad* dihubungkan ke PORTB mikrokontroler AVR ATMega16. Nilai dari *keypad* tersebut mewakili nilai kecepatan putaran dengan satuan *rotation per minute* (rpm). Nilai yang diijinkan sebesar antara 60–150rpm.
2. Nilai *setpoint* akan diolah sedemikian rupa dengan menggunakan suatu program yang ditanamkan pada mikrokontroler AVR ATMega16, sehingga nilai tersebut dapat ditampilkan pada LCD dan digunakan pada register OCR1A yang berfungsi untuk membangkitkan sinyal PWM.
3. Sinyal PWM dibangkitkan setelah nilai pada register OCR1A terisi (OCR1A ≠ 0) dan sinyal PWM dikeluarkan melalui *Pin* 19 PORTD.5 (OC1A) pada mikrokontroler AVR ATMega16.
4. Sinyal PWM tersebut akan memicu TRIAC *Optoisilator* pada rangkaian *driver* motor AC. Rangkaian *zero crossing detector* telah tersedia dalam satu paket (*On-package*) IC TRIAC *Optoisolators* MOC3041, sehingga titik acuan sinyal PWM pada saat mengatur sinyal sinusoidal (tegangan AC) dimulai pada saat perpotongan titik nol *(zero crossing*). Keluaran dari TRIAC *Optoisolators* tersebut (*pin* 6) akan memicu *gate* TRIAC (Q4004LT) sehingga memberikan arus pada motor AC.
5. Motor AC akan menggerakkan alat putar keramik dan alat putar tersebut akan dideteksi putarannya setiap 1 detik (*Timer* 0) oleh sensor putaran (*phototransistor*). Pendeteksian putaran alat putar ini bertujuan untuk mengetahui apakah kecepatan putaran alat putar sama dengan nilai *setpoint* yang diberikan. Kecepatan alat putar akan dipengaruhi oleh beban (tanah liat) yang bervariasi. Kecepatan akan berkurang apabila beban semakin berat, oleh karena itu sensor pun berfungsi untuk

mengetahui kecepatan alat putar terhadap beban. Sensor putaran ini menggunakan fitur interupsi eksternal 1 pada mikrokontroler ATMEGA16, sehingga mikrokontroler akan mengetahui setiap kali sensor memberikan respon (logika 0).

6. Program pengaturan alat putar pada mikrokontroler ATMega16 akan mengatur dan membandingkan antara kecepatan *setpoint* dan kecepatan yang sesungguhnya (*real time*). Jika sensor mendeteksi kecepatan alat putar lebih cepat dari pada kecepatan *setpoint*-nya, maka program akan mengatur (mengurangi) nilai pada *register* OCR1A dan jika sensor mendeteksi kecepatan alat putar lebih lambat dari pada kecepatan *setpoint*-nya, maka program akan mengatur (menambahi) nilai pada *register* OCR1A.

3.1.1 Perancangan Rangkaian Sistem Minimum AVR ATMega16

Mikrokontroler AVR ATMega16 dapat dioperasikan dengan cara menambahkan beberapa komponen elektronika yang berfungsi sebagai komponen pendukungnya. Mikrokontroler dan komponen – komponen pendukung tersebut tergabung dalam satu rangkaian yang disebut sebagai rangkaian sistem minimum.



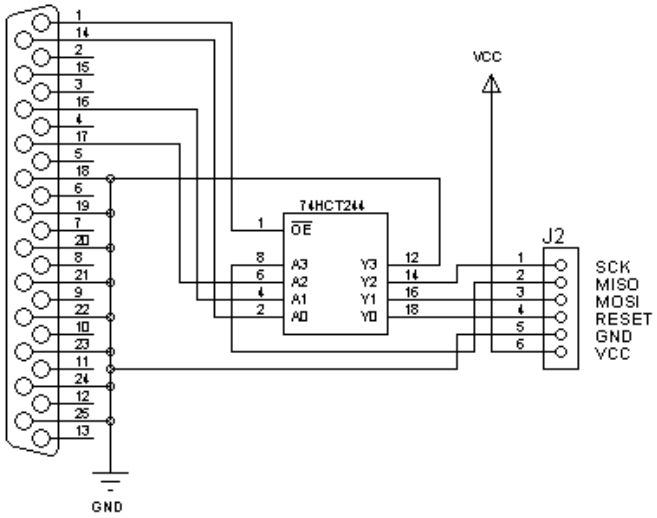
Gambar 3.2 Rangkaian Sistem Minimum AVR ATMega16

Kristal yang digunakan pada rangkaian sistem minimum di atas, mengunakan frekuensi 4 MHz. Kristal tersebut digunakan untuk pembangkit *clock* (osilator), dimana setiap 1 intruksi/perintah dalam program dieksekusi dalam 1 siklus *clock*. *Pin* RESET dihubungkan dengan rangkaian kombinasi RC dan *push button*, yang bertujuan agar mikrokontroler dapat di-*reset.* Fungsi dari *port - port* lainnya adalah sebagai berikut.

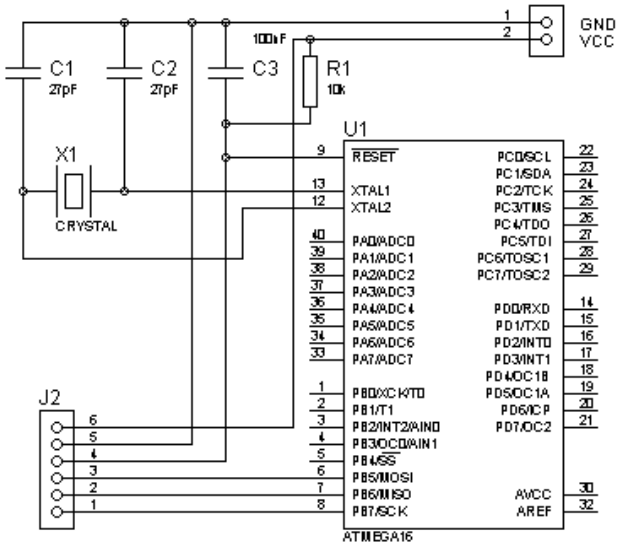
1. PORTA, digunakan sebagai *pin* masukkan untuk *Keypad* 3X4
2. PORTB, digunakan sebagai *pin* keluaran untuk LCD
3. PORTD.0 dan PORTD.1, digunakan sebagai *pin* masukan untuk tombol START dan tombol STOP
4. PORTD.3, digunakan sebagai *pin* masukkan untuk sensor putaran (*Ext. Interrupt*)
5. PORTD.5 dan PORTD.6, digunakan sebagai *pin* keluaran untuk sinyal PWM.

