

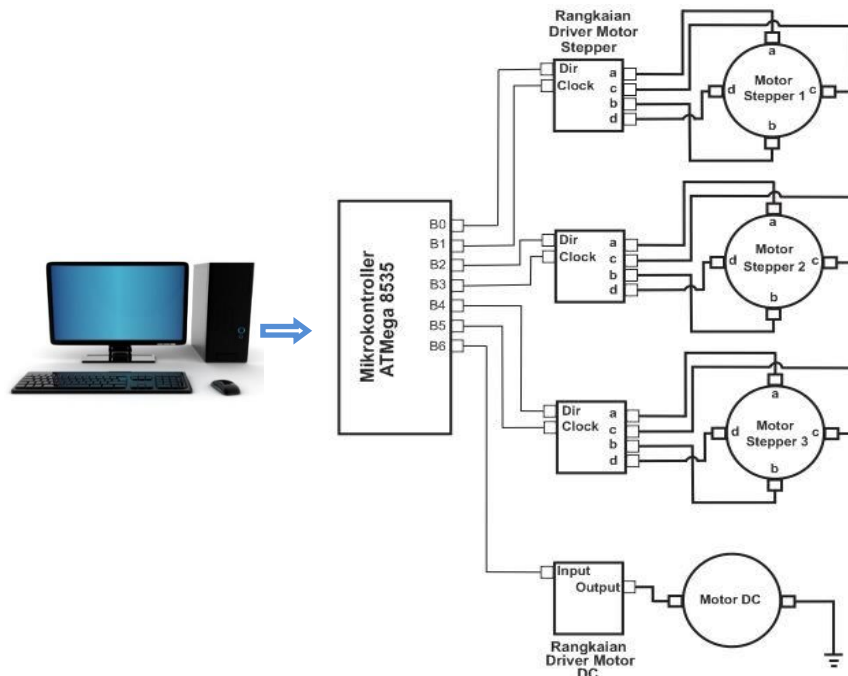
BAB III

PENGENDALIAN PENGGERAK PAHAT MESIN ROUTER CNC ARAH SUMBU X, SUMBU Y DAN SUMBU Z

Pada bab ini dibahas mengenai rangkaian elektronika yang akan digunakan untuk mengendalikan gerak pahat dan program yang akan dibuat pada *software Code Vision AVR* serta *Visual Basic 6.0*. Untuk dapat menggerakkan mesin router otomatis ini dibutuhkan beberapa rangkaian elektronika yaitu rangkaian penggerak motor stepper dan rangkaian mikrokontroler ATmega 8535.

3.1 Rangkaian Pengendalian Penggerak Pahat Mesin Router CNC Sumbu X, Sumbu Y dan Sumbu Z

Rangkaian pengendalian penggerak pahat mesin router CNC sumbu X, sumbu Y dan sumbu Z merupakan rangkaian pengendali motor penggerak (motor stepper). Rangkaian pengendali motor stepper penggerak tersebut diprogram menggunakan komputer (*Visual Basic*) dengan cara menghubungkan rangkaian mikrokontroler dengan tiga buah *driver* motor stepper yang mempunyai dua kaki *input* yaitu kaki *dir* dan *clock*. Skematis rangkaian pengendali penggerak pahat sumbu X sumbu Y dan sumbu Z dapat dilihat pada gambar 3.1.

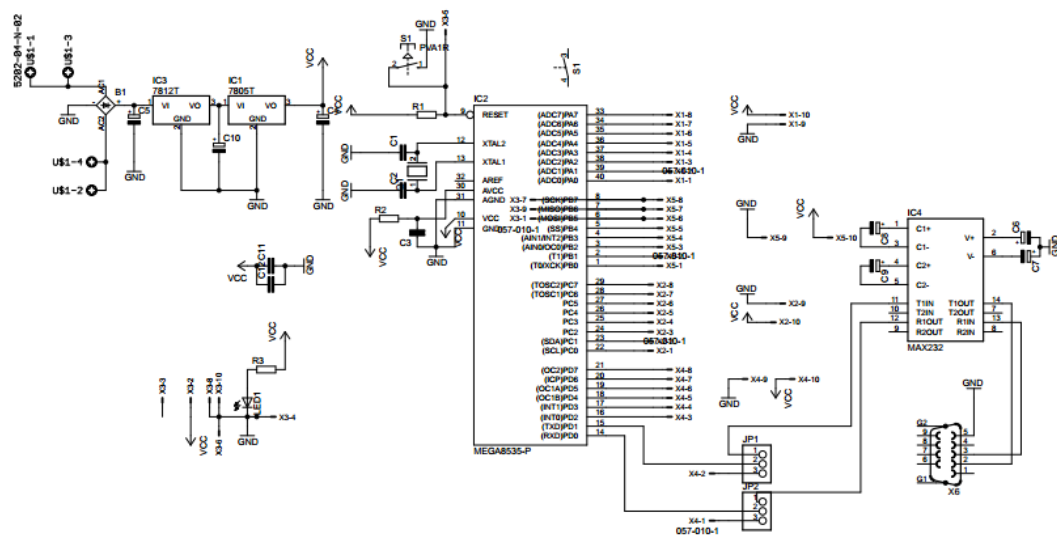


Gambar 3.1 Skematis Rangkaian Pengendali Penggerak Pahat Sumbu X Sumbu Y dan Sumbu Z

Rangkaian elektronika yang dihubungkan dengan mikrokontroler adalah tiga buah rangkaian *driver* motor stepper. Skematis rangkaian pengendali penggerak pahat mesin router CNC sumbu X, sumbu Y dan sumbu Z dapat dilihat pada gambar 3.1.

