

IMPLEMENTASI IEC 61131-3 PADA OTOMATISASI PEMBUATAN TIRAMISU

Trias Andromeda^{*}), Iwan Setiawan^{*}), Riva Amrulloh^{**})

^{*)} Staf Pengajar Teknik Elektro Fakultas Teknik Universitas Diponegoro

^{**}) Mahasiswa Teknik Elektro Fakultas Teknik Universitas Diponegoro

Jl. Prof. Sudarto S.H., Tembalang, Semarang

Abstrak

Penelitian ini dilakukan untuk mempelajari dan mengimplementasikan IEC 61131-3 pada otomatisasi pembuatan tiramisu. Selama sepuluh hingga lima belas tahun terakhir, berbagai bahasa pemrograman yang berbeda telah digunakan untuk memprogram *Programmable Logic Controller (PLC)*. Untuk satu bahasa pemrograman saja misalnya *Ladder Diagram*, tiap-tiap jenis PLC mempunyai aturan dan cara pemrograman yang berbeda-beda. Padahal industri modern biasanya tidak hanya menggunakan satu jenis PLC saja melainkan berbagai jenis PLC, hal ini akan menjadi tidak efisien, baik dari segi waktu maupun uang.

IEC 61131-3 merupakan bagian ketiga dari IEC 61131 yang membahas mengenai standar pemrograman PLC. Pada bagian bahasa pemrograman, IEC 61131-3 ini mengakomodasi 5 bahasa pemrograman PLC berbeda, yang kelimanya bisa digabung/digunakan secara bersamaan ataupun terpisah tergantung proses yang akan dikendalikan. Sedangkan pada bagian elemen umum terdapat *Program Organization Units (POUs)* yang terdiri dari *FUN*, *FB* dan *PRG*, ketiganya merupakan unsur yang digunakan untuk membangun sebuah *project* aplikasi kendali industri. *Project* dibuat dengan bantuan perangkat lunak FPWIN Pro 4.0 versi demo dan PLC Nais FP0-C16T sebagai kontroler.

Dalam penelitian ini, IEC 61131-3 diimplementasikan untuk otomatisasi pembuatan tiramisu. Hasil pengujian dan implementasi membuktikan bahwa antara satu bahasa dengan bahasa lainnya, memiliki hasil dan tingkat keberhasilan yang sama, artinya dengan IEC 61131-3, *project* yang sama dapat diprogram dengan bahasa pemrograman yang berbeda-beda.

Kata-kunci : IEC 61131-3, PLC Nais, FPWIN Pro 4.0 dan Tiramisu.

1. PENDAHULUAN

1.1 Latar Belakang

Selama sepuluh hingga lima belas tahun terakhir, berbagai bahasa pemrograman yang berbeda telah digunakan untuk memprogram PLC. Untuk satu bahasa pemrograman saja misalnya *Ladder Diagram*, tiap-tiap jenis PLC mempunyai aturan dan cara pemrograman yang berbeda-beda. Padahal industri modern biasanya tidak hanya menggunakan satu jenis PLC saja melainkan berbagai jenis PLC, hal ini menjadi tidak efisien, baik dari segi waktu maupun uang.

Metode yang digunakan dalam implementasi IEC 61131-3 adalah dengan teknik *top-down* artinya implementasi dimulai dengan menentukan terlebih dahulu elemen setelah itu baru menentukan bahasa pemrograman yang digunakan (*Ladder Diagram, LD; Function Block Diagram, FBD; Sequential Function Chart, SFC; Instruction List, IL; atau Structure Text, ST*).

Otomatisasi pembuatan tiramisu merupakan miniatur proses di industri makanan, sehingga dapat dijadikan objek untuk implementasi IEC 61131-3.

1.2 Tujuan

Tujuan yang hendak dicapai pada penelitian ini, yaitu :

1. Mempelajari dan memperkenalkan pemrograman PLC standar IEC 61131-3.
2. Mengimplementasikan IEC 61131-3 pada otomatisasi pembuatan tiramisu.

1.3 Pembatasan Masalah

penelitian ini dibatasi pada hal-hal sebagai berikut :

1. Membahas IEC 61131 bagian ketiga (IEC 61131-3): Standarisasi pemrograman PLC.
2. Menggunakan perangkat lunak FPWIN Pro 4.0 versi demo untuk memprogram PLC sesuai standar IEC 61131-3.
3. Menggunakan PLC Nais seri FP0-C16T.
4. Plant yang dikendalikan adalah sistem pembuat Tiramisu.
5. Metode kontrol yang digunakan adalah on-off.