3.1.2 Perancangan Rangkaian *Downloader*

Rangkaian *downloader* merupakan rangkaian penghubung antara komputer dan mikrokontroler yang berfungsi untuk memasukan *listing* program (berupa *bit – bit* logika) ke dalam mikrokontroler. *Listing* program yang dikirim oleh *software* dari komputer ke dalam mikrokontroler biasanya berbentuk *file* *.hex (heksadesimal). Pada umumnya rangkaian *downloader* terdiri dari kabel penghubung jenis DB25 atau jenis DB9. Sinkronisasi tegangan antara tegangan dari komputer dan tegangan mikrokontroler menggunakan sebuah *buffer*. Rangkaian *downloader* ditunjukkan seperti Gambar 3.3 dan Gambar 3.4.

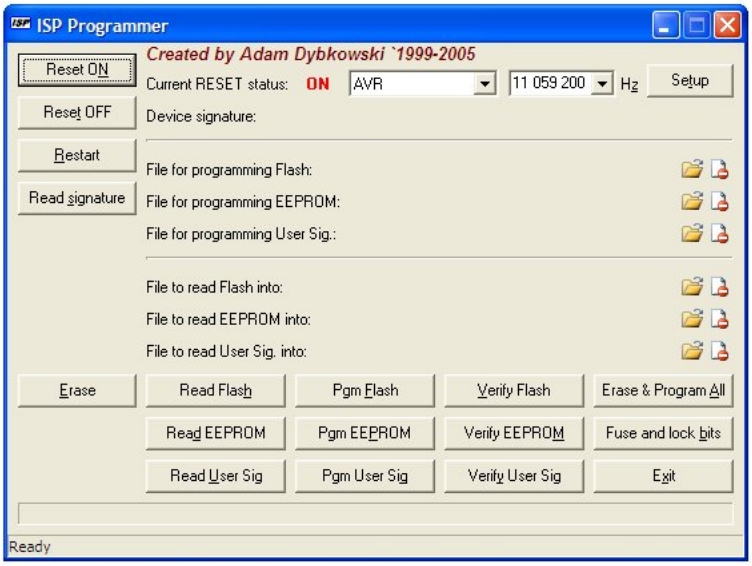


Gambar 3.3 Rangkaian *Downloader* (DB25)



Gambar 3.4 Rangkaian *Downloader* (Mikrokontroler)

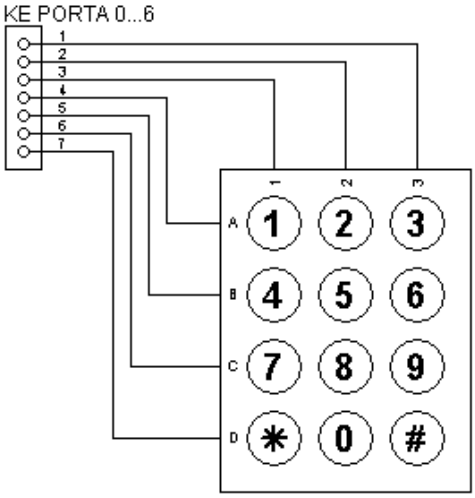
Rangkaian di atas menggunakan sebuah *port* DB25 sebagai alat penghubung antara komputer dan rangkaian *downloader*, sedangkan IC 74HCT244 digunakan sebagai *buffer*. *Software* yang digunakan untuk men-*download* program (*file*: *.hex) ke dalam mikrokontroler ini adalah ISP *Programmer* (Adam Dybkowsky).



Gambar 3.5 Tampilan *Software* ISP *Programmer* (Adam Dybkowsky)

3.1.3 Perancangan Rangkaian Keypad dan LCD

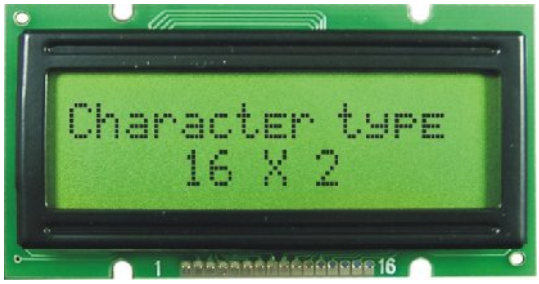
Keypad merupakan tombol elektronik yang terdiri dari kombinasi beberapa saklar yang terrangkai dalam bentuk kolom dan baris. Pada perancangan alat putar ini, *keypad* digunakan sebagai alat untuk masukan nilai *setpoint* kecepatan putaran alat putar keramik. *Keypad* yang digunakan adalah *keypad* 3×4 yang terdiri dari 3 kolom dan 4 baris (7 *pin*). Untuk mengetahui tombol mana yang sedang ditekan, *keypad* diatur oleh mikrokontroler dengan cara memberikan bit – bit logika pada baris atau kolomnya. *Keypad* ini dihubungkan melalui kabel *pin* (7 *pin*) ke salah satu *port* mikrokontroler. Rangkaian *keypad* ditunjukkan seperti Gambar 3.6 di bawah ini.



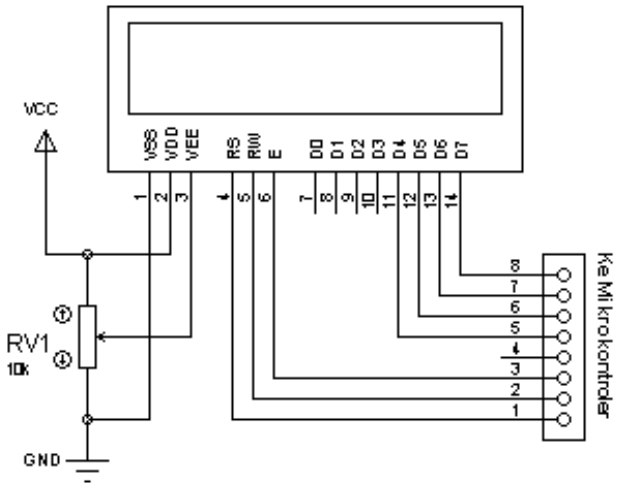
Gambar 3.6 Rangkaian *Keypad* 3×4

Keypad ini akan diaktifkan dan dideteksi oleh *bit – bit* logika dari *port – port* mikrokontroler (PORTB1…7). Bagian kolom *Keypad* akan diberi logika *low* (“0”) oleh mikrokontroler, sedangkan bagian baris akan diberi logika *high* (“1”). Pada setiap *port* pada mikrokontroler AVR ATMega16, telah terintegrasi rangkaian *pull-up resistor*, sehingga apabila salah satu baris dari *keypad* terhubung (*short*) dengan salah satu kolom, maka akan memberikan logika *low* pada baris yang terhubung tersebut. Cara mendeteksi *bit – bit* untuk bagian baris tersebut menggunakan teknik *scanning port.*

Liquid Crystal Display atau disingkat LCD merupakan suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. Pada perancangan ini, LCD akan digunakaan sebagai alat penampil kode ASCII (huruf dan angka) yang berdasarkan masukkan dari *keypad*. LCD yang digunakan berjenis LCD 16×2 seri DV-16230 Data Vision (Taiwan). LCD ini tidak memiliki cahaya latar (*back light*) akan tetapi dapat diatur kekontrasannya. Bentuk dan rangkaian LCD ditunjukkan oleh gambar di bawah ini.