3.1.1 Rangkaian mikrokontroler

Mikrokontroler merupakan sebuah piranti elektronik berupa *integrated circuit (IC)* yang memiliki kemampuan memanipulasi data berdasarkan instruksi program yang dibuat. Mikrokontroler yang digunakan pada pengendalian penggerak pahat mesin router CNC pada sumbu X, sumbu Y dan sumbu Z adalah mikrokontroler ATmega8535. Mikrokontroler ini berfungsi untuk mengolah data-data yang dikirim dari komputer ke mikrokontroler. Rangkaian mikrokontroler ini telah dilengkapi jalur komunikasi serial yang menghubungkan antara komputer dan mikrokontroler. Skematik rangkaian mikrokontroler ATmega8535 dapat dilihat pada gambar 3.2.

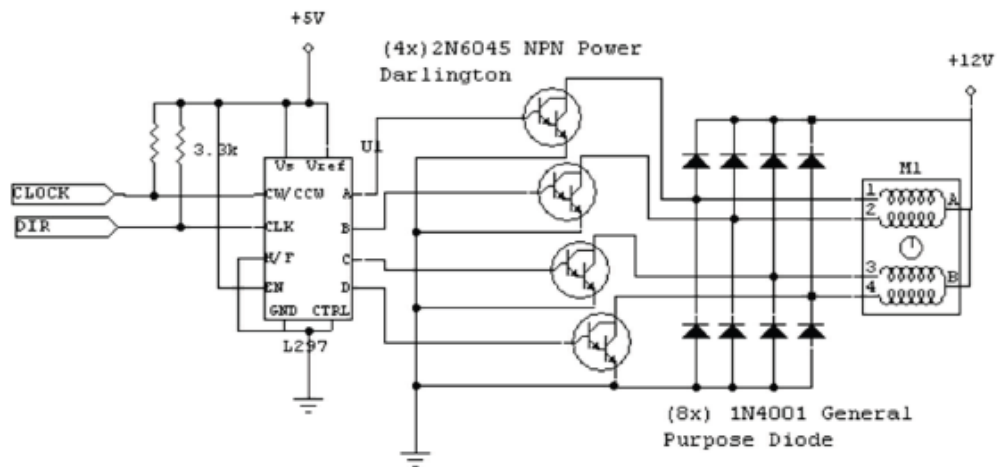


Gambar 3.2 Skematik Mikrokontroler ATmega8535

3.1.2 Rangkaian driver motor stepper

Driver motor stepper merupakan perangkat elektronika yang berfungsi sebagai media program untuk menggerakkan motor stepper. Dalam menjalankan motor stepper, *driver* ini mempunyai *IC (Integrated Circuit)* khusus yang digunakan untuk motor stepper yaitu *IC L297*. *Output* rangkaian *driver* motor stepper L297 memberikan nilai *high/low* pada kaki keluaran (A, C, B, dan D) secara berurutan dan bergantian terhadap motor stepper. *Driver* motor stepper L297 mempunyai dua *input* fungsional yaitu pulsa (*clock*) dan arah putaran

(*direction*). Skema rangkaian penggerak motor stepper pada tugas akhir ini dapat dilihat pada gambar 3.3.



Gambar 3.3 Skema *Driver* Motor Stepper

3.2 Instalasi Sistem Pengendali Mesin Router CNC

Instalasi sistem pengendali mesin router pada arah sumbu X, sumbu Y dan sumbu Z merupakan urutan pemasangan peralatan – peralatan sistem pengendali pada mesin router. Peralatan sistem pengendali yang digunakan adalah catu daya 12 V, *Driver IC* L297, dan mikrokontroler ATmega 8535. Langkah – langkah pemasangan peralatan sistem pengendali adalah sebagai berikut :

1. Sumber arus catu daya 12 V dihubungkan kepada mikrokontroler,
2. Mikrokontroler dihubungkan ke perangkat elektronik yaitu rangkaian *driver IC* L297 untuk sumbu X, sumbu Y dan sumbu Z,
3. Kaki output L297 pada *driver* motor stepper satu dihubungkan kepada kaki KOIL (A, C, B dan D) untuk motor stepper sumbu X,
4. Kaki output L297 pada *driver* motor stepper dua dihubungkan kepada kaki KOIL (A, C, B dan D) untuk motor stepper sumbu Y,
5. Kaki output L297 pada *driver* motor stepper tiga dihubungkan kepada kaki KOIL (A, C, B dan D) untuk motor stepper sumbu Z.