2. KAJIAN PUSTAKA

2.1 Sistem Kontrol Terdistribusi (DCs)^[3]

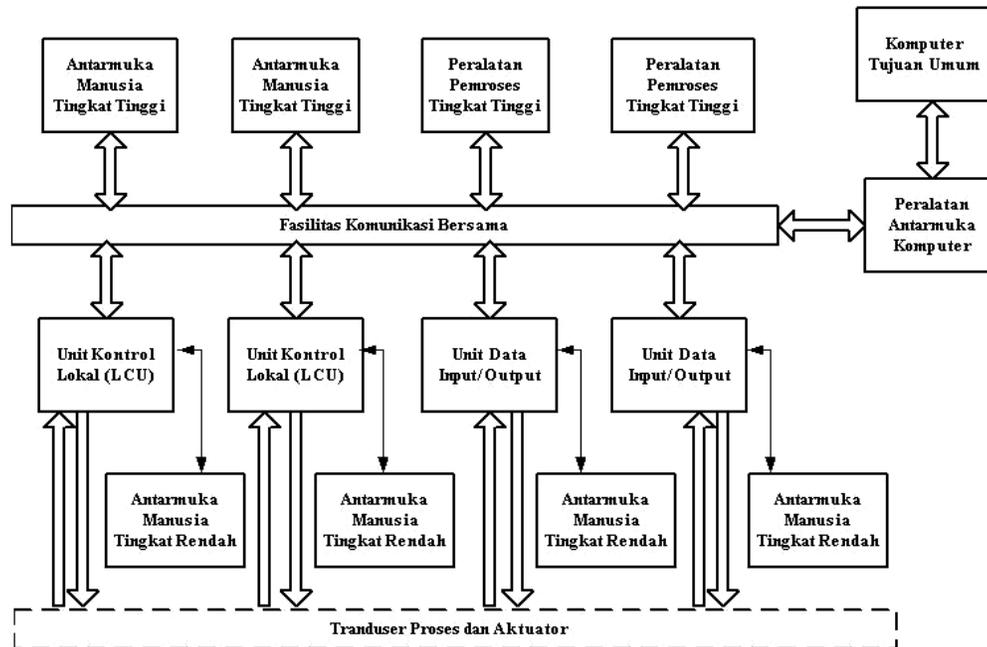
Salah satu keunggulan DCS adalah tidak bersifat *single point failure* artinya kerusakan pada salah satu unit kontrol tidak akan mematikan seluruh sistem yang akan berakibat kerugian, hal ini dikarenakan pada DCS, plant yang tersebar di berbagai tempat dikontrol secara langsung oleh masing-masing unit pengontrol lokal (*Local Control*

Unit, LCU). LCU merupakan bagian terkecil dari DCS yang mampu melakukan fungsi kontrol, misalnya PLC atau mikroprosesor.

Sebuah LCU tidak akan mampu melakukan fungsi kontrol secara baik jika tidak diprogram. Bahasa pemrograman yang digunakan untuk memprogram LCU diantaranya harus mampu mengakomodasi kemampuan pengguna dengan latar belakang pendidikan yang berbeda untuk menentukan

fungsi dan komputasi kontrol yang akan digunakan untuk mengendalikan plant. Disamping itu bahasa pemrograman LCU harus mampu melakukan fungsi komunikasi.

Gambar 2.1 adalah bentuk arsitektur DCS secara umum:



Gambar 2.1 Arsitektur DCS

2.2 IEC 61131-3^[10]

Untuk mengakomodasi semua persyaratan yang dibutuhkan oleh LCU dalam hal ini adalah *Programmable Logic Controller (PLC)* maka negara-negara industri di dunia mendirikan *International Electro-technical Commission (IEC)* yang bertugas untuk mengkaji perancangan lengkap dari sebuah PLC. IEC 61131-3 merupakan bagian ketiga dari IEC 61131 yang membahas mengenai standar pemrograman PLC.

2.2.1 Elemen Umum

Elemen umum ini terdiri dari kumpulan karakter (ISO 646), tipe data, variabel, POU (FUN, FB dan PRG) dan elemen konfigurasi (*Configuration, Resources* dan *Task*). *Configuration* adalah keseluruhan perangkat lunak yang ada pada tingkat tertinggi dari sebuah sistem *Programmable Logic Controller (PLC)*. Dalam sebuah *Configuration* dapat ditentukan satu atau lebih *resources*. *Resources* merupakan perangkat lunak yang berfungsi sebagai fungsi pemrosesan sinyal dan fungsi antarmuka

antara mesin dengan manusia. Dalam sebuah *resources* dapat ditentukan satu atau lebih *tasks*. *Tasks* mengendalikan eksekusi dari himpunan program (PRG) dan/atau blok fungsi (FB), baik secara periodik (*time-driven*) maupun ketika terjadi *trigger* khusus/interupsi (*event-driven*).

2.2.2 Bahasa Pemrograman

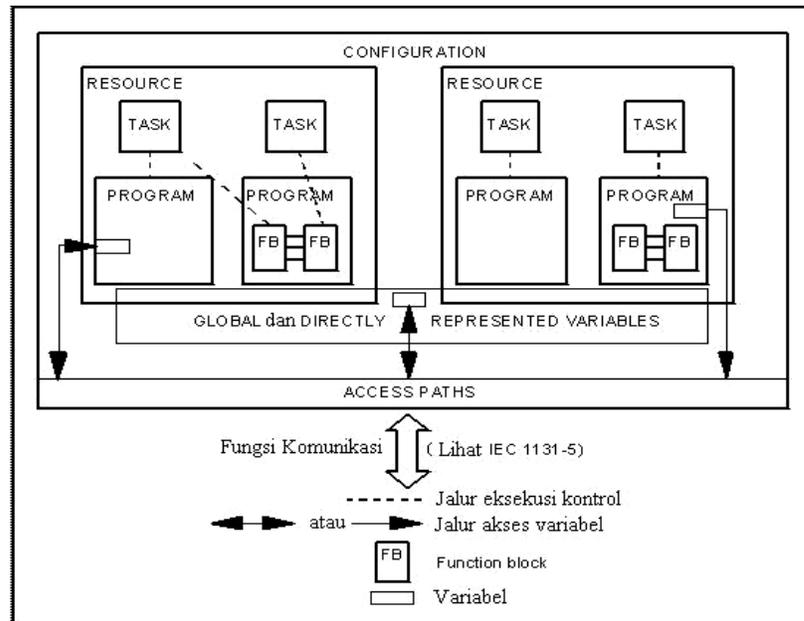
IEC 61131-3 mengakomodasi 5 bahasa pemrograman PLC berbeda, yaitu:

a. Ladder Diagram (LD)

Pemrograman PLC dengan menggunakan simbol-simbol relay elektromekanis yang terdiri dari *contact* dan *coil*.

