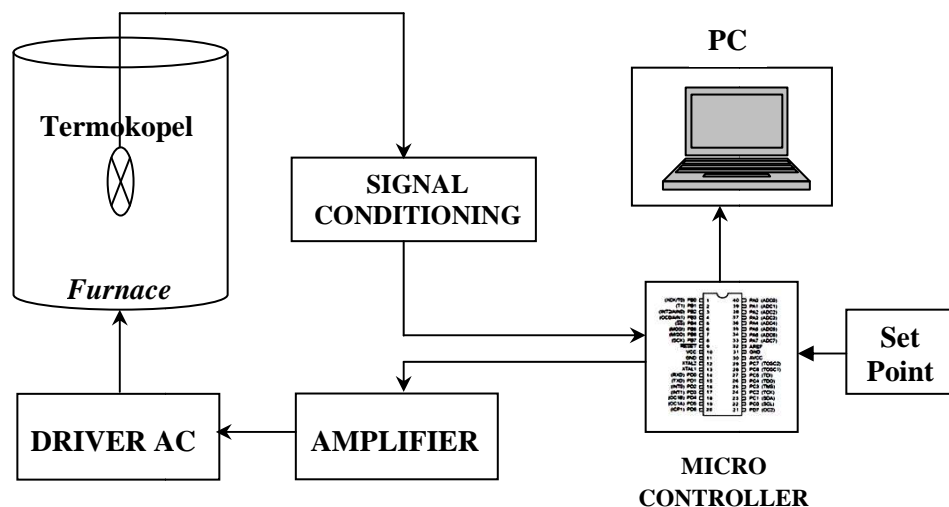


BAB III

PERANCANGAN SISTEM

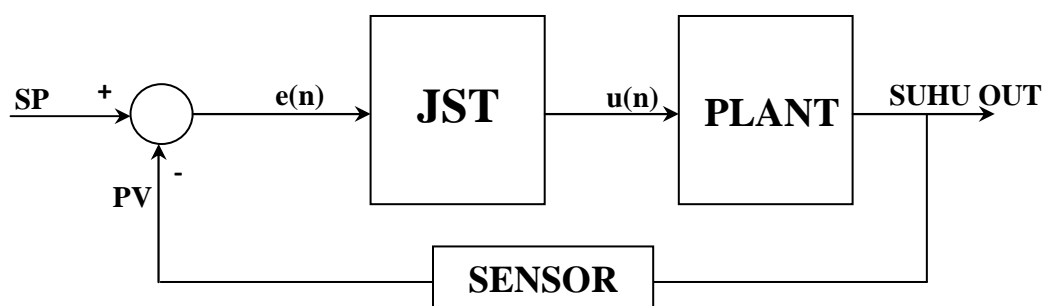
Perancangan sistem pengendalian *furnace* untuk plant pyrolisis meliputi perancangan perangkat keras (hardware) dan perancangan perangkat lunak (*software*). Sistem yang akan dirancang akan membentuk suatu pengendalian suhu pada *furnace*. Pengendalian suhu *furnace* dilakukan dengan cara mengatur tegangan kontrol yang akan diberikan ke *furnace*, dimana pengaturan tersebut diatur oleh driver AC berdasarkan output kontrol JST (Jaringan Syaraf Tiruan). Berikut adalah gambar diagram blok dari sistem pengendalian suhu pada *furnace*.



Gambar 3.1 Blok Diagram Sistem Kendali Suhu *Furnace*

Furnace merupakan sebuah tungku pemanas yang dapat menghasilkan panas sampai 1000°C. Bentuknya menyerupai sebuah tabung silinder yang dilapisi oleh material yang dapat menahan panas dan lapisan terluar dilapisi oleh

aluminium. Sensor yang digunakan adalah sebuah sensor termokopel seri-K yang dapat mengukur suhu dari rentang $-270^{\circ} - 1.370^{\circ}\text{C}$. Termokopel dilapisi dengan bahan logam yang dapat menahan suhu tinggi. Setiap kenaikan 1°C akan menghasilkan tegangan output sebesar $0,04\text{ mV}$ dan output sensor yang sangat tidak linier karna output yang dihasilkan tidak stabil. Tegangan output yang dihasilkan sangat kecil dalam mV dan tidak linier, sehingga perlu ditambahkan rangkaian pengkondisi sinyal yang dapat mengurangi efek tidak stabil yang dihasilkan oleh output termokopel. Di dalam rangkaian pengkondisi sinyal sensor termokopel, terdapat sebuah rangkaian filter dan rangkaian penguat. Fungsi dari rangkaian filter adalah untuk mengurangi efek tidak stabil pada output termokopel, setelah itu output yang telah difilter akan diberi penguatan agar dapat dibaca oleh mikrokontroler sebagai data masukan. Kemudian data tersebut akan diolah mikrokontroler lebih lanjut. Sistem kendali suhu *furnace* juga dilengkapi sebuah *software* akuisisi data yang berfungsi untuk merekam data yang dikirim oleh mikrokontroler untuk keperluan pengolahan data. Berikut adalah sistem kendali suhu berbasis Jaringan Syaraf Tiruan.