Gambar 3.7 Bentuk LCD 16×2



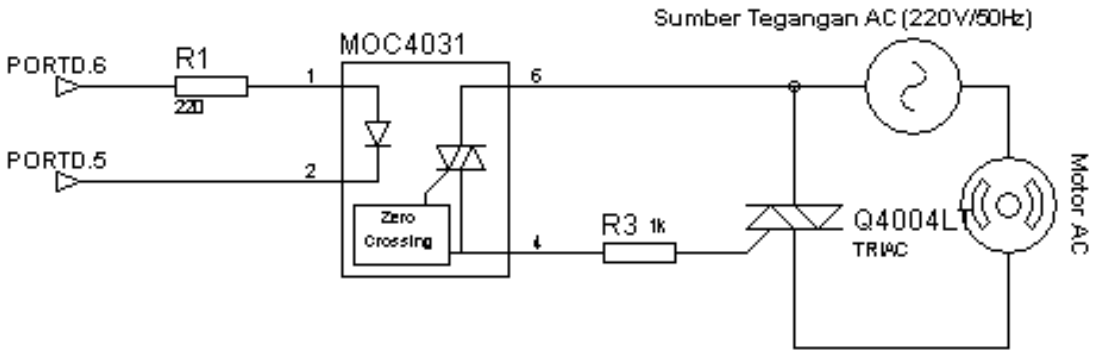
Gambar 3.8 Rangkaian LCD 16×2

Software Code Vision AVR telah menyediakan fitur LCD untuk mengatur *port* I/O LCD, sehingga tidak akan banyak mengalami kesulitan dalam merancang instruksi programnya. LCD akan menampilkan nilai kecepatan (*setpoint*), hasil deteksi sensor putaran dan nilai *register* OCR1A untuk mengatur *duty cycle Fast* PMW.

3.1.4 Perancangan Rangkaian *Driver* Motor AC

Komponen utama dari rangkaian *driver* motor AC ini adalah IC TRIAC Optoisolators Tipe MOC3041 dan TRIAC Tipe Q4004LT. TRIAC *Optoisolators* yang digunakan telah memiliki rangkaian *zero crossing* di dalamnya.. Rangkaian *driver* motor AC akan memicu motor AC jika *pin* 2 pada IC MOC3041 diberi logika “0”

(*low*). *Bit – bit* logika yang diberikan pada IC tersebut berupa sinyal PWM yang diatur oleh *register* OCR1A (*duty cycle*) pada mikrokontroler. Rangkaian *driver* motor AC ini ditunjukkan oleh Gambar 3.9 di bawah ini.

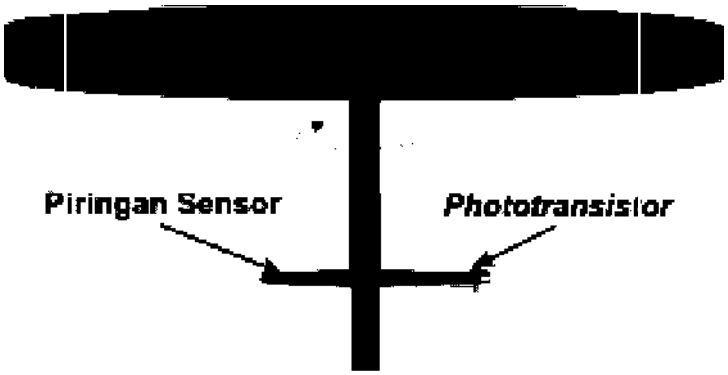


Gambar 3.9 Rangkaian *Driver* Motor AC

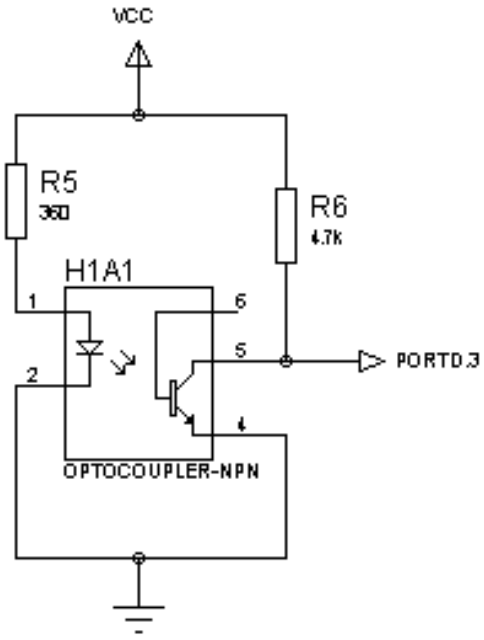
Rangkaian *Zero Crossing* yang terdapat IC MOC3041 berfungsi untuk menentukan titik nol gelombang sinusoidal, dimana titik nol tersebut merupakan titik acuan untuk dimulainya pemicuan oleh sinyal PWM.

3.1.5 Perancangan Rangkaian Sensor Putaran

Rangkaian sensor untuk mendeteksi putaran alat putar ini, terdiri dari *Phototransistor* tipe H21A1 dan piringan sensor dengan lubang sebanyak 12 lubang.