3.3 Program Pengendali

Program untuk menggerakkan tiga buah motor stepper pada mesin router dilakukan secara serial dengan menggunakan bahasa c. Jenis pengiriman data komunikasi serial yang digunakan pada TA ini adalah pengiriman data dari komputer ke mikrokontroler. Data yang

dikirim oleh komputer biasanya diolah kembali dan data hasil olahan tersebut digunakan untuk melakukan suatu aktivitas seperti mengaktifkan pemanas, membuka saluran air, menggerakkan motor dan lain – lain. Aktivitas tersebut dikerjakan oleh mikrokontroler. Supaya mikrokontroler dapat menjalankan aktivitas tersebut, mikrokontroler perlu data masukan dari komputer.

Selain memerlukan data masukan dari komputer, mikrokontroler harus mempunyai program yang berfungsi untuk menerima data dari komputer, mengolah data dari komputer dan mengeksekusi data dari komputer untuk menjalankan tiga motor stepper. Aplikasi yang digunakan untuk membuat program di komputer adalah *Visual Basic 6.0* sedangkan aplikasi yang digunakan untuk membuat program di mikrokontroler adalah *CodeVision AVR*. Alasan pemilihan menggunakan kedua aplikasi tersebut karena aplikasi tersebut sangat mudah digunakan dan memiliki kemampuan untuk melakukan komunikasi secara serial.

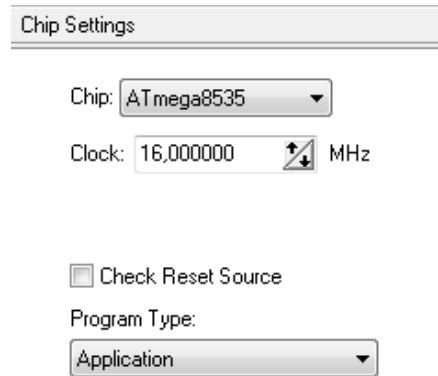
3.3.1 *Code vision AVR*

Program *CodeVisionAVR* digunakan sebagai program penghubung antara komputer dan *driver* motor stepper. Program yang dibuat pada *CodeVisionAVR* adalah program yang dapat mengirim data dari mikrokontroler ke komputer serta program yang dapat menerima data dari komputer ke mikrokontroler. Data yang diterima mikrokontroler dari komputer berupa sinyal digital. Sinyal digital tersebut selanjutnya dikirim ke *driver* motor stepper untuk melakukan gerakan motor.

Agar hal ini dapat dilakukan, diperlukan suatu komunikasi antara mikrokontroler dengan komputer. Salah satu jenis komunikasi yang sering digunakan antara mikrokontroler dengan komputer adalah komunikasi serial secara asinkron.

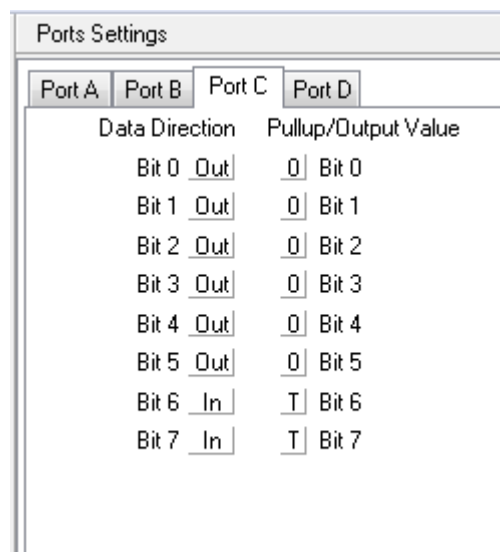
Program tugas akhir yang dibuat menggunakan *Code Vision AVR*. Pada *Code Vision AVR* terdapat beberapa fitur yang harus diatur. Fitur – fitur tersebut diantaranya pemilihan jenis *chip* yang digunakan, *clock* yang digunakan, *PORT* yang digunakan, pengaturan beberapa kaki pada *PORT* untuk fungsional kaki *dir* serta *clock* pada *IC L297*, pengaturan *USART* untuk komunikasi serial dan pengaturan *timer*.

Pengaturan pertama adalah pemilihan jenis *chip* dan nilai *clock*. *Chip* yang digunakan adalah mikrokontroler *ATMega 8535*. Nilai *clock* yang dipilih adalah dengan nilai *clock* 16 MHz. Tampilan pengaturan *chip* dan nilai *clock* dapat dilihat pada gambar 3.4.