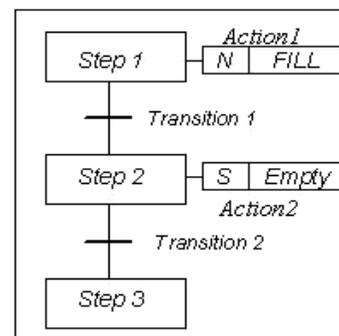
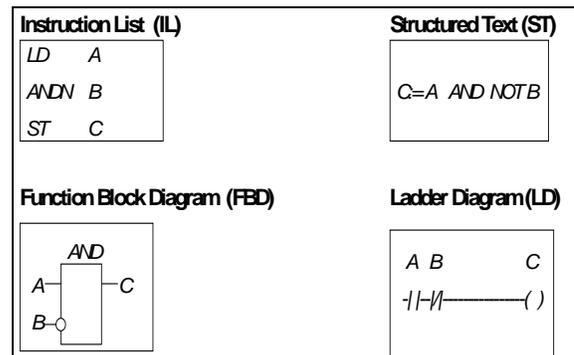
b. Function Block Diagram (FBD)

Penggunaan blok-blok fungsi standar maupun buatan pengguna sendiri dalam memprogram PLC.



Gambar 2.2 Model perangkat lunak IEC 61131-3

- c. *Sequential Function Chart (SFC)*
SFC menggambarkan secara grafis aksi sekuensial dari sebuah kontrol proses. SFC terdiri dari step yang terhubung dengan blok aksi dan transisi. Masing-masing step merepresentasikan keadaan (state) tertentu dari sebuah sistem yang dikendalikan. Sebuah transisi berkenaan dengan sebuah kondisi, dimana jika benar akan menyebabkan step sebelumnya tidak aktif dan step selanjutnya aktif. Step-step yang terhubung ke blok aksi akan menjalankan aksi kontrol tertentu. Masing-masing elemen SFC dapat diprogram dengan sembarang bahasa IEC, termasuk SFC itu sendiri. Karena elemen SFC membutuhkan memori untuk menyimpan informasi state maka SFC hanya dapat digunakan dalam POU jenis Program (PRG) dan Function Block (FB).
- d. *Instruction List (IL)*
PLC diprogram dengan serangkaian instruksi/perintah dan tiap instruksi harus dimulai pada baris baru.
- e. *Structured Text (ST)*
Pemrograman PLC dengan menggunakan bahasa tingkat tinggi seperti PASCAL.



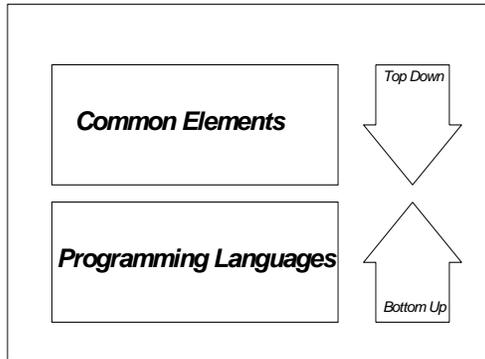
Gambar 2.3 Bahasa pemrograman IEC 61131-3

Kelima bahasa pemrograman tersebut bisa digabung/digunakan secara bersamaan ataupun sendiri-sendiri tergantung proses yang akan dikendalikan.

2.2.3 Fitur-fitur IEC 61131-3

IEC 61131-3 ini memiliki fitur-fitur antara lain:

- a. IEC 61131-3 menyarankan pengembangan program yang terstruktur rapih yaitu dengan teknik 'top-down' atau 'bottom-up'.



Gambar 2.4 Teknik pemrograman IEC 61131-3

- b. Standar IEC 61131-3 memerlukan penegasan tipe data.
- c. Fasilitas eksekusi kontrol dengan adanya *tasks*.
- d. Adanya SFC yang menyediakan fasilitas untuk memecah program dengan karakteristik sekuensial kompleks.
- e. Standar IEC 61131-3 mengakomodasi 5 bahasa pemrograman yang fleksibel, yakni tiga bahasa grafis (LD, FBD dan SFC) dan dua bahasa tektual (IL dan ST) untuk memecahkan aplikasi kontrol yang berbeda. Perancang sistem bebas memilih bahasa yang paling cocok untuk jenis aplikasi yang paling sesuai. Tiap POU's dapat diprogram dengan bahasa apapun.