Gambar 3.2 Blok Diagram Sistem Pengendali Jaringan Syaraf Tiruan

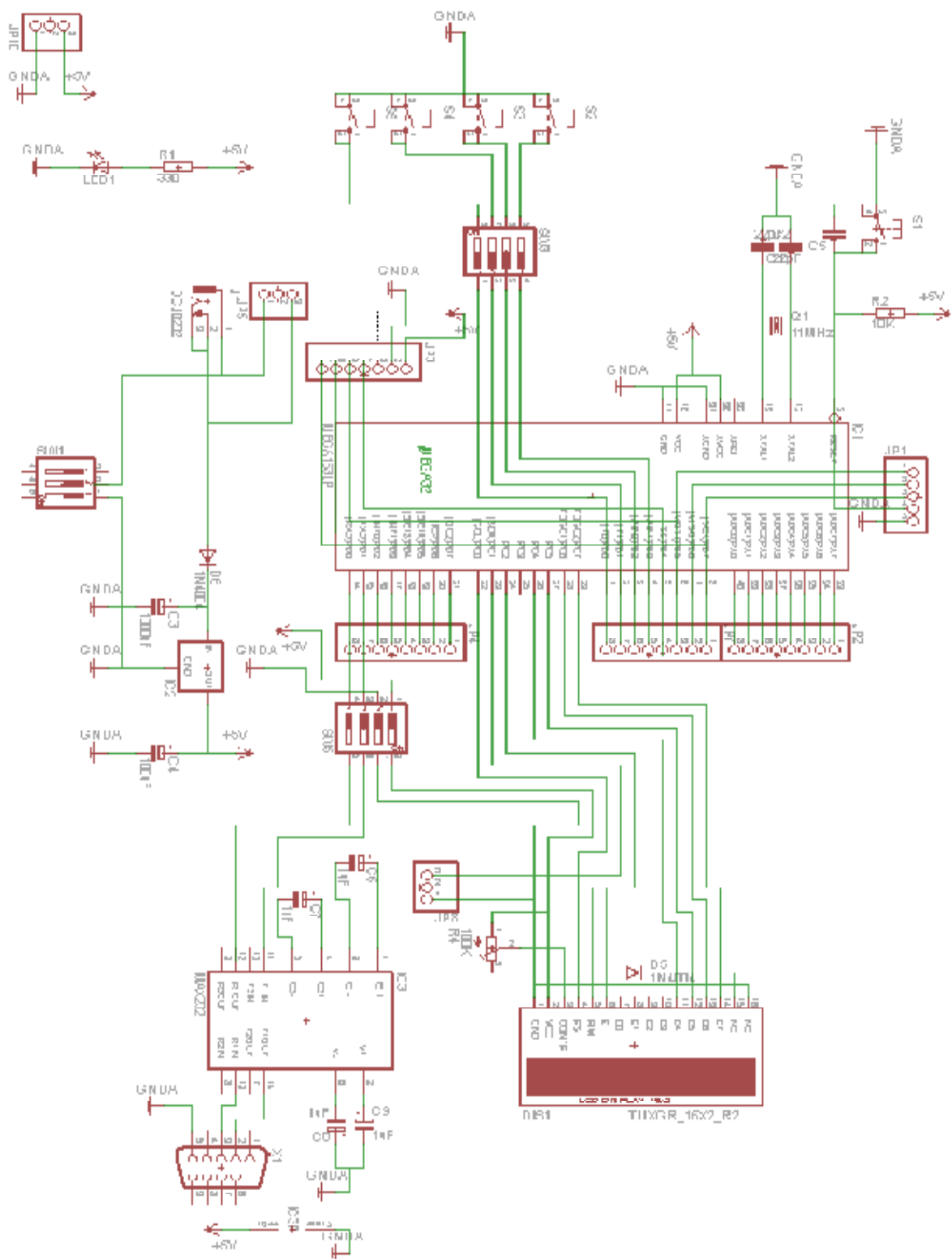
Ketika SP (Set Point) diinputkan ke dalam sistem, maka sistem akan menghitung $e(n)$ atau *error*. *Error* merupakan input untuk kontrol JST yang akan mengontrol sistem. Output atau $u(n)$ dari kontrol JST berupa tegangan kontrol yang mengatur unjuk kerja plant. Tegangan kontrol yang dihasilkan digunakan untuk memberi supply tegangan 0-220 VAC ke plant. Output dari plant yang berupa suhu akan diumpanbalikkan untuk menghitung sinyal *error* pada sistem.

3.1 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras (*hardware*) pada sistem pengendalian suhu *furnace* meliputi rangkaian sistem minimum, pengkondisi sinyal sensor termokopel, penguat tegangan control, pengontrol sudut fasa tegangan AC, dan saklar elektronik. Semua rangkaian yang telah didesain dengan menggunakan *software* Eagle.

3.1.1 Rangkaian Sistem Minimum

Sistem minimum merupakan rangkaian pengendali utama yang akan mengendalikan plant *furnace* dengan menanamkan program kontrol JST di dalamnya. Sistem minimum yang dirancang menggunakan mikrokontroler dengan tipe AVR ATmega 32. Sistem minimum juga telah terintegrasi dengan rangkaian LCD dan komunikasi serial. LCD berfungsi sebagai interface untuk menampilkan berbagai menu yang telah deprogram. Begitupun dengan komunikasi serial berfungsi sebagai jalur data yang dikirim dari pengendali untuk keperluan pengolahan data.



Gambar 3.3 Rangkaian Sistem Minimum AVR ATmega 32

Port I/O yang digunakan meliputi port ADC (Analog Digital Converter) sebagai port untuk membaca input sensor termokopel. Selain itu penggunaan port input pada sistem minimum digunakan untuk masukan keypad dan push button sebagai tombol dan masukan nilai. Port PWM pada sistem minimum digunakan untuk mengontrol *furnace* melalui driver pengontrol sudut fasa tegangan AC sebagai supply tegangan kontrol ke *furnace*.

3.1.2 Rangkaian Pengkondisi Sinyal Sensor Termokopel

Sensor termokopel seri-K memiliki output tegangan yang sangat kecil, dimana setiap kenaikan satu derajat akan menghasilkan tegangan sebesar 0,04 mV. Sensor termokopel juga bersifat tidak linier atau tidak stabil dikarenakan output yang dihasilkan oleh sensor sangat sensitif sehingga berubah-ubah. Untuk itu pada rangkaian pengkondisi sinyal sensor termokopel terdapat rangkaian *low pass filter* analog dan rangkaian penguat tegangan. Rangkaian *low pass filter* berfungsi untuk menghilangkan noise atau meredam output dari sensor yang tidak stabil atau berubah-ubah menjadi lebih stabil. Sedangkan rangkaian penguat (Op-Amp) berfungsi untuk menguatkan nilai output dari sensor agar dapat dibaca oleh mikrokontroler sehingga menjadi lebih mudah dalam proses kalibrasi sensor.