Gambar 3.10 Pemasangan Sensor Putaran Pada Alat Putar Keramik

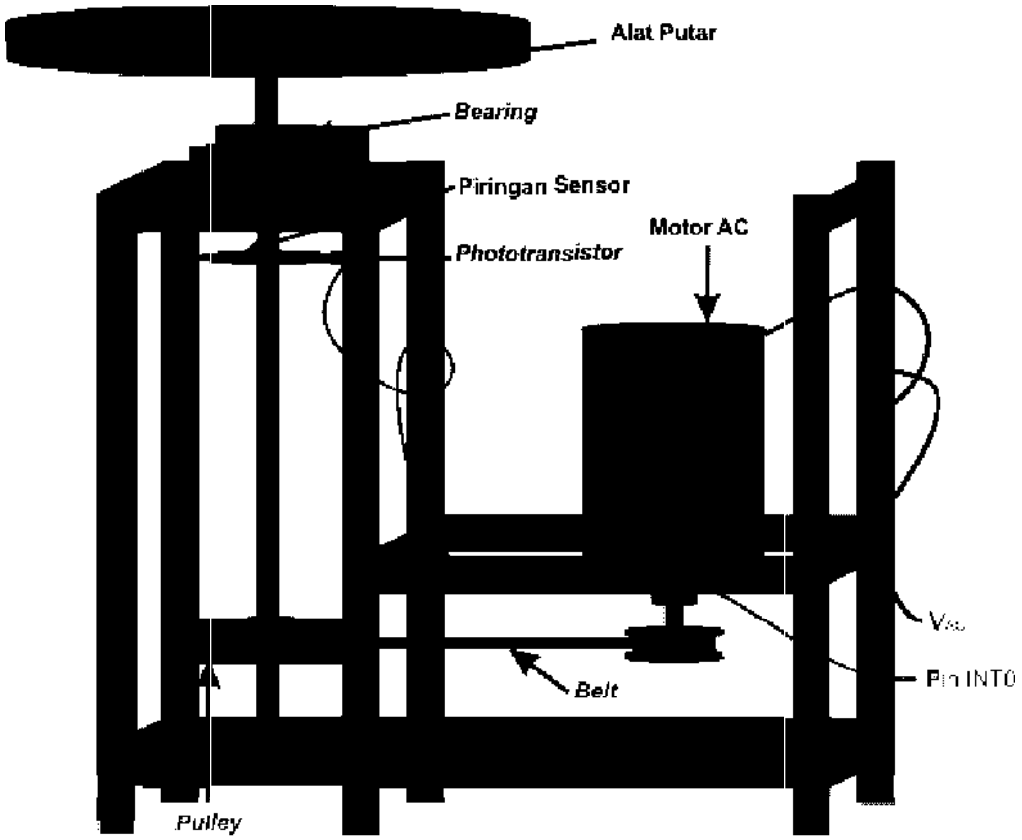


Gambar 3.11 Rangkaian *Phototransistor*

Sensor putaran akan memberikan logika "0" pada saat celah pada *phototransistor* mendeteksi lubang dan sensor putaran akan memberikan logika "1" pada saat celah pada *phototransistor* tidak mendeteksi lubang. Jarak antar lubang pada piringan sensor dari keenam lubang tersebut mewakili 5rpm dan banyaknya lubang yang dideteksi sensor akan dijumlahkan setiap 1 detik sekali oleh interupsi *Timer* 0 (Contoh: 60rpm = 12 lubang per detik). Hal ini bertujuan untuk membandingkan kecepatan antara kecepatan *setpoint* dan kecepatan *real time*. Pendeteksian ini menggunakan fitur interupsi eksternal 0 (*pin* INT0) pada mikrokontroler ATMEGA 16.

3.1.6 Perancangan Alat Putar Keramik

Alat putar keramik yang akan dirancang terbuat dari bahan semen dan berporoskan batang besi. Pada bagian besi poros tersebut dipasangi *bearing*, *pulley* dan piringan sensor. *Pulley* dihubungkan motor AC dengan menggunakan tali penghubung (*belt*). Gambar 3.12 di bawah ini menunjukkan gambar rancangan alat putar keramik.



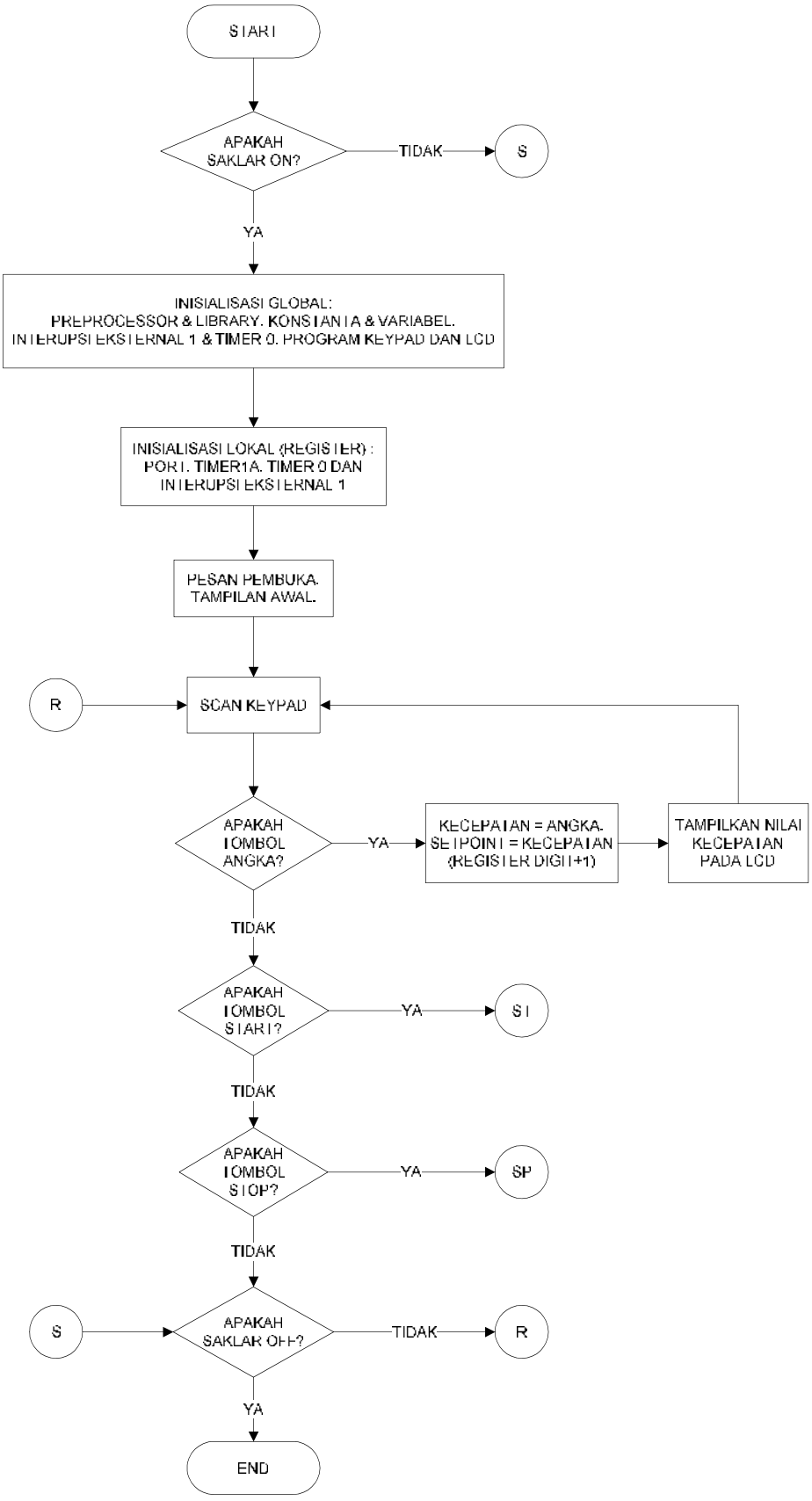
Gambar 3.12 Rancangan Alat Putar Keramik

Rangka terbuat dari besi pipih yang dihubungkan dengan mur dan baud. Rangka didesain sedemikian rupa (Gambar 3.12), sehingga kokoh dan dapat digunakan untuk menyimpan dan menyangga alat putar, motor AC dan sensor.