2.3 Pembuatan Tiramisu^[5]

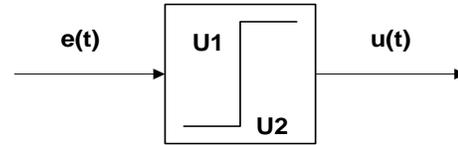
Secara umum proses pembuatan tiramisu dibagi menjadi 3 yaitu:

- a. Pencampuran/Pemanasan
Adonan dipanaskan sampai mencapai suhu 40°C dengan tujuan membunuh bakteri-bakteri yang mungkin masih terkandung dalam telur.
- b. Homogenisasi
Homogenisasi ini bertujuan untuk penguatan adonan yaitu dengan cara mendinginkan adonan sampai suhu sekitar 4°C.
- c. Pembekuan
Pembekuan bertujuan membentuk tekstur tiramisu supaya mengeras. Pembekuan merupakan proses yang terus-menerus dan ini dilaksanakan pada suhu sekitar -10 °C.

2.4 Sistem Kontrol ON-OFF^[4]

Pada sistem kontrol ON-OFF, elemen kontrol hanya memiliki dua posisi tertentu yaitu ON dan OFF. Kontrol ON-OFF memiliki karakteristik sinyal

keluaran dari kontroler $u(t)$ tetap pada salah satu nilai maksimum atau minimum tergantung apakah sinyal kontrol kesalahan positif atau negatif. Diagram blok kontroler ON-OFF yang memiliki masukan $e(t)$ dan keluaran $u(t)$, ditunjukkan pada Gambar 2.5.



Gambar 2.5 Diagram blok kontroler ON-OFF.

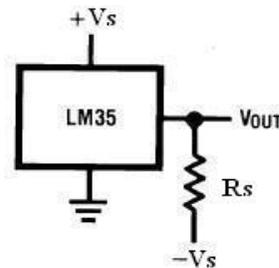
Aksi kontrol ON-OFF ditunjukkan pada persamaan (2.1).

$$u(t) = \begin{cases} U_1, & e(t) > 0 \\ U_2, & e(t) < 0 \end{cases} \dots\dots 2.1$$

Persamaan (2.1) memiliki nilai U_1 dan U_2 yang konstan. Nilai minimum U_2 dapat sebesar nol atau $-U_1$. Pada sistem kontrol kalang tertutup (*close loop*), sinyal $e(t)$ merupakan sinyal kesalahan aktuasi (*error*) sebesar selisih antara sinyal masukan dengan sinyal umpan balik.

2.5 Sensor suhu LM35

Merupakan sensor suhu linear dengan keluaran 10mV/°C. Rangkaian *Full-Centrigade Temperature Sensor* seperti yang ditunjukkan pada Gambar 2.6 digunakan untuk mengukur suhu dari -55 °C sampai 150°C. Nilai V_s untuk sensor ini dapat berkisar antara +35V- -0,2V.



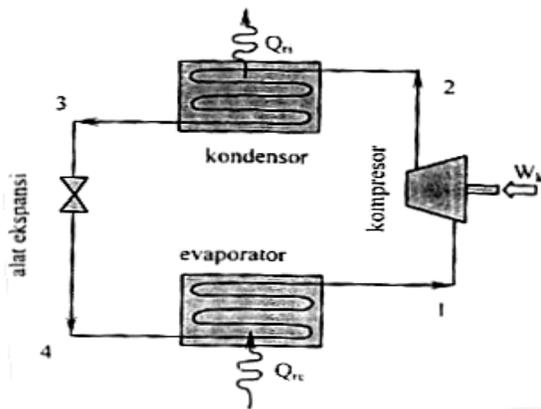
Gambar 2.6 Rangkaian *Full-Centrigade Temperature Sensor*

Nilai R_s dihitung menurut persamaan:

$$R_s = \frac{-V_s}{50\mu A} \dots\dots\dots 2.2$$

2.6 Mesin Refrigerasi

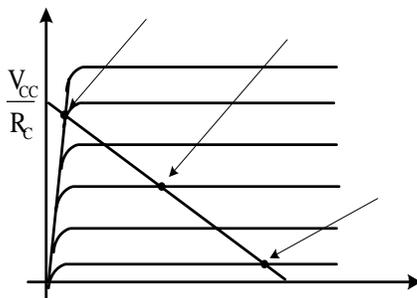
Setiap mesin pendingin memiliki beberapa komponen dasar dan suatu siklus kompresi uap seperti ditunjukkan pada Gambar 2.7



Gambar 2.7 Siklus kompresi uap standar.

2.7 Transistor sebagai sakelar

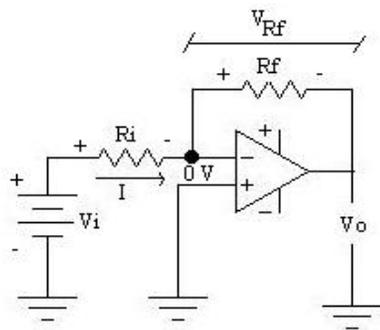
Transistor dapat digunakan sebagai sakelar elektronik jika transistor tersebut dioperasikan pada salah satu dari saturasi atau titik sumbat dan tidak di tempat-tempat sepanjang garis beban dc.



Gambar 2.8 Kurva karakteristik dan garis beban dc transistor

2.8 Penguat Membalik

Penguat membalik merupakan suatu penguat dimana tegangan keluarannya (V_o) mempunyai polaritas yang berkebalikan dengan tegangan masukan atau V_i .