Perhitungan untuk mendapat nilai R_4 pada rangkaian Op-Amp adalah :

Diketahui bahwa nilai $R_5 = 1,298 \text{ MOhm}$

$$V_{out} = \left[\frac{R_5}{R_4} + 1 \right] \times V_{in}$$

$$4950\text{mV} = \left[\frac{1298}{R_4} + 1 \right] 41,665\text{mV}$$

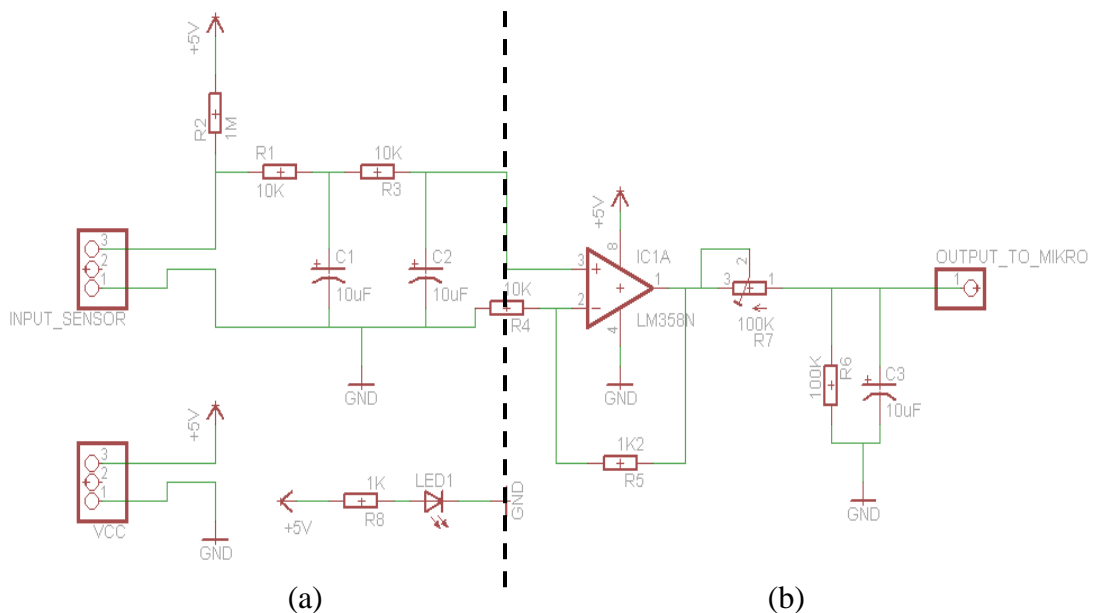
$$R_1 = \frac{1298}{119,52 - 1} = 10,95 \text{ K}\Omega$$

Perhitungan penguatan rangkaian Op-Amp adalah :

$$A_v = \left[\frac{R_5}{R_4} \right] + 1$$

$$A_v = \left[\frac{1298}{10,95} \right] + 1$$

$$A_v = 119,5x$$



Gambar 3.4 (a) Rangkaian *Low pass filter* Analog dan (b) Rangkaian Op-Amp

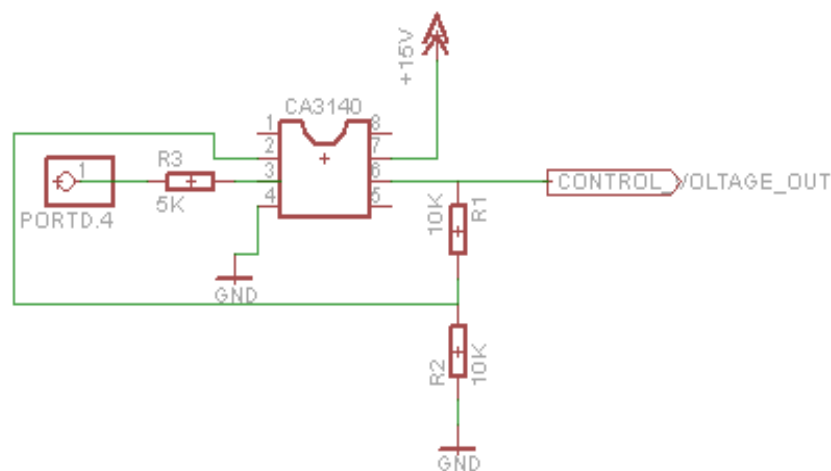
3.1.3 Rangkaian Penguat Tegangan Kontrol

Rangkaian pengontrolan sudut fasa tegangan AC memerlukan tegangan kontrol sebesar 0 sampai 10 V_{DC} untuk menggeser pulsa 0° sampai 180° . Kemampuan dari mikrokontroler ternyata hanya mampu menghasilkan tegangan output maksimal sebesar 0 sampai 5 V_{DC} , sehingga perlu dilakukan penguatan sebanyak 2 kali agar dapat menghasilkan tegangan kontrol hingga 10 V_{DC} . Pada

rangkaian ini digunakan Op-Amp CA3140 yang digunakan sebagai penguat non inverting.

Perhitungan nilai penguatan rangkaian penguat tegangan kontrol adalah :

$$\begin{aligned}
 A &= V_o/V_i \\
 &= (1+ R_{10}/R_{11}) \\
 &= (1+ 10K/10K) = \mathbf{2x}
 \end{aligned}$$