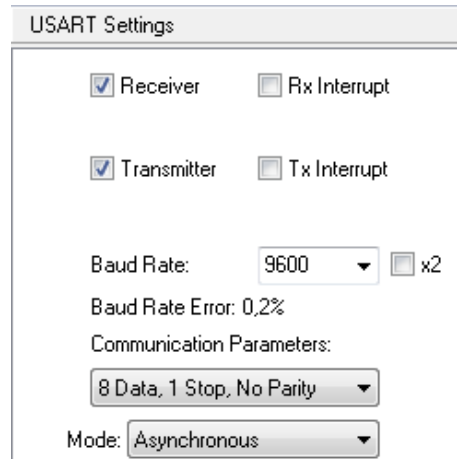
Gambar 3.4 Tampilan pengaturan *chip* dan *clock*

Mikrokontroler ATmega8535 mempunyai 4 *PORT*, dimana setiap *PORT* mempunyai 8 kaki. Pada tugas akhir ini *PORT* yang digunakan adalah *PORTC* dimana kaki 1 sampai kaki 6 (bit 0 sampai bit 5) diatur sebagai *output*. *PORTC* pada kaki *output* bit 0, bit 2 dan bit 3 digunakan untuk melakukan pengiriman sinyal *clock* pada *driver* motor stepper 1, *driver* motor stepper 2 dan *driver* motor stepper 3. *PORTC* pada kaki *output* bit 1, 3 dan 5 digunakan untuk melakukan pengiriman sinyal *dir* pada *driver* motor stepper 1, *driver* motor stepper 2 dan *dirver* motor stepper 3. Tampilan pengaturan *PORT* yang digunakan dapat dilihat pada gambar 3.5.



Gambar 3.5 Tampilan Pengaturan *PORT* Yang Digunakan

Program untuk menggerakkan tiga buah motor stepper dilakukan secara serial. Oleh sebab itu supaya program yang dibuat dapat dikirim secara serial, diperlukan pengaturan *USART* pada fitur *Code Vision AVR*. Pengaturan *USART* yang diatur adalah dengan memberi tanda centang pada kolom *receiver* dan *transmitter*. Tampilan pengaturan *USART* dapat dilihat pada gambar 3.6.

Gambar 3.6 Tampilan Pengaturan *USART*

Tiga buah motor stepper yang digerakkan oleh program harus bergerak secara bersamaan. Untuk menjalankan tiga buah motor stepper secara bersamaan, diperlukan suatu perintah yang digunakan memanggil fungsi dimana fungsi tersebut terus dikerjakan secara berulang – ulang selama batas fungsi tersebut dipenuhi. Perintah untuk menjalankan hal tersebut adalah *timer*.

Pada mikrokontroler ATMega 8535 terdapat tiga buah fasilitas *timer* yaitu *timer0*, *timer1* dan *timer 3*. *Timer0* mempunyai kapasitas maksimum 8 bit, *timer1* mempunyai kapasitas maksimum 16 bit dan *timer2* mempunyai kapasitas maksimum 8 bit. Kapasitas *timer* dapat diatur dengan cara memberikan nilai awal pada *timer* tersebut. Jika *timer0* diberi nilai awal 5 maka kapasitas timer tersebut yang semula 255 (8bit) akan berubah menjadi 250. Jika frekuensi pengisian *timer* yang dipilih adalah 250 kHz maka waktu yang diperlukan untuk mengisi *timer* adalah 0,001 detik atau 1 mili detik. Secara umum waktu yang diperlukan untuk mengisi suatu *timer* dimulai dari kondisi kosong sampai penuh dapat ditentukan menggunakan persamaan 3.1.

$$t = \frac{\text{kapasitas timer} - \text{nilai awal}}{\text{frekwensi pengisian timer}} \dots \dots \text{persamaan 3.1}$$

Timer yang digunakan pada program tugas akhir ini adalah *timer0*. Parameter yang perlu diatur pada *timer0* adalah menentukan nilai frekuensi (*clock value*) dengan nilai 2000 kHz, memberikan tanda ceklis pada *overflow Interrupt* dan memberikan nilai awal *timer0* (*timer value*) dengan nilai 37 h. Jika frekuensi *timer* bernilai 2000 kHz, nilai awal *timer0* bernilai 37 h maka waktu yang diperlukan untuk mengerjakan *timer* tersebut adalah 0,1 ms. Tampilan pengaturan *timer* dapat dilihat pada gambar 3.7.

Timers/Counters Settings

Timer0 Status Requirements

Timer Period: 0,1005 ms

Timer0 Timer1 Timer2 Watchdog

Clock Source: System Clock ▼

Clock Value: 2000,000 kHz ▼

Mode: Normal top=0xFF ▼

Output: Disconnected ▼

Overflow Interrupt

Compare Match Interrupt

Timer Value: 37 h

Compare: 0

Gambar 3.7 Tampilan Pengaturan *Timer*

Fungsi yang terkait dengan *timer0* adalah fungsi **interrupt [TIM0_OVF] void timer0_ovf_isr(void)**. Fungsi ini akan dieksekusi setiap selang waktu 0,1 ms. Fungsi tersebut merupakan fungsi untuk mengatur nilai *clock* pada *driver* motor stepper 1, *driver* motor stepper 2 dan *driver* motor stepper 3. Fungsi **interrupt [TIM0_OVF] void timer0_ovf_isr(void)** dapat dilihat pada tabel 3.1.