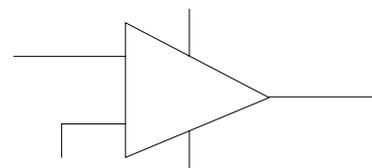


Gambar 2.9 Rangkaian penguat membalik

$$V_o = -\frac{R_f}{R_i} \times V_i \dots\dots\dots (2.3)$$

2.9 Pembanding

Pembanding hanya mempunyai 2 level keluaran yaitu $+V_{sat}$ dan $-V_{sat}$. Keluaran pembanding akan berharga $+V_{sat}$ jika level tegangan masukan (V_{in}) pada terminal tak membalik lebih besar dari level tegangan masukan terminal membalik (V_{ref}), dan akan berharga $-V_{sat}$ jika level tegangan masukan (V_{in}) pada terminal tak-membalik lebih kecil dari level tegangan masukan pada terminal membalik (V_{ref}).



Gambar 2.10 Rangkaian dasar pembanding (comparator)

2.10 PLC Nais FP0-C16T

PLC ini memiliki spesifikasi antara lain sebagai berikut:

- a. Memiliki ukuran W25 x H90 x D60 mm.
- b. Memiliki 8 masukan dan 8 keluaran dengan ekspansi maksimal 112 point.
- c. Kapasitas program 2720 step (5,4 K Byte).
- d. Beroperasi kecepatan tinggi yaitu dengan kecepatan *scanning* sekitar 1 ms untuk program dengan 500 step.
- e. Memiliki EEPROM sebagai memori program.
- f. Memiliki 1008 internal relay, 144 timer dan counter, 1660 data register dan 64 spesial internal relay.
- g. Beroperasi dengan catu daya 24 V DC dan arus masukan sekitar 4,3 mA.
- h. Tipe keluarannya *NPN open collector*.
- i. Menggunakan *photocoupler* sebagai metode isolasi rangkaian internal.

2.11 Solid State Relay (SSR)

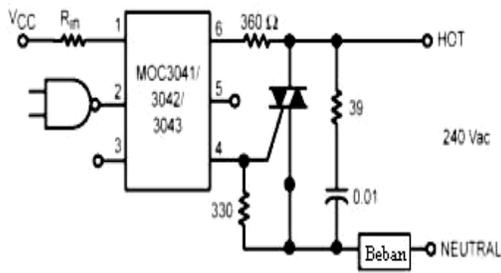
SSR dirangkai dengan menggunakan *optotriac-coupler* jenis MOC3041 yang difungsikan sebagai saklar AC sebagai pengganti relay mekanis. Rangkaian terpadu ini dilengkapi dengan *Zero Cross Detector* yang digunakan untuk menyulut triac setiap terjadi perpindahan dari siklus positif ke negatif atau sebaliknya, tepatnya pada saat tegangan nol dari sumber AC.

$$I_B > I_{B(SAT)}$$

SATURASI

$$I_B = I_{B(SAT)}$$

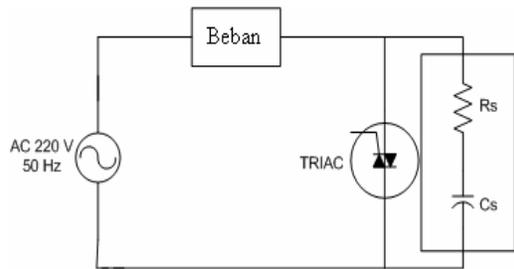
$$I_B$$



Gambar 2.11 Rangkaian SSR dengan MOC 3041

2.12 Rangkaian Snubber

Rangkaian snubber digunakan sebagai rangkaian proteksi triac terhadap dv/dt yang tinggi pada beban induktif.



Gambar 2.12 Rangkaian snubber

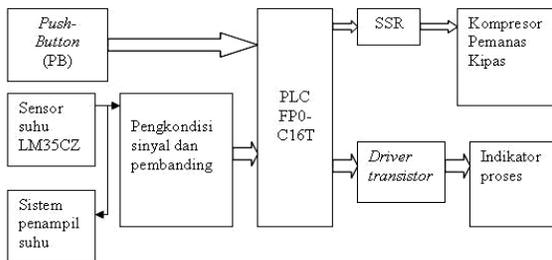
$$R_s = \frac{V_s}{I_{TD}} \dots\dots\dots (2.4)$$

$$\frac{dV}{dt} = \frac{0,632.V_s}{R_s C_s} \dots\dots\dots (2.5)$$

3. PERANCANGAN SISTEM

3.1 Perancangan Perangkat Keras

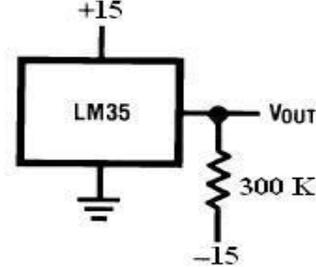
Diagram blok sistem secara keseluruhan pada perancangan alat ini dapat dilihat pada Gambar 3.1 sebagai berikut :



Gambar 3.1 Diagram blok sistem.