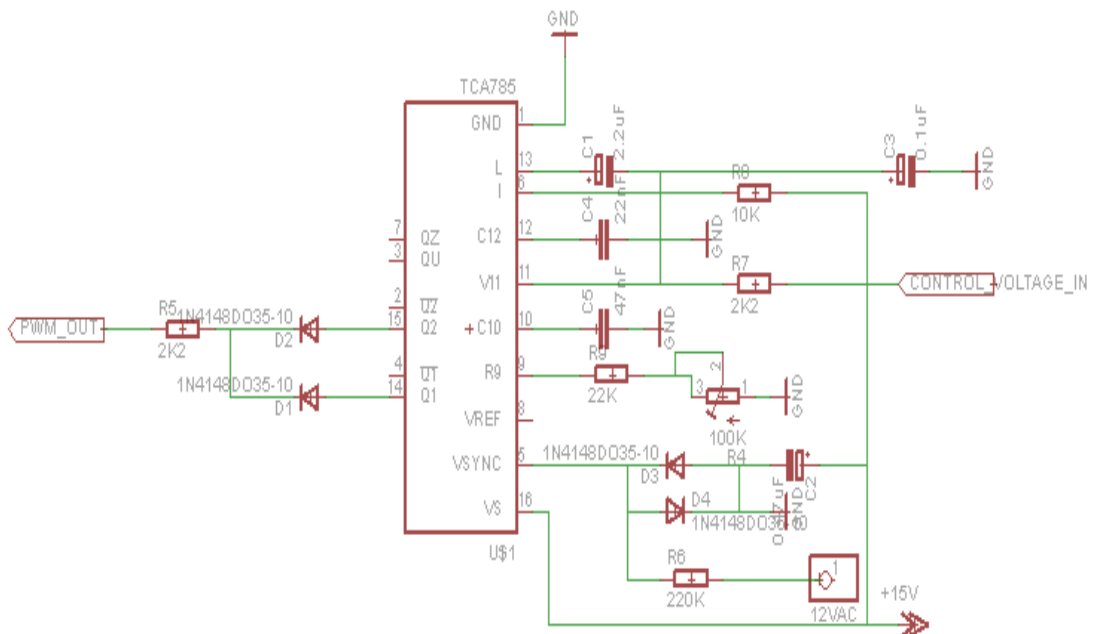


Gambar 3.5 Rangkaian Penguat Tegangan Kontrol

3.1.4 Rangkaian Pengontrol Sudut Fasa Tegangan AC

Pada rangkaian pengontrol sudut fasa tegangan AC pada pengendalian *furnace* menggunakan IC TCA785 sebagai komponen utama. IC TCA 785 adalah komponen pengendali tegangan AC dengan cara menggeser sudut fasa 0 – 180°. Komponen ini telah terintegrasi dengan rangkaian Zero Crossing Detector dan

penggeser sudut fasa dari 0° sampai 180° . Berikut adalah rangkaian pengontrol sudut fasa tegangan AC pada pengendalian *furnace*.



Gambar 3.6 Rangkaian Pengontrol Sudut Fasa Tegangan AC

Sinyal sinkronisasi berasal dari tegangan 12 V_{AC} yang dihubungkan pada kaki nomor 5. Untuk menghindari arus yang sangat besar pada IC maka sebelum masuk kaki 5, tegangan 12 V_{AC} diberikan hambatan terlebih dahulu sebesar $220\text{ k}\Omega$ dan juga diberi rangkaian dioda antiparalel antara kaki 1 dan 5. Arus yang masuk ke kaki 5 sangat kecil yaitu :

$$I_{SYNC} = \frac{V_{SYNC}}{R}$$

$$I_{SYNC} = \frac{12}{220000} = 5,4\ \mu A$$

Dimana :

V_{sync} = Tegangan sinkronisasi (Volt)

I_{sync} = Arus sinkronisasi(Ampere)

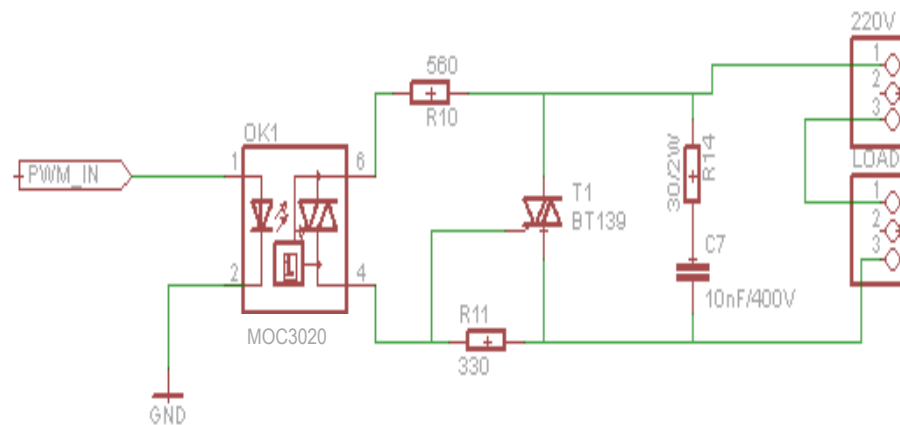
R = Hambatan (Ohm)

Pada IC TCA 785 terdapat peraba nol (ZCD) yang akan menentukan letak titik nol sinyal AC dan disimpan ke dalam memori sinkron. Detektor ini kemudian akan mengendalikan generator gelombang tegangan gigi gergaji yang sesuai dengan frekuensi sumber tegangan. Kapasitor C_2 , Resistor R_9 , dan R_{10} akan menentukan kemiringan dari gelombang gigi gergaji yang dihasilkan. Gelombang ini kemudian dibandingkan dengan tegangan referensi V_{11} oleh komparator (pemanding). Sinyal output dari komparator ini kemudian akan diteruskan ke rangkaian logika. Bila tegangan referensi pada kaki 11 (V_{11}) pada posisi terendah, maka sudut penyulutan akan menunjukkan $\alpha = 0^\circ$. Sehingga untuk mengatur pemicuan dapat dilakukan dengan cara memberikan variasi tegangan masuk pada kaki 11 (tegangan kontrol).

Keluaran dari IC TCA 785 adalah berupa pulsa pada kaki 14 dan 15 yang digabungkan dengan 2 buah dioda untuk menghindari adanya tegangan balik ke IC tersebut. Kaki 14 akan membangkitkan pulsa ketika sinyal AC berada pada daerah negatif ($180^\circ - 360^\circ$), sedangkan kaki 15 akan membangkitkan pulsa saat sinyal AC berada pada daerah positif ($0^\circ - 180^\circ$). Sinyal pulsa yang dihasilkan akan mempunyai fase yang sama dengan sinyal tegangan AC, dengan lebar yang dapat divariasikan sesuai dengan besar tegangan kontrolnya.