Tabel 3.1 FungsiInterrupt [TIM0_OVF] Void Timer0_Ovf_Isr(Void)

```

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x37;
// Place your code here
//motor stepper 1
if(a1==1)
{
/*
(1) a0=a0+1;
(2) if(a0<waktu1) PORTC.0=1;
    else PORTC.0=0;

(3) if(a0>2*waktu1)
    {**
        a0=0;
        a2=a2+1;
    }**
(4) if(a2>=jumlah1)
    {***
        a1=0;
    }***
}*/

//motor stepper 2
if(b1==1)
{
    b0=b0+1;
    if(b0<waktu2) PORTC.2=1;
    else PORTC.2=0;

if(b0>2*waktu2)
    {
        b0=0;
        b2=b2+1;
    }

if(b2>jumlah2)
    {
        b1=0;
    }

//motor stepper 3
if(c1==1)
{
    c0=c0+1;
    if(c0<waktu3) PORTC.2=1;
    else PORTC.2=0;
}
}

```



```

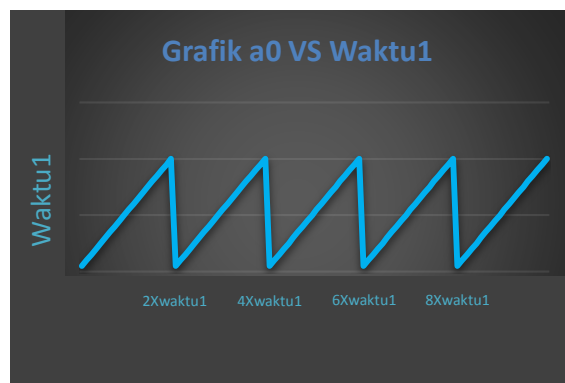
if(c0>2*waktu3)
{
    c0=0;
    c2=c2+1;
}

if(c2>jumlah3)
{
    c1=0;
}

(5)check1 = 0;
    check2 = 0;
(6)if((a1==0)&&(b1==0)&&(c1==0)) check1=1;
(7)if((a2>0)&&(b2>0)&&(c2>0)) check2=1;
(8)if ((check1==1)&&(check2==1))
{
    ****
    a2=0;
    b2=0;
    c2=0;
    putchar(0);
}****
}

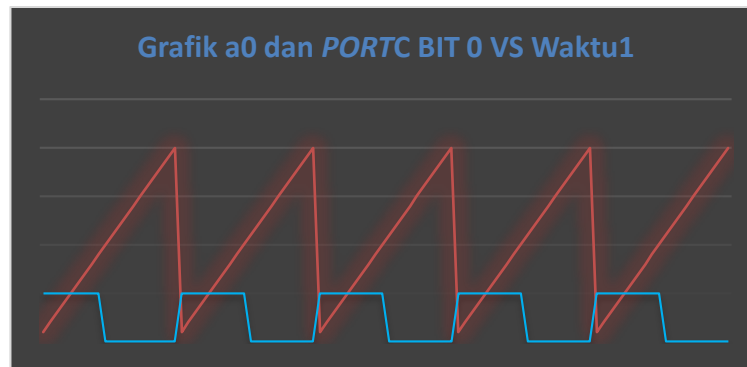
```

Pada fungsi interrupt semua perintah yang didalam kurung kurawal (ditandai dengan *) dikerjakan jika a1 sama dengan 1. Perintah pertama adalah variabel a0 selalu bertambah 1 (baris 1). Jika a0 lebih besar dari dua kali waktu1 maka a0 akan direset menjadi nilai 0 (baris 3). Grafik a0 terhadap waktu1 dapat dilihat pada gambar 3.8. Jika waktu1 sama dengan 10.



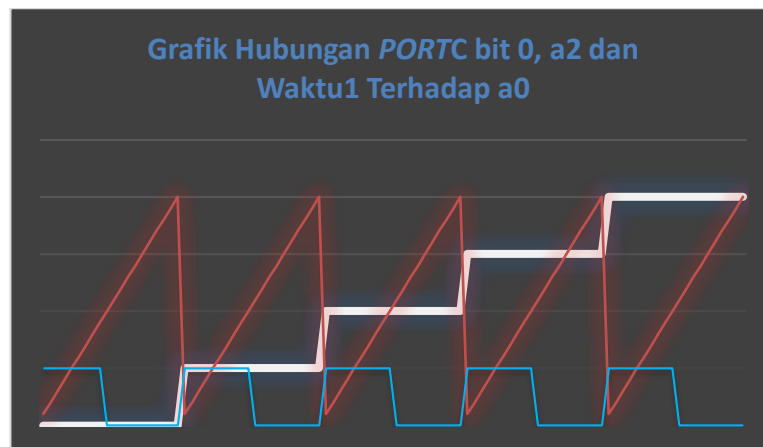
Gambar 3.8 Grafik a0 VS Waktu1

Perintah kedua adalah jika $a0$ lebih kecil dari $waktu1$ maka $PORTC$ bit 0 bernilai high (baris 2). Jika $a0$ lebih besar dari $waktu1$ maka $PORTC$ bit 0 akan bernilai low. Grafik nilai $a0$ dan $PORTC$ bit 0 dapat dilihat pada gambar 3.9.



Gambar 3.9 Grafik $a0$ dan $PORTC$ bit 0 VS Waktu1

Perintah ketiga (ditandai dengan **) jika $a0$ lebih besar dari dua kali $waktu1$ maka $a0$ akan direset menjadi nilai 0 dan variabel $a2$ ditambah 1 (baris 3). Grafik $a0$, $PORTC$ bit 0, dan $a2$ vs $waktu1$ dapat dilihat pada gambar 3.9.