3.1.1 Perancangan rangkaian sensor suhu LM35CZ

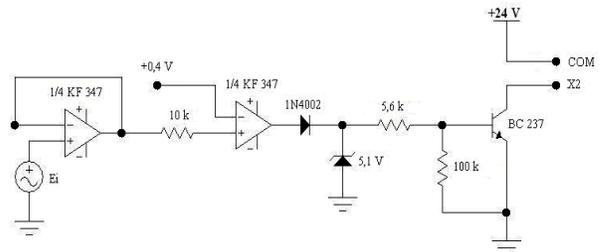
Sensor ini dirangkai Full-Centrigade Temperature Sensor agar bisa mendeteksi suhu dengan range maksimal yaitu dari -55°C sampai 150°C, seperti ditunjukkan pada Gambar 3.2



Gambar 3.2 Rangkaian Full-Centrigade Temperature Sensor

3.1.2 Perancangan pembanding 1

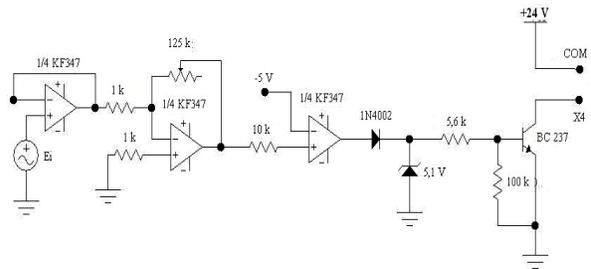
Pembanding 1 dirancang untuk menghasilkan logika '1' pada masukan X2 PLC ketika suhu mencapai 40°C.



Gambar 3.3 Rangkaian pembanding 1

3.1.3 Perancangan pengkondisi sinyal 1 dan pembanding 2

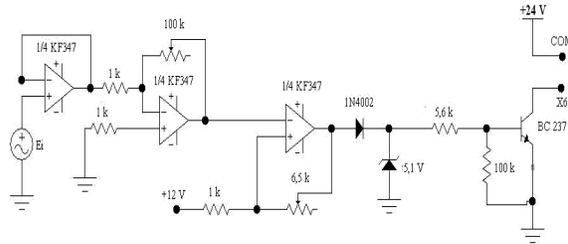
Pembanding 2 dirancang untuk menghasilkan logika '1' pada masukan X4 PLC ketika suhu mencapai 4°C.



Gambar 3.4 Rangkaian pengkondisi sinyal 1 dan pembanding 2

3.1.4 Perancangan pengkondisi sinyal 2 dan pembanding 3

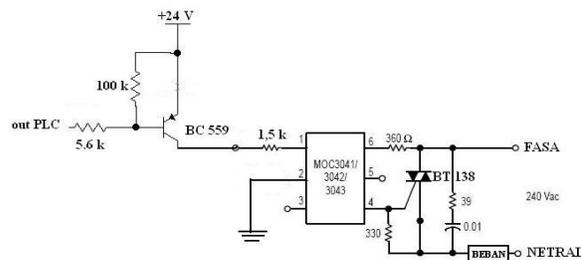
Pembanding 3 ini dirancang menghasilkan logika '0' ketika suhu mencapai -12°C dan logika '1' ketika suhu mencapai -8°C pada alamat masukan PLC X6.



Gambar 3.5 Rangkaian pengkondisi sinyal 2 dan pembanding 3

3.1.5 Perancangan rangkaian SSR

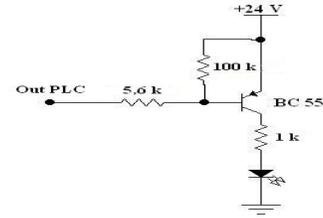
Rangkaian SSR digunakan sebagai pengganti relay mekanis. Pada penelitian ini SSR dirangkai dengan *optotriac-coupler* jenis MOC3041 yang dilengkapi dengan *zero crossing detector* pada rangkaian internalnya. SSR ini dirancang aktif *low*. SSR untuk kompresor dan kipas dilengkapi dengan rangkaian snubber sebagai proteksi, sedangkan pada pemanas tidak dilengkapi snubber, hal ini dikarenakan pemanas dianggap beban resistif.



Gambar 3.6 Rangkaian SSR

3.1.6 Perancangan driver transistor untuk indikator proses

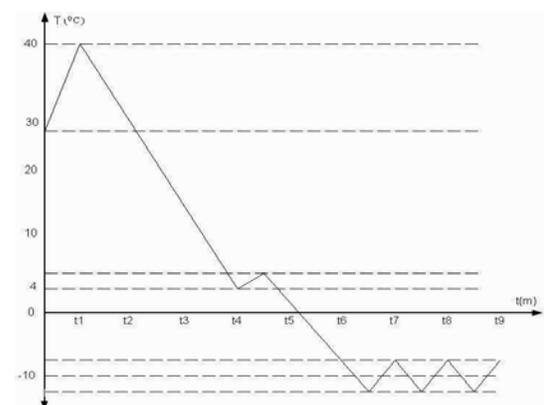
Driver transistor dirancang dengan memanfaatkan rangkaian transistor sebagai sakelar.