3.1.5 Rangkaian Saklar Elektronik

Rangkaian saklar elektronik berfungsi sebagai penyalur daya keluaran atau arus keluaran ke beban dan dapat dikendalikan untuk menghubungkan atau memutusinya. Pada rangkaian saklar elektronik menggunakan TRIAC BT139 sebagai komponen utama. TRIAC BT139 adalah komponen yang dapat mengatur jalur tegangan AC yang lewat dengan *gate* sebagai pemicunya. Untuk menghindari *short* atau lonjakan tegangan yang dapat merusak komponen, maka sebelum masuk ke rangkaian utama diberikan rangkaian MOC3020 sebagai pengaman. Berikut adalah rangkaian saklar elektronik dalam pengendalian *furnace*.



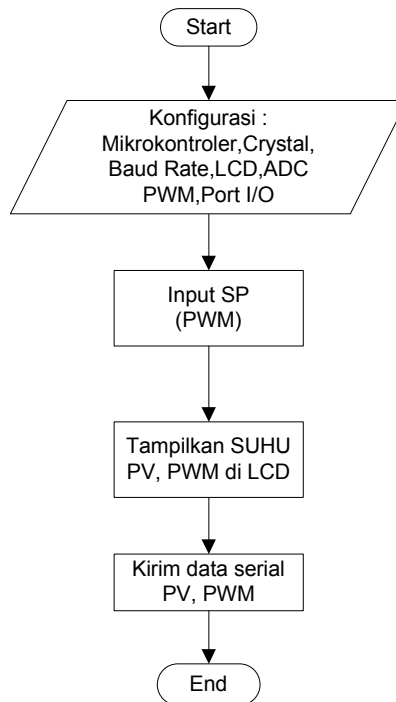
Gambar 3.7 Rangkaian Saklar Elektronik

3.2 Perancangan Perangkat Lunak (*Software*)

Pada perancangan lunak (*Software*) pada pengendalian suhu *furnace* beberapa diantaranya yang akan dibahas meliputi perancangan program pengambilan data pelatihan JST (Jaringan Syaraf Tiruan), Pelatihan data JST, pengujian data JST, dan kontrol JST. Perancangan perangkat lunak pada sistem pengendalian ini, menggunakan *software* BASCOM AVR 5.11 untuk proses pengambilan data dan pemrograman kontrol utama JST, sedangkan untuk proses pelatihan data dan pengujian data JST menggunakan *software* MATLAB 2009.

3.2.1 Program Pengambilan Data Pelatihan JST

Pada perancangan program pengambilan data pelatihan JST (Jaringan Syaraf Tiruan) dengan cara memberikan variasi tegangan kontrol dari mikrokontroler berupa nilai PWM (Pulse Width Modulation) selama selang waktu tertentu, sampai terlihat perubahan suhu yang terjadi pada *furnace*. Nilai suhu PV (Present Value) pada *furnace* diprogram dan ditampilkan di LCD. Nilai PWM merupakan set point yang dimasukkan ke dalam kontroler sebagai supply tegangan ke *furnace*, sedangkan nilai PV menunjukkan suhu faktual yang terukur pada *furnace* sepanjang pemberian tegangan kontrol PWM. Sistem kontrol pada proses pengambilan data JST ini merupakan sistem kontrol *open loop*, dikarenakan tidak ada proses umpan balik dari suhu output ke kontroler untuk memperoleh nilai *error*. Berikut adalah alur program yang menunjukkan proses pengambilan data JST.



Gambar 3.8 *Flowchart* Program Pengambilan Data JST

Data input SP (Set Point) dan suhu PV *furnace* dikirim melalui port serial dan direkam oleh *software* akuisisi data yang dibuat dengan software Visual Basic 6. Data yang direkam oleh *software* akuisisi data kemudian dikonversi ke microsoft excel sebagai data mentah untuk diolah lebih lanjut untuk proses selanjutnya. Gambar 3.9 menunjukkan potongan listing program utama pengambilan data JST. Program dibuat dengan menggunakan *software* BASCOM AVR dengan dasar pemrograman menggunakan bahasa *basic*.

```

Start ADC
Cursor Off
Cls
Do
A = Getadc(0)
B = A * 1.1194
C = B - 75.4894
F = Fusing(c , "#.##")
E = C

Potensio = Getadc(7)
Shift Potensio , Right , 2
Pwmla = Potensio
J = Pwmla

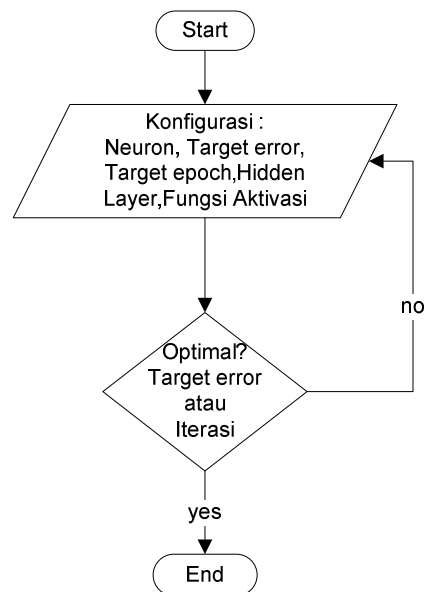
Cls
Locate 1 , 2
Lcd "Temp=" ; E
Locate 2 , 1
Lcd J
Print "S " ; F
Print "T " ; Pwmla
Waitms 100
Loop

```

Gambar 3.9 Potongan *Listing* Program Utama Pengambilan Data JST

3.2.2 Program Pelatihan Data JST

Program pelatihan data JST dibuat dengan menggunakan *software* MATLAB 2009. Program dibuat berdasarkan desain arsitektur JST terbaik yang telah diperoleh. Arsitektur JST yang akan dibuat dengan susunan1 *hidden layer* sebanyak 20 *neuron* dan fungsi aktivasi menggunakan fungsi *logsig*. Berikut adalah alur program pelatihan data JST.



Gambar 3.10 *Flowchart* Program Pelatihan Data JST

Pemrograman pelatihan data JST pada MATLAB 2009, harus memperhatikan parameter – parameter penting yang terkait dengan hasil performa pelatihan. Parameter tersebut diantaranya adalah jumlah hidden layer, jumlah neuron di setiap *hidden layer*, fungsi aktivasi, target *error*, target *epoch* atau iterasi, dan *learning rate*. Berikut adalah daftar komposisi parameter dari arsitektur terbaik pada sistem pengendalian suhu *furnace*.