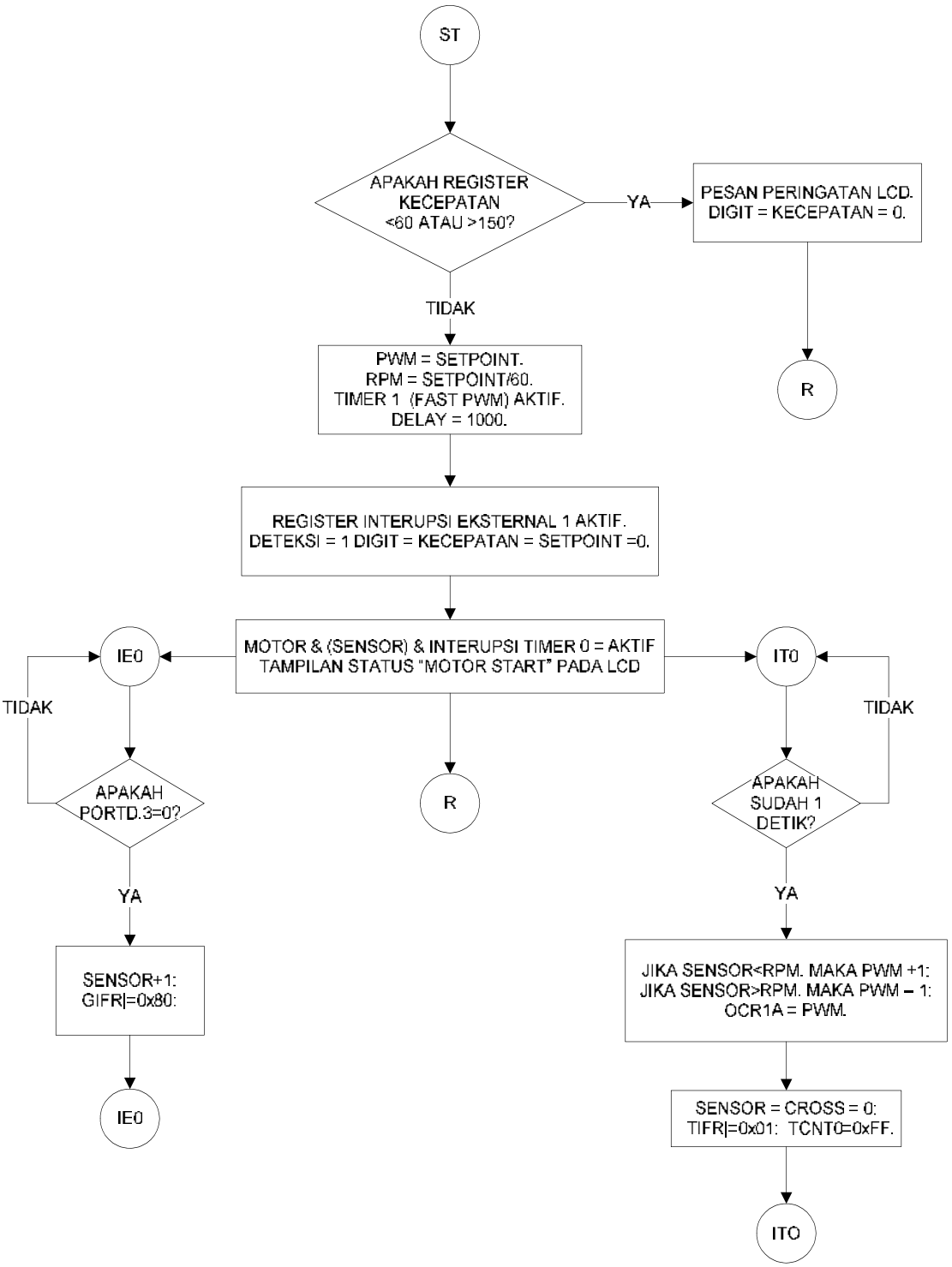
3.2 Perancangan Program Mikrokontroler

Tahap kedua dari perancangan ini adalah merancang suatu program mikrokontroler yang bertujuan untuk mengolah nilai suatu variabel (*keypad* dan sensor), menampilkan suatu karakter pada LCD, mengatur interupsi eksternal 1, mengubah nilai *setpoint* menjadi sinyal *Fast* PWM dan mengatur kecepatan motor AC.

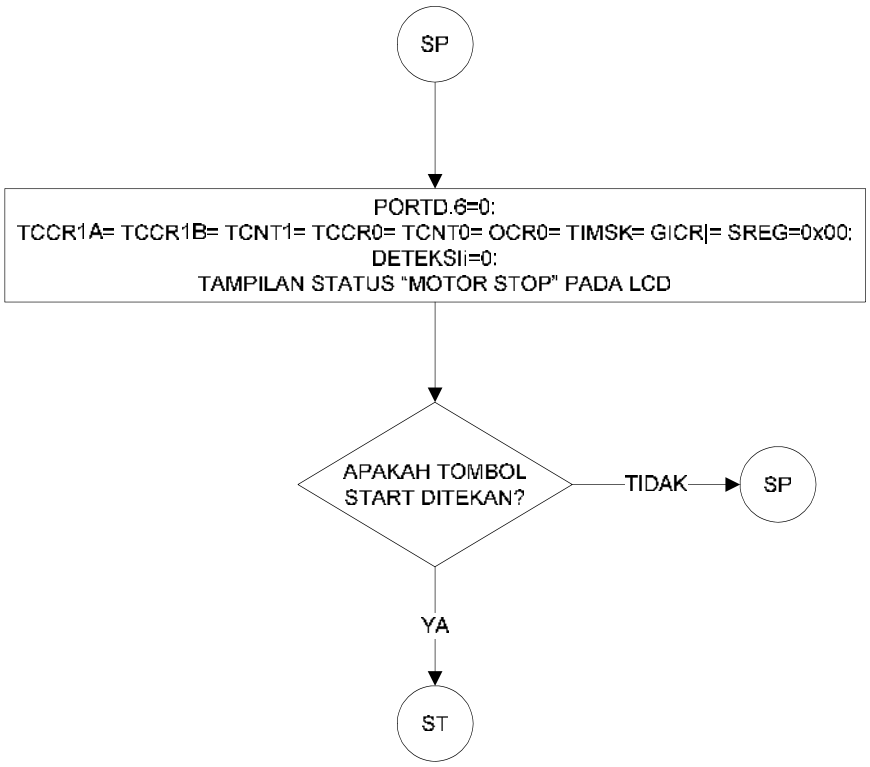
Program mikrokontroler yang akan dibuat menggunakan bahasa C dan beberapa bahasa *assembly* pada *software Code Vision*. Cara kerja dari program pengaturan motor AC terhadap beban bervariasi adalah seperti *flowchart* di bawah ini.