Gambar 3.10 Grafik Hubungan $PORTC$ bit 0, $a2$ dan $waktu1$ Terhadap $a0$

Perintah keempat (ditandai dengan ***) jika $a2$ lebih besar dari $jumlah1$ maka variabel $a1$ akan direset menjadi nilai 0 (baris 4). Grafik hubungan $PORTC$ bit 0, $a2$ dan $waktu1$ terhadap $a0$ dapat dilihat pada gambar 3.10. Program timer motor stepper 2 dan motor stepper 3 sama dengan program motor stepper 1.

Perintah kelima adalah $check1$ dan $check2$ diberi nilai 0 (baris 5). Perintah keenam adalah Jika $a1$, $b1$ dan $c1$ sama dengan nol maka $check1$ diberikan nilai 1 (baris 6). Perintah ketujuh adalah

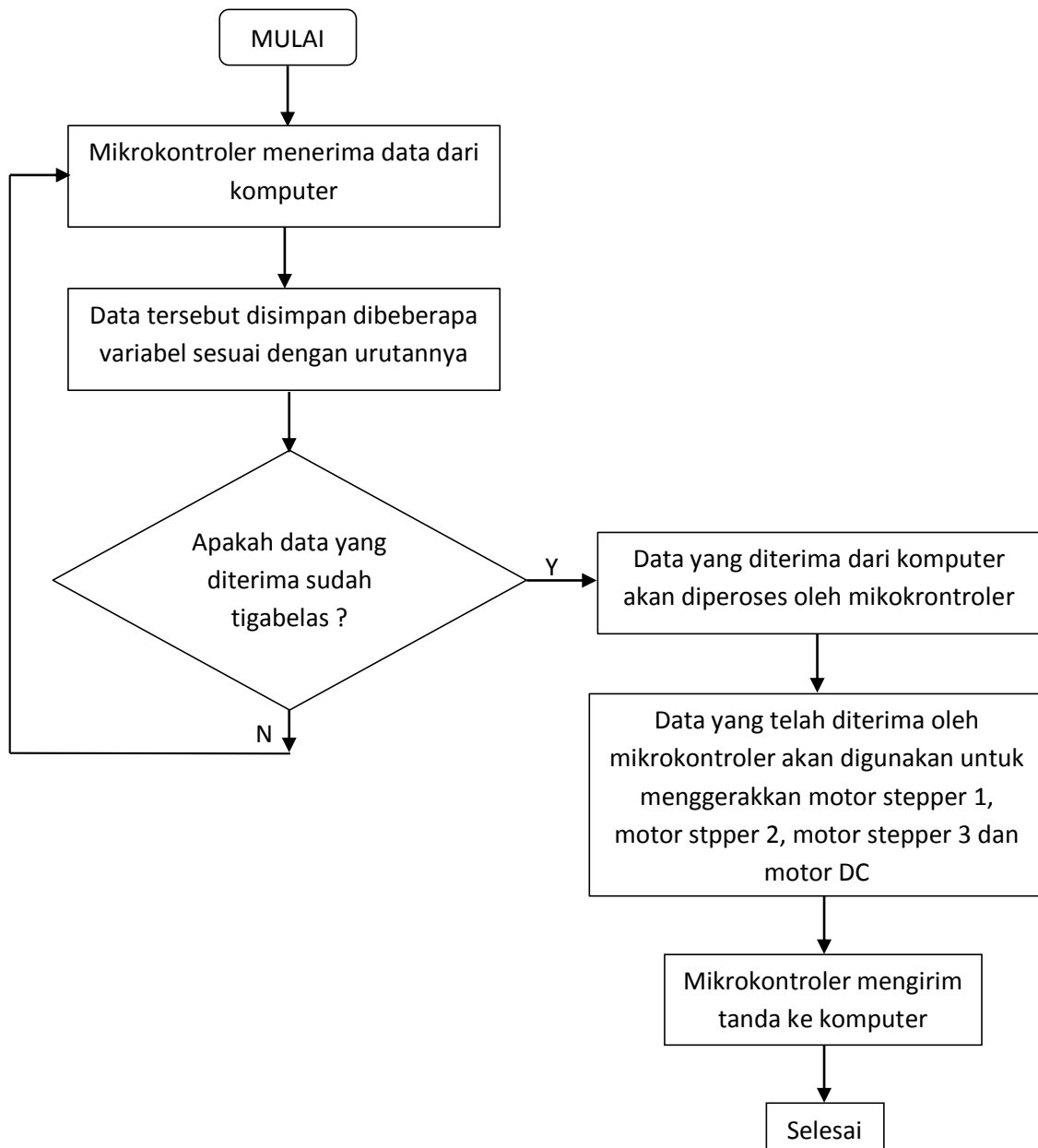
Jika a2, b2 dan c2 lebih besar dari 0 maka check2 diberikan nilai 1 (baris 7). Perintah kedelapan (ditandai dengan****) Jika check1 dan check2 sama dengan 1 maka program yang ada didalam kurung kurawal akan dikerjakan perintah tersebut yaitu a2, b2 dan c2 akan direset 0, selanjutnya adalah mikrokontroler mengirim tanda 0 (putchar) ke komputer sebagai tanda bahwa mikrokontroler siap menerima data yang baru.