Gambar 3.7 Rangkaian driver transistor

3.2 Perancangan Perangkat Lunak

Pada tahap pertama refrigerator dipanaskan sampai mencapai suhu 40°C dengan menghidupkan pemanas. Setelah suhu 40°C tercapai maka pemanas dimatikan dan kompresor dihidupkan untuk mendinginkan suhu ruangan refrigerator sampai 4°C , tahap ini disebut tahap homogenisasi. Kompresor akan dihidupkan lagi setelah melewati waktu tunda minimal 3 menit untuk menyamakan tekanan antara kondensator yang bertekanan tinggi dengan evaporator yang bertekanan rendah sehingga tidak terjadi *overload* pada kompresor yang dapat menyebabkan kompresor rusak. Selama waktu tunda ini suhu ruangan akan naik sekitar 3°C . Setelah melewati waktu tunda maka tahap ketiga yaitu pembekuan dimulai dengan menghidupkan kompresor lagi sampai suhu ruangan refrigerator mencapai -10°C dan dijaga pada suhu tersebut yaitu dengan menghidup-matikan kompresor diantara suhu batas atas dan suhu batas bawah, hal ini dikarenakan PLC yang digunakan tidak mendukung untuk masukan analog sehingga digunakan pembanding dengan histeresis. Dari proses pembuatan tiramisu yang telah dideskripsikan maka dapat dibuat diagram state-nya



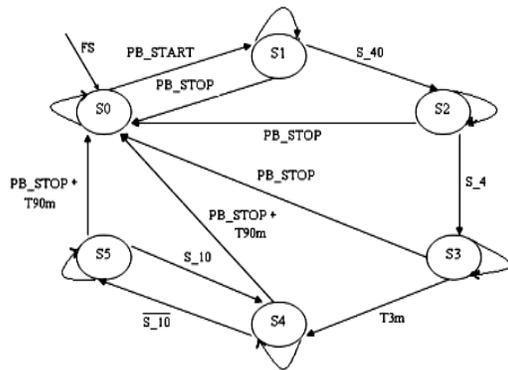
Gambar 3.8 Grafik proses pembuatan tiramisu

Tabel 3.1 State pembuatan tiramisu

State	Keluaran							
	H	F	C	LN	LP1	LP2	LP3	LPS
S0	0	0	0	0	0	0	0	0
S1	1	1	0	1	1	0	0	0
S2	0	1	1	1	0	1	0	0
S3	0	1	0	1	0	0	0	0
S4	0	1	1	1	0	0	1	0
S5	0	1	0	1	0	0	1	0

Keterangan:

- H : Pemanas (*Heater*)
- F : Kipas (*Fan*)
- C : Kompresor (*Compressor*)
- LN : LED indikator Normal
- LP1 : LED indikator Proses 1
- LP2 : LED indikator Proses 2
- LP3 : LED indikator Proses 3
- LPS : LED indikator Proses Selesai
- 1 : Aktif/hidup
- 0 : Mati



Gambar 3.9 Diagram state proses pembuatan tiramisu

4. HASIL PENELITIAN

4.1 Pengujian perangkat keras

4.1.1 Pengujian pembandingan 1

Pengujian pada semua pembandingan dilakukan secara tidak langsung artinya pengukuran tidak dilakukan pada keluaran pembandingan secara langsung melainkan pada masukan pembandingan yang berasal dari keluaran sensor.

Tabel 4.1 Hasil pengujian pembandingan 1

No	Keluaran Sensor	Konversi Tegangan ke Suhu (OC)	Penunjukan Termometer	Kesalahan
1	406	40,6	40	0,6
2	406	40,6	40	0,6
3	405	40,5	40	0,5
4	405	40,5	40	0,5
5	406	40,6	40	0,6
6	406	40,6	40	0,6
7	406	40,6	40	0,6
8	406	40,6	40	0,6
9	407	40,7	40	0,7
10	405	40,5	40	0,5
12	405	40,5	40	0,5
13	405	40,5	40	0,5
14	405	40,5	40	0,5
15	406	40,6	40	0,6
	405,60	40,56	40	0,56
Kesalahan rata-rata				1,40%

Jadi pembandingan 1 akan aktif jika suhu mencapai 40,56 °C atau tegangan keluaran sensor sebesar 405,6 mV dengan kesalahan rata-rata sebesar 1,4 % jika dibandingkan dengan thermometer.

4.1.2 Pengujian pengkondisi sinyal 1 dan pembandingan 2

Tabel 4.2 Hasil pengujian pengkondisi sinyal 1 dan pembandingan 2

No	Keluaran Sensor	Konversi Tegangan ke Suhu (OC)	Penunjukan Termometer	Kesalahan
1	40,5	4,05	4	0,05
2	39,7	3,97	3	0,97
3	39,7	3,97	3	0,95
4	40,1	4,01	4	0,01
5	41	4,10	4	0,1
6	40,1	4,01	4	0,01
7	40,2	4,02	4	0,02
8	42,6	4,26	4	0,26
9	40,3	4,03	4	0,03
10	39,1	3,91	3	0,91
11	39,5	3,95	3	0,95
12	40,4	4,04	4	0,04
13	40,1	4,01	4	0,01
14	40,9	4,09	4	0,09
15	40	4,00	4	0
	40,27	4,027	3,733	0,293
Kesalahan rata-rata				7,86%

Jadi pembanding 2 akan aktif jika suhu mencapai $4,027^{\circ}\text{C}$ atau tegangan keluaran sensor sebesar $40,27\text{ mV}$ dengan kesalahan rata-rata sebesar $7,86\%$ jika dibandingkan dengan thermometer.

4.1.3 Pengujian pengkondisi sinyal 2 dan pembanding 3

Pembanding 3 memiliki batas atas $-11,047^{\circ}\text{C}$ dengan kesalahan rata-rata sebesar $5,54\%$, batas bawah $-8,57^{\circ}\text{C}$ dengan kesalahan rata-rata sebesar $6,21\%$ dan histeresis sebesar $-11,07 - (-8,57) = 2,47^{\circ}\text{C}$.