Gambar 3.13 *Flowchart* Pengaturan Kecepatan Putaran Alat Putar Keramik



Gambar 3.14 *Flowchart* Pengaturan Kecepatan Putaran Alat Putar Keramik (*Lanjutan* I)



Gambar 3.15 *Flowchart* Pengaturan Kecepatan Putaran Alat Putar Keramik (*Lanjutan* II)

Interupsi eksternal 0 digunakan untuk mendeteksi bit "0" pada PORD.3 (*pin* INT0). Banyaknya bit "0" yang masuk pada PORD.3 akan dihitung dan disimpan dalam *register* "sensor" setiap 1 detik sekali. Dengan kata lain, nilai pada *register* "sensor" akan dinolkan terlebih dahulu jika sudah mencapai 1 detik.

Interupsi timer 1 digunakan untuk membuat interupsi setiap 1 detik. Pada saat interupsi timer 1 ini terjadi, nilai pada *register* "sensor" akan di-*update* dan dibandingkan dengan nilai *register* "PWM". Nilai register OCR1A akan diubah (ditambah/dikurangi) apabila hasil perbandingan berbeda.

Timer/Counter 1A berfungsi sebagai pembangkit sinyal *Fast* PWM. Besar *duty cycle* dari *Fast* PWM ini diatur oleh register OCR1A, sehingga dari *register* inilah motor AC dapat diatur kecepatannya.

3.2.1 Program *Keypad*

Listing program *keypad* yang digunakan pada perancangan alat putar ini dapat dilihat pada potongan program dibawah ini.

```
#include <mega16.h>
#include <delay.h>
#define DDR_Keypad      DDRA
#define PORT_Keypad     PORTA
#define PIN_Keypad      PINA
#define Hi_Keypad(x)    PORT_Keypad|=(1<<x)
#define Low_Keypad(x)   PORT_Keypad&=~(1<<x)
#define Is_Hi_(x)       PIN_Keypad&(1<<x)

unsigned char SCAN_KEYPAD()
{
unsigned char tombol[12]={'1','4','7','*','2','5','8','0','3','6','9','#'};
unsigned char tombol_ke=0, baris, kolom, key_press;
key_press=('$');
DDR_Keypad=0x07;
for (kolom==0; kolom<=2; kolom++)
    {
     PORT_Keypad=0xFF;
     Low_Keypad(kolom);
     for (baris==3; baris<=6; baris++)
         {
         if (~Is_Hi(baris))  {key_press=tombol[tombol_ke];}
         }
    }
    delay_ms(50);   //anti bouncing
    return key_press;
}
unsigned char key;
void main(void)
{


while (1)
      {
      key=SCAN_KEYPAD();    //pemanggilan fungsi KEYPAD
      // program utama
      };
```

Gambar 3.16 Rancangan *Listing* Program *Keypad* Pada *Code Vision* AVR

Keypad diaktifkan dengan cara memberikan bit *low* pada bagian kolom *keypad* dan memindai bit *low* pada bagian baris.

3.2.2 Program LCD

Listing program LCD yang digunakan pada perancangan alat putar ini dapat dilihat pada potongan program dibawah ini.

```
#include <mega16.h>
#include <lcd.h>
#include <delay.h>

//fungsi modul LCD alfanumerik
#asm
    .equ __lcd_port=0x12  ;PORTD
#endasm

unsigned char posx,posy;

void main(void)
{
PORTD=0xFF;
DDRD=0xFF;
//inisialisasi LCD
lcd_init(16);   //LCD yg digunakan 16X2
posx=0; posy=0; // variabel untuk kolom dan baris
lcd_clear();    //menghapus karakter pada LCD
lcd_gotoxy(posx,posy);      //pengalamatan kolom dan baris
lcd_putsf("Alat Putar");    //menampilkan karakter (String)
lcd_gotoxy(posx,1);
lcd_putsf("Keramik");
delay_ms(2000);


while (1)
      {
      lcd_clear();
      lcd_putchar(XXX)      //menampilkan karakter (Variabel dalam bahasa C)
      // program utama lainnya
      };
}
```

Gambar 3.17 Rancangan *Listing* Program LCD Pada *Code Vision* AVR

3.2.3 Program Interupsi Eksternal 1 dan *Timer/Counter* 0

Program interupsi eksternal 1 diatur oleh fitur interupsi eksternal 0 (INT0) ATMEGA16. Interupsi eksternal 1 ini akan berfungsi jika *register* – *register* yang berhubungan dengan interupsi eksternal 1 diatur. Interupsi eksternal 1 akan terjadi apabila PORTD.3 diberi logika 0 (*clear bit*). *Listing* program Interupsi Eksternal 1 yang digunakan pada perancangan alat putar ini dapat dilihat pada potongan program dibawah ini.

```
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
delay_ms(20);
sensor++;
GIFR|=0x80;
}
```

Gambar 3.18 Rancangan Program Sensor Putaran Pada *Code Vision* AVR

Jika *phototransistor* mendeteksi lubang pada piringan sensor, maka rangkaian sensor akan memberikan logika "0" ke PORTD.3 dan logika 1 jika tidak mendeteksi lubang.

Program *Timer/Counter* 1 pada perancangan ini digunakan untuk membuat interupsi setiap 1 detik. Besarnya waktu selama 1 detik ini dapat diperoleh dengan menggunakan persamaan (2.9). *Timer/Counter* 0 (TCNT0) hanya mempunyai register sebesar FF (255), sehingga T_{MAX} dapat dihitung sebagai berikut.

$$= \frac{}{} \times (\quad h +)1 \times \quad = \frac{}{} \times (256) \times 1024 \; = 64 \cdot 10 \quad \times 1024$$

$$= 65{,}536\text{ms}.$$

Untuk mempermudah mendapatkan waktu 1 detik dapat dilakukan dengan cara membuat interupsi setiap 20mS, kemudian pada program interupsi tersebut ditambahkan program *counter* untuk menghitung kejadian interupsi sebanyak 50 kali. Contoh *listing* program interupsi pada *Timer/Counter* 0 seperti dibawah ini.