Supaya fungsi *timer* dapat berfungsi, diperlukan program utama yang berisi tentang perintah untuk menerima data dari komputer, perintah untuk memanggil fungsi dan perintah untuk mengeksekusi program. Agar lebih mudah memahami cara kerja program utama, dibuat diagram alir. Adapun diagram alir program penerimaan data dari komputer ke mikrokontroler dapat dilihat pada gambar 3.11.

Pada diagram alir seperti yang ditunjukkan pada gambar 3.11, mikrokontroler dalam keadaan siap untuk menerima data dari komputer. Kemudian, mikrokontroler menerima data dari komputer, data yang akan diterima dari komputer umumnya berjumlah tigabelas data. Jika komputer menerima data angka 255 maka *PORTC* bit 0 di-set 1 yang mengakibatkan spindel aktif. Jika komputer menerima data angka 254 maka *PORTC* bit 0 di-set 0 yang mengakibatkan spindel tidak aktif. Jika komputer menerima data angka 253 maka variabel h akan direset, tujuan dari mereset data adalah supaya tidak ada penumpukan data yang telah dikerjakan sebelumnya. Jika komputer menerima data angka kurang dari 253 maka variabel h akan terus bertambah 1.

Jika variabel h sama dengan 1 maka data yang diterima mikrokontroler akan disimpan pada variabel x0, variabel x0 digunakan untuk menyimpan data tanda. Jika variabel h sama dengan dua maka data yang diterima mikrokontroler akan disimpan pada variabel x1, variabel x1 digunakan untuk menyimpan data hasil bagi. Jika variabel h sama dengan tiga maka data yang diterima mikrokontroler akan disimpan pada variabel x2, variabel x2 digunakan untuk menyimpan data sisa bagi. Jika variabel h sama dengan empat maka data yang diterima mikrokontroler akan disimpan pada variabel x3, variabel x3 digunakan untuk menyimpan data delay. Variabel x4 berisi perintah penjumlahan x1 kali 252 tambah variabel x2. Jika variabel x0 sama dengan 0 maka *PORTC* bit 1 di-set 1 jika tidak *PORTC* bit 1 di-set 0. Jumlah1 berisi variabel x4. Waktu1 berisi variabel x3.

Jika variabel h sama dengan 5 sampai dengan 8 maka data-data tersebut akan disimpan dan digunakan untuk menggerakkan motor stepper 2. Jika variabel h sama dengan 9 sampai dengan 12 maka data-data tersebut akan disimpan dan digunakan untuk menggerakkan motor stepper 3, kemudian a1, b1 dan c1 di-set 1 supaya program pada fungsi timer dikerjakan.