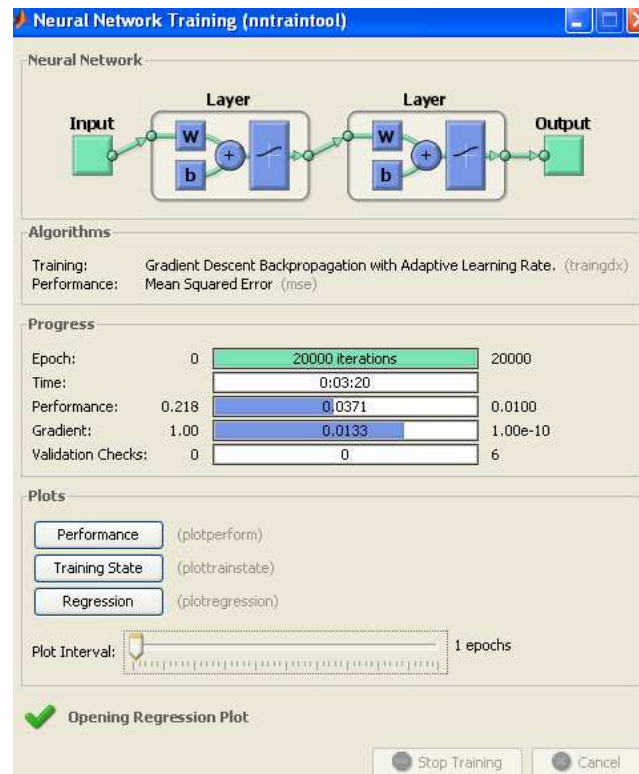
Tabel 3.1 Parameter Arsitektur JST

Hidden Layer	1
Neuron	20
Fungsi Aktivasi	logsig
Error	0,01
Epoch	20
Learning Rate	0,09

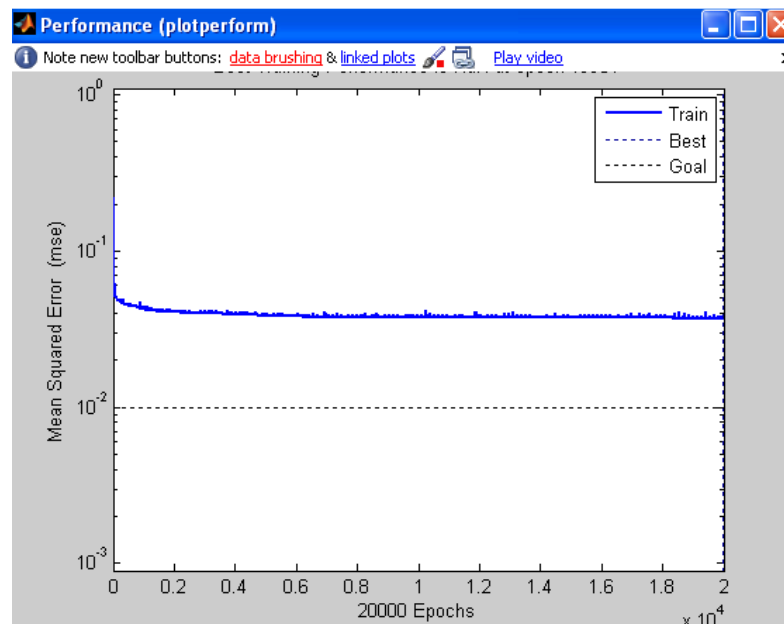
Setelah mendapatkan komposisi diatas, tahap berikutnya adalah implementasi nilai parameter diatas ke dalam *script* program. Script berikut ini adalah listing program utama pelatihan data JST menggunakan *script editor* pada MATLAB 2009 :

```
clear;
clc;
load datalatihku3
inp=datlatihku3(:,1)';
tar=datlatihku3(:,2)';
[inpn,inpns]=mapminmax(inp,0,1);
[tarn,tarns]=mapminmax(tar,0,1);
net=newff(minmax(inpn),[10,1],{'logsig','logsig'},'traingdx');
net.trainparam.epochs=20000;
net.trainparam.goal=0.01;
net.trainparam.lr=0.9;
net=train(net,inpn,tarn);
```

Hasil dari pelatihan menunjukkan target *error* belum tercapai, dan hanya mampu mencapai sebesar 0,0372 dengan iterasi sebanyak 20.000 iterasi. Berikut adalah grafik performa pelatihan yang telah tercapai berdasarkan arsitektur yang telah didapatkan.



Gambar 3.11 Tampilan Performa Pelatihan Data JST



Gambar 3.12 Grafik Performa Pelatihan JST Berdasarkan Target Error

Hasil akhir dari proses pelatihan data JST adalah nilai bobot dan bias yang didapatkan dari arsitektur JST yang ditentukan. Nilai bobot dan bias inilah yang akan menjadi parameter penting pada kontrol JST nantinya. Berikut adalah hasil nilai bobot dan bias yang didapatkan :

```
>> net.IW{1}
```

```
ans =
```

```
-110.9141  
111.3384  
-111.3710  
-110.5104  
113.5221  
112.5293  
-110.6973  
111.5415  
-109.4089  
111.7808  
111.4236  
112.6477  
110.7990  
112.4203  
-111.9490  
-111.9981  
112.0000  
-112.0000  
-112.0000  
112.0000
```

Nilai diatas merupakan nilai bobot pada sisi input, dimana terdapat sebanyak 20 bobot sesuai dengan arsitektur yang telah dilatih.

```
>> net.b{1}
```

```
ans =
```

```
113.0859  
-106.8399  
101.1730  
96.0446  
-86.6486  
-83.7800
```

```

79.6296
-73.3227
70.4101
-59.6486
-54.5841
-47.2885
-44.9505
-34.9031
29.6580
23.5859
-17.6842
11.7895
5.8959
0.0822