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
cross++;
if (cross==50)   // untuk mendapatkan waktu 1 detik
    {
    if (sensor<rpm) {PWM++; OCR1A=PWM;}
    if (sensor>rpm) {PWM--; OCR1A=PWM;}
    sensor=0;
    cross=0;
    };
TIFR|=0x01; TCNT0=0xFF;  //reinisialisasi timer
}
void main(void)
{
//inisialisasi lokal variabel
while (1)
    {
    TCCR0=0x05; ; TCNT0=0xFF;  OCR0=0x02;
    }
}
```

Gambar 3.19 Rancangan *Listing* Program Interupsi Pada *Timer/Counter* 0

3.2.4 Program *Fast* PWM

Motor AC yang digunakan untuk memutar alat putar keramik, menggunakan listrik bertegangan 220VAC dengan frekuensi sebesar 50Hz. Tegangan tersebut akan diatur oleh sinyal PMW dari mikrokontroler dengan memanfaatkan fitur *Timer/Counter* 1 dengan mode *Fast* PWM. Tegangan AC dalam satu periode mempunyai 2 siklus, yaitu siklus positif dan siklus negatif. Sinyal PWM akan mengatur tegangan AC tersebut setiap 1 siklus (positif dan negatif), dimana 1 siklus tegangan AC tersebut mempunyai periode sebesar:

$$= - = — = 20$$

$$- \quad - = —— = 10$$

Maka, *Fast* PWM yang diatur oleh *Timer/Counter* 1 akan mengatur kedua siklus tersebut setiap 10mS sekali. Tatacara untuk mengatur *Fast* PWM tersebut telah dibahas dalam Bab II. Dengan menggunakan persamaan (2.3), maka diperoleh hasil perhitungan *Fast* PWM sebagai berikut.

$$f_{OCnxPWM} = \frac{\quad}{\cdot(\quad)}$$

$$50\text{Hz} = \frac{\quad}{\cdot(\quad)} = \frac{\quad}{\cdot(\quad)}$$

$$\text{N} = —— = 78{,}125$$

Nilai *prescaler* yang tersedia hanya sebesar 0, 1, 8, 64, 256 dan 1024, maka diambil yang nilai lebih mendekati nilai tersebut, yaitu 64.

Listing program *Fast* PWM yang digunakan pada perancangan alat putar ini dapat dilihat pada potongan program dibawah ini.

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


void main(void)
{

TCCR1A=0xF3;     //Pengaturan Fast PWM
TCCR1B=0x0C;     //Prescaler = 64
TCNT1=0x00;
OCR1A=0;
PORTD.1=1;

while (1)
      {
      //program utama
      };
}
```

Gambar 3.20 Rancangan *Listing* Program *Fast* PWM Pada *Code Vision* AVR

3.2.5 Program Pengaturan Kecepatan Motor AC

Listing program pengaturan motor AC yang digunakan pada perancangan alat putar ini dapat dilihat pada potongan program dibawah ini.

```
if (setpoint<60 | setpoint>300)
        {
        digit=0; kecepatan=0;
        lcd_clear();
        for (disp=0;disp<4;disp++)
            {
            lcd_gotoxy(0,0);
            lcd_putsf(" Minimal 60rpm");
            lcd_gotoxy(0,1);
            lcd_putsf("Maksimal 300rpm");
            delay_ms(150);
            lcd_clear();
            delay_ms(125);
            }
        disp=0;
        lcd_init(16);
        goto miss;
        }

PWM=setpoint;     //Fast PWM = setpoint

TCCR1A=0xF3;  TCCR1B=0x0C;  TCNT1=0x00; OCR1A=PWM;  PORTD.6=1;
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("  MOTOR START");

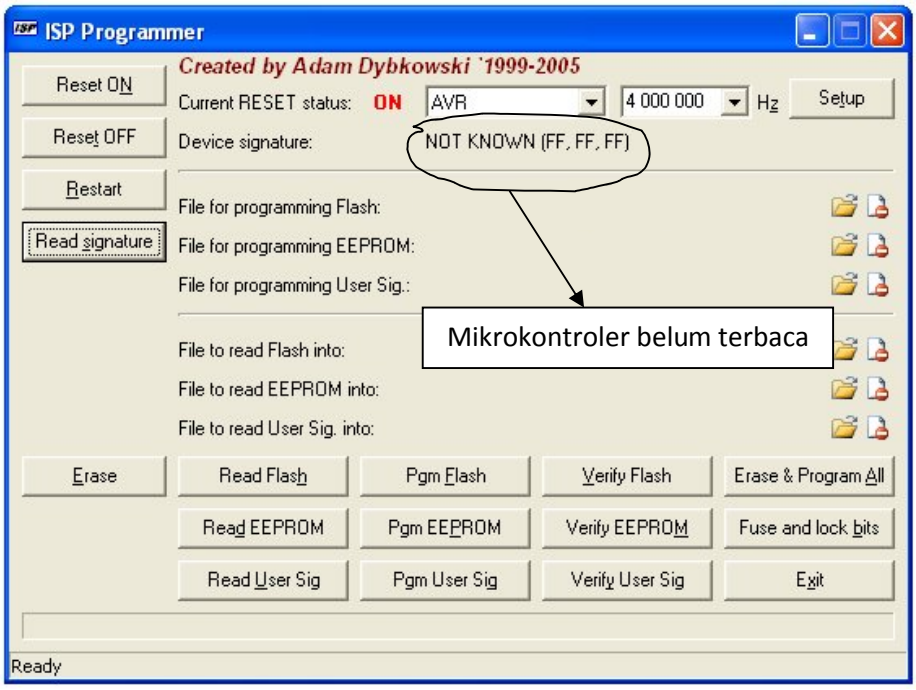
rpm=setpoint/60;
delay_ms(1000);     //delay untuk menunggu putaran konstan
GICR|=0x80; SREG=0x80;
TCCR0=0x05; ; TCNT0=0xFF;  OCR0=0x01; TIMSK=0x01;
deteksi=1; digit=kecepatan=setpoint=0;
lcd_clear();
```

Gambar 3.21 Rancangan *Listing* Program Pengaturan Kecepatan Motor AC Pada *Code Vision* AVR

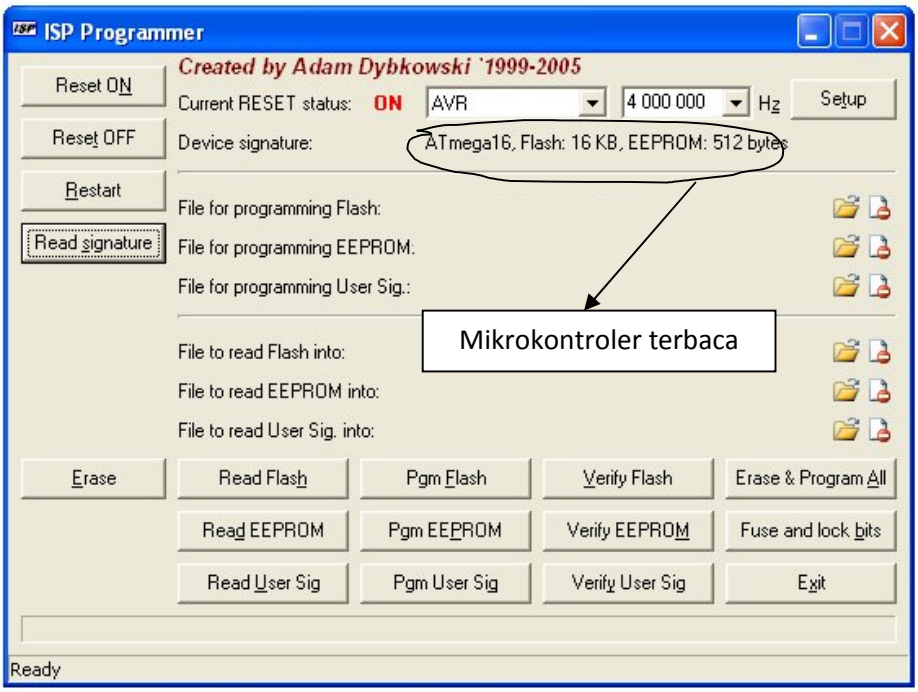
3.2.6 Pengisian Program Ke Mikrokontroler