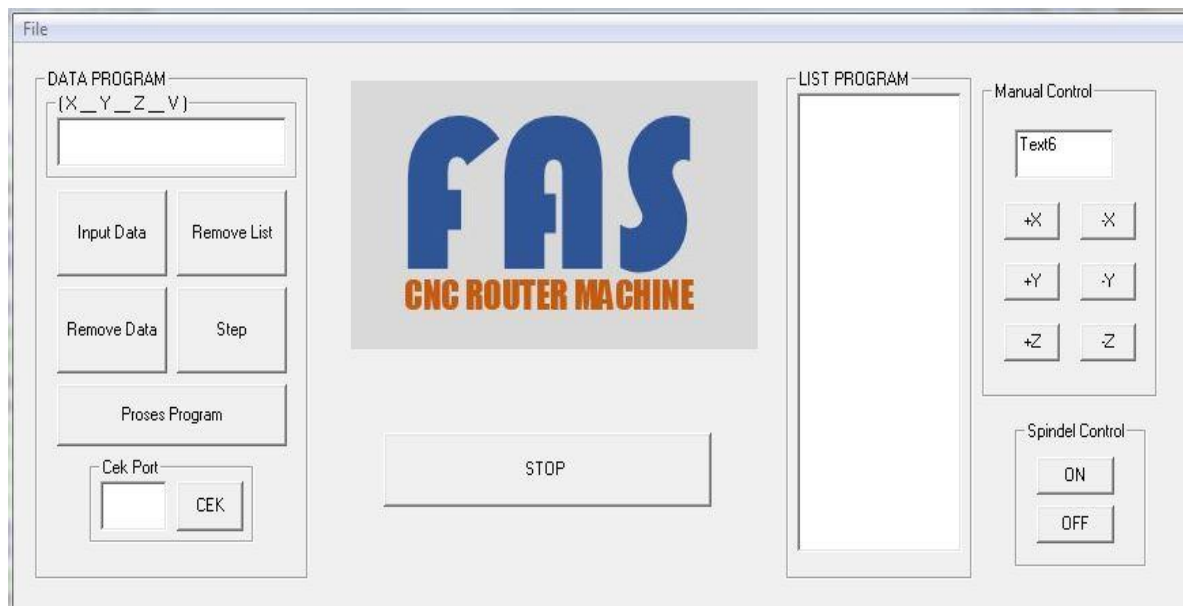
Gambar 3.11 Diagram Alir Program Utama

3.3.2 Visual basic 6.0

Perangkat lunak yang dibuat harus mampu mengirimkan sejumlah data yang telah diolah dari komputer ke mikrokontroler. Pengiriman data dilakukan dengan menggunakan komunikasi secara serial. Data-data yang dikirimkan dari komputer menuju mikrokontroler kemudian ditampilkan ke dalam komputer agar dapat diketahui apakah data yang telah diterima oleh mikrokontroler sesuai dengan data yang dikirim dari komputer. Selain dapat mengirim

sejumlah data-data dari komputer ke mikrokontroler, program juga harus dapat mengirim sejumlah data secara berulang.

Tahapan pembuatan perangkat lunak pemrograman pengendalian penggerak pahat mesin router CNC pada sumbu X, sumbu Y dan sumbu Z dimulai dengan mendesain *form*. *Form* merupakan jendela kerja pada *Visual Basic* yang berfungsi sebagai lembar kerja program. *Form* jalur komunikasi serial dibuat untuk mempermudah mengatur jalur komunikasi antara komputer dan mikrokontroler. Objek yang terdapat pada *form* terdiri dari *textbox*, *commandbutton*, dan *listbox*. Tampilan *form* yang sedang dijalankan dapat dilihat pada gambar 3.12.



Gambar 3.12 Tampilan *Form* Yang Sedang Dijalankan

Pada *form* pengendalian penggerak pahat pada sumbu X, sumbu Y dan sumbu Z terdapat 15 *commandbutton*, 3 *textbox* dan 1 *listbox*. Sebelum melakukan komunikasi serial antara mikrokontroler dan komputer, beberapa properti pada *MSComm* perlu diatur ulang. Beberapa properti pada *MSComm* yang harus diatur ulang dapat dilihat pada tabel 3.1.

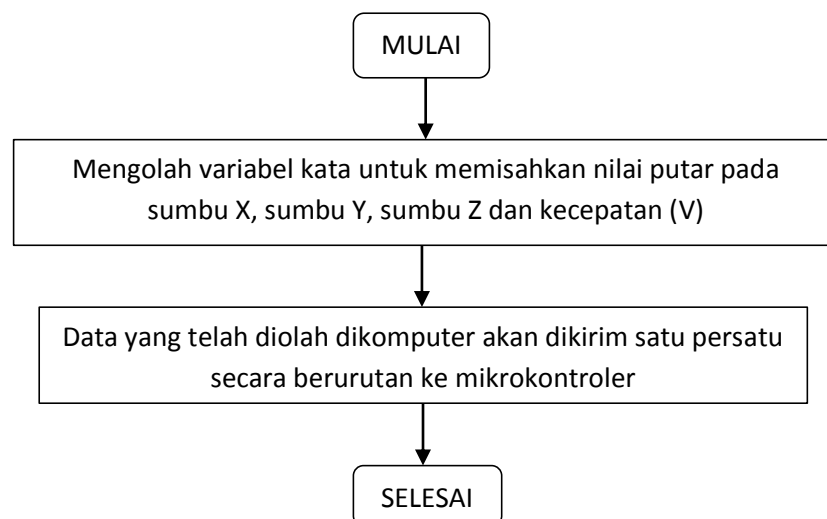
Tabel 3.2 Beberapa properti pada *MSComm* yang harus di-set

Objek	Properti	Harga
MSComm	Name	MSComm1
	Rthreshold	1
	Srthreshold	1
	Setting	9600,n,8,1

Kapasitas pengiriman data komunikasi serial pengendalian penggerak pahat mesin router CNC pada sumbu X, sumbu Y dan sumbu Z adalah 8 bit. Oleh karenanya nilai maksimum bilangan yang dapat dikirimkan melalui komunikasi serial adalah 255. Bilangan yang nilainya lebih dari 255, harus dikirimkan dengan cara tertentu.

Angka yang dikirimkan dibagi menjadi 4: 255, 254, 253, dan <253. Jika nilai angka berjumlah tiga karakter kemudian jika nilai kata1 bernilai 255 maka komputer akan mengirim angka 255 yang mengakibatkan spindel aktif. Jika nilai kata1 bernilai 254 maka komputer akan mengirim angka 254 yang mengakibatkan spindel tidak aktif. Jika jumlah karakter lebih dari enam maka komputer akan mengirimkan sejumlah data untuk menggerakkan motor stepper sumbu X, motor stepper sumbu Y dan motor stepper sumbu Z akan bergerak.

Agar lebih mudah dalam memahami cara kerja program pengiriman data dari komputer ke mikrokontroler dibuat diagram alir seperti pada gambar 3.10. Detail List program pengiriman data secara serial yang sesuai dengan diagram alir 3.10 dapat dilihat pada lampiran.



Gambar 3.13 Digram Alir Porgram Pengiriman Data Dari Komputer