```

Nilai diatas adalah nilai bias yang berada pada sisi input, dimana terdapat sebanyak 20 nilai yang diperoleh.

```
>> net.LW{2,1}
```

```
ans =
```

```
Columns 1 through 9
```

```
1.7619 6.6201 5.2459 -3.6330 3.6323 -11.2315 -11.2173 -9.2599 -4.4931
```

```
Columns 10 through 18
```

```
0.6679 0.1900 3.7499 -4.5873 7.3047 -0.8856 -1.6945 1.0740 0.0931
```

```
Columns 19 through 20
```

```
-1.5613 0.4405
```

Nilai di atas adalah nilai bobot yang berada pada sisi output, dimana terdapat 20 nilai bobot yang diperoleh.

```
>> net.b{2}
```

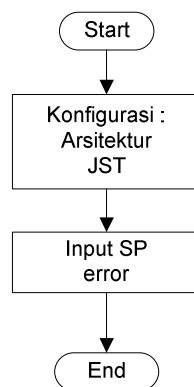
```
ans =
```

```
3.4955
```

Nilai $net.b\{2\}$ adalah nilai bias yang berada pada sisi output, dimana terdapat hanya ada satu nilai bias untuk output layer.

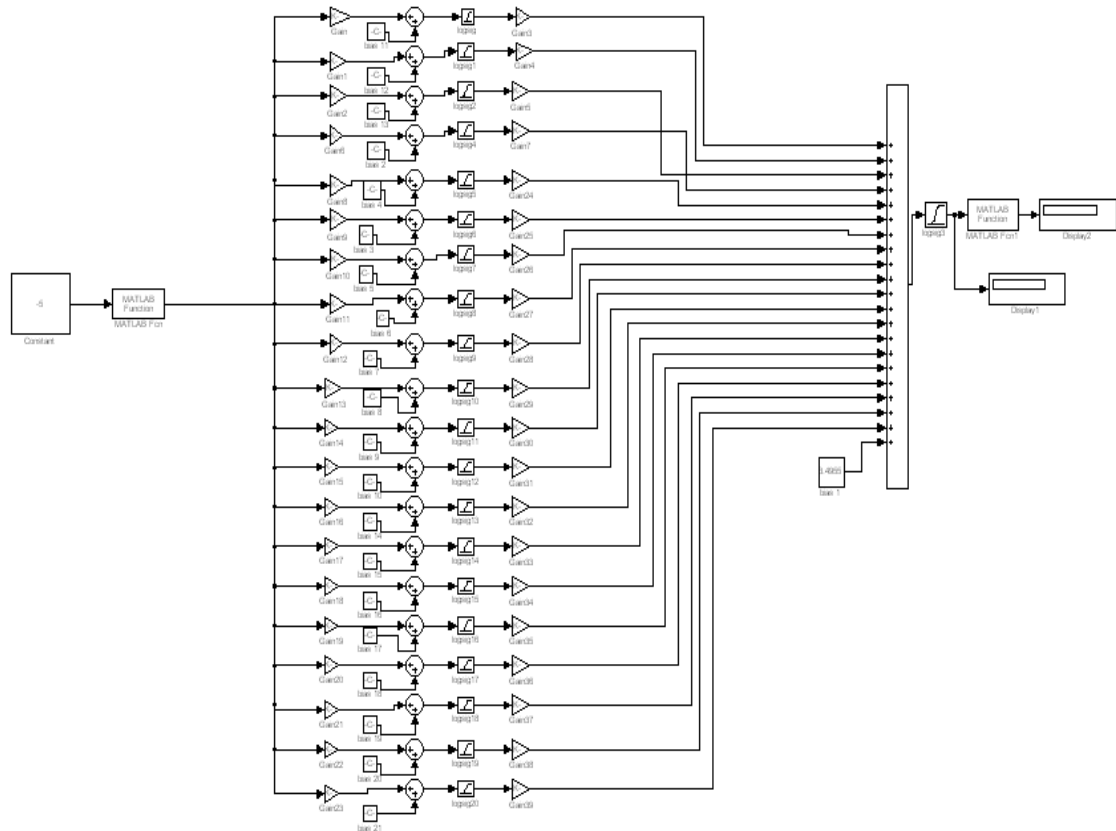
3.2.3 Program Pengujian Data JST

Program pengujian data JST menggunakan *software* MATLAB 2009 dengan memanfaatkan fasilitas *simulink* untuk proses simulasi sebelum diimplementasikan ke *hardware*. Hal ini bertujuan untuk mengetahui apakah sistem kontrol JST sudah menunjukkan unjuk kerja yang baik untuk diterapkan ke *plant*. Berikut adalah alur program pengujian data JST menggunakan *simulink* MATLAB 2009.



Gambar 3.13 *Flowchart* Program Pengujian Data JST

Metode pengujian yang akan dilakukan juga sangat sederhana. Setelah mendapatkan arsitektur terbaik berdasarkan proses pelatihan data JST, langkah selanjutnya adalah mendesain arsitektur tersebut ke dalam blok - blok simulasi pada *simulink* MATLAB. Gambar 3.12 menunjukkan implementasi arsitektur terbaik yang telah didapat ke dalam bentuk blok simulasi pada *simulink* MATLAB 2009.

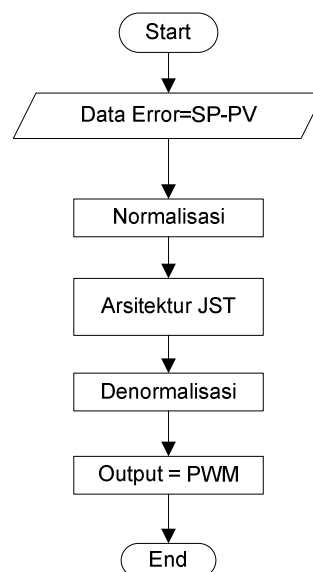


Gambar 3.14 Arsitektur JST Menggunakan *Simulink* MATLAB 2009

Pengujian kemudian dilakukan dengan memberikan *set point error* berdasarkan data pelatihan yang diperoleh. Kemudian simulasi dijalankan dan akan menghasilkan nilai output berupa tegangan kontrol yang akan mengatur kerja *plant* nantinya. Apabila antara *set point* dengan output tegangan kontrol menunjukkan hubungan yang optimal, maka arsitektur tersebut siap di implementasikan pada program kontrol utama JST.