Listing program yang telah di-*compile* ke dalam bentuk *file* *.*hex* oleh *Code Vision* AVR dapat langsung diisikan ke dalam mikrokontroler. Berikut ini langkah – langkah pengisian program ke mikrokontroler ATMega 16 dengan menggunakan *software* ISP *Programmer.*

1. Rangkaian downloader (DB25) dihubungkan terlebih dahulu ke port DB25 (komputer) menggunakan kabel konektor DB25 dan rangkaian *downloader* (mikrokontroler) menggunakan kabel konektor 6 *pin.*
2. Rangkaian *downloader* diberi tegangan sebesar 5V.
3. Komputer diaktifkan dan *Software* ISP *Programmer* dijalankan, sehingga tampil jendela seperti Gambar 3.5, kemudian pilih “AVR” pada *List Box* di sebelah “Current RESET Status” dan pilih frekuensi sebesar 4MHz.
4. Untuk mengetahui apakah rangkaian *downloader* berfungsi dan mikrokontroler terbaca, tombol “Read signature:” ditekan. Jika mikrokontroler belum terbaca, maka akan tampil seperti Gambar 3.22 di bawah ini. Jika mikrokontroler terbaca, maka akan tampil seperti Gambar 3.23.

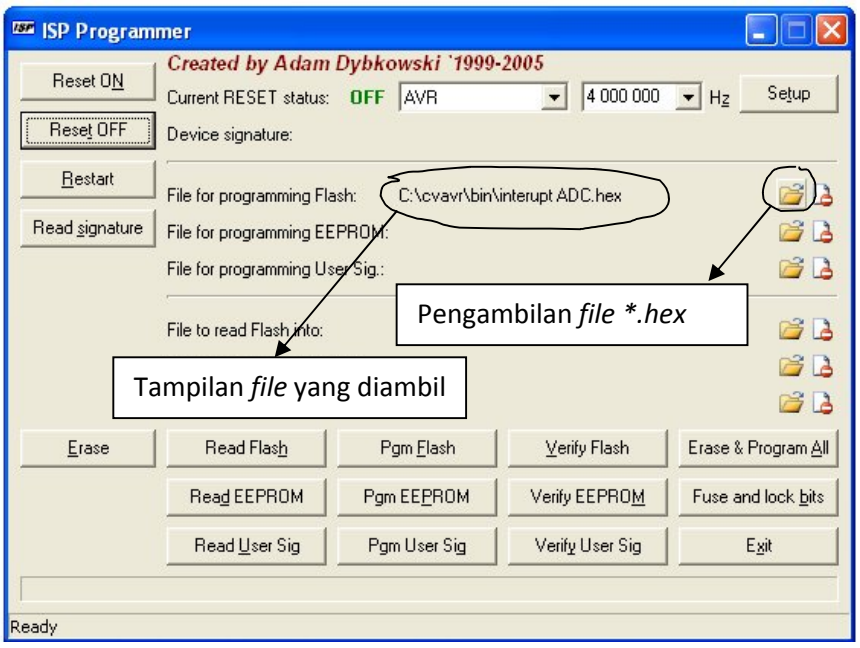


Gambar 3.22 Tampilan Jika Mikrokontroler Belum Terbaca



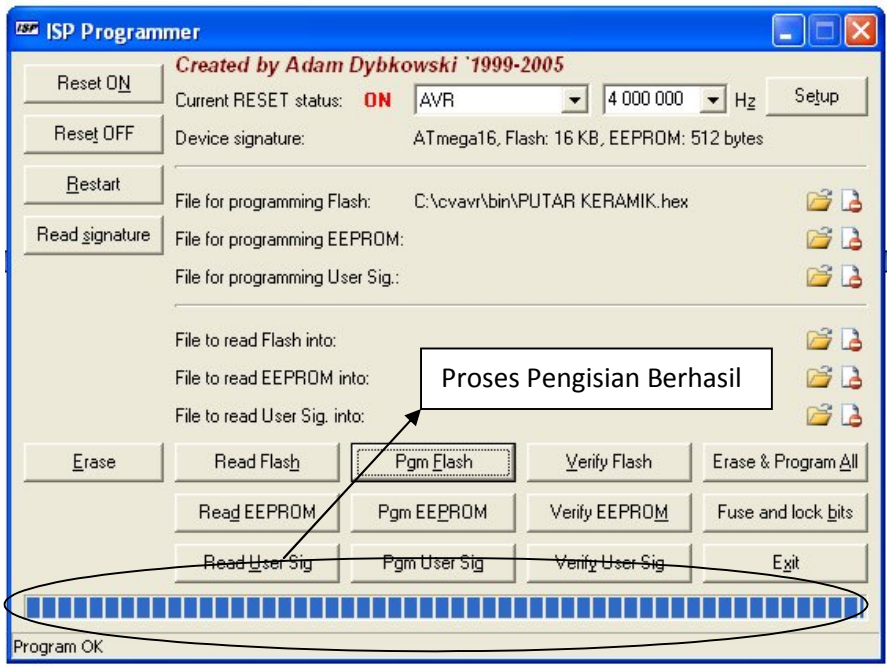
Gambar 3.23 Tampilan Jika Mikrokontroler Sudah Terbaca

5. *File* *.hex yang akan diisikan pada mikrokontroler diambil dari "File for programming Flash:", pilih *file* yang akan digunakan seperti gambar di bawah ini.



Gambar 3.24 Cara Pengambilan File *.*Hex* Pada *Flash* ISP *Prog.*

6. Jika langkah – langkah diatas telah berhasil, maka mikrokontroler siap untuk diisikan program. Tombol "Erase & Program All" ditekan terlebih dahulu, kemudian "Restart" dan tombol "Pgm Flash" ditekan. Proses pengisian selesai jika tampil gambar seperti dibawah ini.



Gambar 3.25 Tampilan Jika Mikrokontroler Telah Selesai/Berhasil Diisi