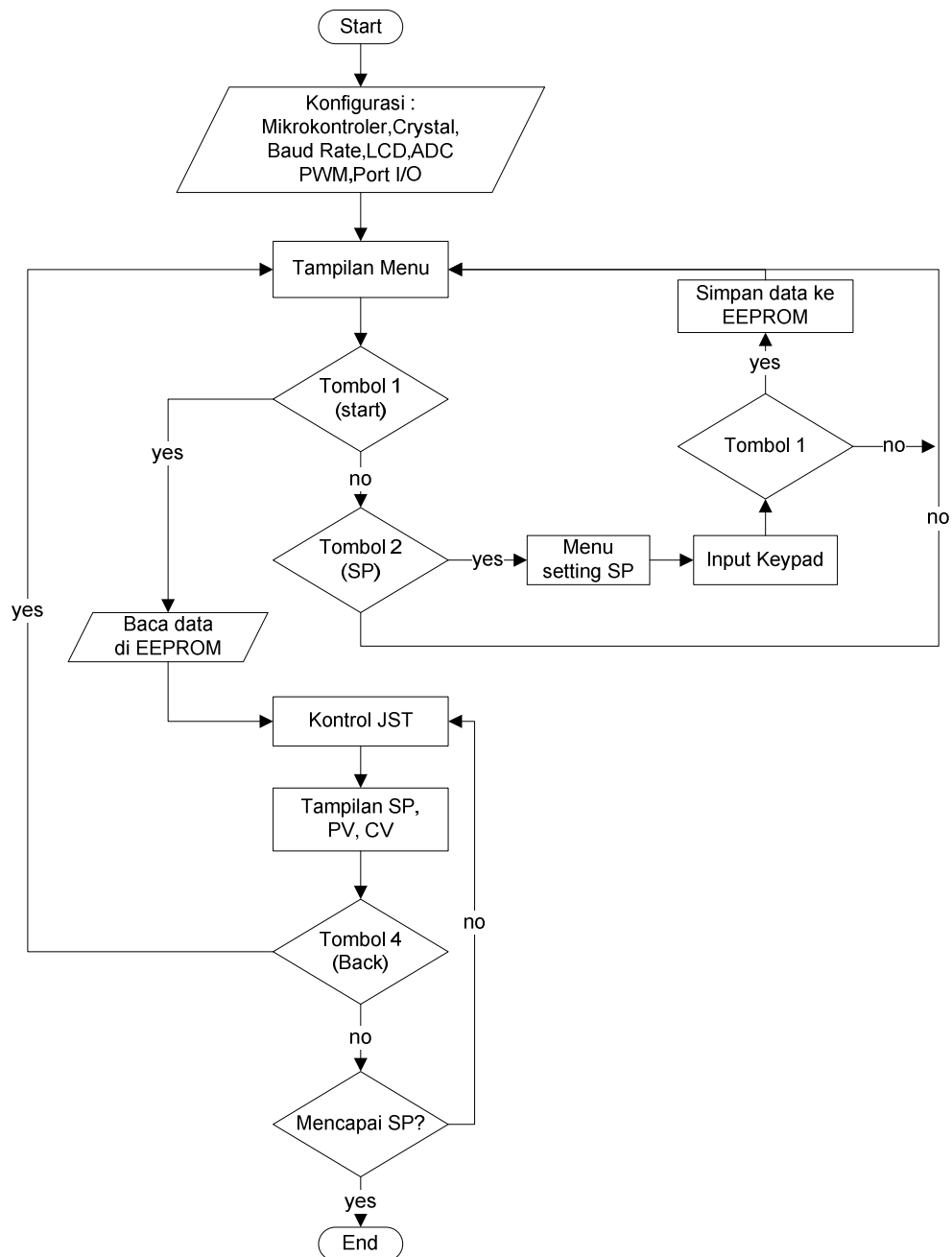
3.2.4 Program Utama Kontrol JST

Program utama kontrol JST diimplementasikan ketika telah mendapatkan arsitektur JST yang paling optimal. Implementasi dilakukan dengan cara memasukkan nilai bobot dan bias yang telah didapatkan ke dalam program utama JST. Program dimulai dengan nilai *error* yang didapat dari selisih dari nilai *set point* dengan nilai *present value* dari suhu *plant*. Kemudian data tersebut diolah lebih lanjut dengan proses normalisasi terlebih dahulu. Hasil normalisasi kemudian diproses ke dalam arsitektur JST yang telah dibuat. Proses optimasi JST mengacu pada nilai bobot dan bias yang telah didapatkan pada proses pelatihan sebelumnya. Hasil proses JST, kemudian menghasilkan nilai output yang akan dinormalisasikan selanjutnya. Hasil denormalisasi, output JST berupa nilai PWM yang akan memberikan tegangan kontrol ke *plant furnace*. Berikut alur program kontrol JST dijelaskan berikut ini.



Gambar 3.15 Flowchart Program Kontrol JST

Program kontrol JST pada gambar 3.15, kemudian diimplementasikan dengan program sistem secara keseluruhan, dimana meliputi pemrograman yang merujuk pada kesatuan sistem kendali JST.



Gambar 3.16 Flowchart Program Kontroler Hardware

Gambar 3.16 menunjukkan *flowchart* program keseluruhan kerja sistem kontrol JST pada mikrokontroler. Program berisi menu *set point* dimana merupakan langkah awal sebelum sistem kontrol berjalan. Setelah *set point* diatur, hasil pengaturan kemudian disimpan ke EEPROM. Ketika tombol *start* ditekan maka sistem kontrol JST akan melakukan komputasi untuk membangun suatu optimasi JST sesuai dengan penjelasan mengenai *flowchart* program kontrol utaman JST. Sistem terus menerus berlangsung sampai mencapai nilai *set point* yang telah diatur sebelumnya. Apabila sistem telah mencapai nilai *set point*, maka sistem akan bertahan pada kondisi nilai tersebut untuk mendapatkan performa kerja sistem yang baik.