Tabel 4.3 Hasil pengujian pengkondisi sinyal 2 dan pembanding 3

No	Batas atas pembanding 3				Batas bawah pembanding 3				
	Vo sensor (mV)	Vo ke suhu ($^{\circ}\text{C}$)	Penunjukan Thermometer ($^{\circ}\text{C}$)	Kesalahan	Vo sensor (mV)	Vo ke suhu ($^{\circ}\text{C}$)	Penunjukan Thermometer ($^{\circ}\text{C}$)	Kesalahan	
1	-106,1	-10,61	-10	0,61	-82,5	-8,25	-8	0,25	
2	-107,5	-10,75	-10	0,75	-85,6	-8,56	-8	0,56	
3	-108,0	-10,80	-10	0,80	-87,7	-8,77	-8	0,77	
4	-108,5	-10,85	-10	0,85	-88,7	-8,87	-8	0,87	
5	-108,5	-10,85	-10	0,85	-89,2	-8,92	-8	0,92	
6	-107,5	-10,75	-10	0,75	-92,1	-9,21	-9	1,21	
7	-112,8	-11,28	-10	1,28	-80,7	-8,07	-8	0,07	
8	-114,5	-11,45	-10	1,45	-81,0	-8,10	-8	0,10	
9	-116,0	-11,60	-11	0,60	-94,5	-9,45	-9	0,45	
10	-117,4	-11,74	-11	0,74	-82,5	-8,25	-8	0,25	
11	-110,0	-11,00	-11	0	-84,0	-8,40	-8	0,40	
12	-110,6	-11,06	-11	0,06	-83,6	-8,36	-8	0,36	
13	-110,6	-11,06	-11	0,06	-84,0	-8,40	-8	0,40	
14	-106,0	-10,60	-10	0,60	-84,1	-8,41	-8	0,41	
15	-113,0	-11,30	-11	0,30	-85,0	-8,50	-8	0,50	
Rata-rata	-110,47	-11,047	-10,47	0,58	-85,68	-8,57	-8,07	0,501	
Kesalahan rata-rata				5,54 %	Kesalahan rata-rata				6,21 %

4.1.4 Pengujian SSR

Pengujian SSR dilakukan dengan bantuan perangkat lunak FPWIN Pro 4.0 yang diprogram dengan bahasa LD. Setelah program dijalankan ternyata SSR dapat bekerja dengan baik, terbukti bahwa baik kompresor, kipas maupun pemanas dapat hidup jika ditekan tombol start dan mati jika ditekan tombol stop.

4.1.5 Pengujian driver transistor

Sama seperti pada pengujian SSR, pada pengujian driver transistor dilakukan dengan bantuan perangkat lunak FPWIN Pro 4.0 yang diprogram dengan bahasa LD. Setelah program dijalankan ternyata driver transistor dapat bekerja dengan baik, terbukti led indikator nyala ketika tombol start ditekan dan mati ketika ditekan tombol stop.

4.2 Pengujian perangkat lunak

Pengujian perangkat lunak dilakukan dengan meng-*compile* semua *project* yang telah dibangun. Parameter keberhasilannya adalah tidak adanya kesalahan (*error*) dan peringatan (*warning*).

Tabel 4.4 Hasil pengujian perangkat lunak

No	Bahasa	Hasil Kompilasi Program	Kesalahan & Peringatan
1	LD	173 step (346 Byte)*	0
2	FBD	182 step (346 Byte)	0
3	SFC 1	300 step (346 Byte)	0
4	SFC 2	299 step (346 Byte)	0
5	IL	299 step (346 Byte)	0
6	ST	244 step (346 Byte)	0
* = 1 step pembuatan memory 2 Byte			

4.3 Pengujian sistem keseluruhan

4.3.1 Pengujian pemberhentian proses

Pengujian ini dilakukan dengan cara menekan tombol stop pada tiap tahap. Tahap yang dimaksud disini adalah tahap pembuatan tiramisu secara umum, bukan tahap/step ataupun aksi yang ada dalam SFC. Parameter keberhasilan pada pengujian ini adalah proses akan berhenti dimanapun statenya. Berikut adalah hasil pengujiannya:

Tabel 4.5 Hasil pengujian pemberhentian proses

No	Penekanan tombol stop pada tahap	Bahasa Grafis				Bahasa Tekstual	
		LD	FBD	SFC 1	SFC 2	IL	ST
1	Pemanasan	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
2	Homogenisasi	Berhasil	Berhasil	Berhasil	*	Berhasil	Berhasil
3	Tunda	Berhasil	Berhasil	*	*	Berhasil	Berhasil
4	Pembekuan	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
5	Selesai	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil

* = Memori program FPWIN Pro 4.0 versi demo tidak cukup jika ditambah program pemberhentian proses pada tahap ini, sehingga tidak dilakukan pengujian pada tahap ini.

Dari Tabel 4.5 di atas terlihat bahwa pengujian pemberhentian proses pada tiap tahap untuk semua bahasa sudah berhasil. Pada SFC 1 (dengan bahasa LD) dan SFC 2, penambahan program pemberhentian proses mengakibatkan memori program pada FPWIN Pro 4.0 tidak cukup dikarenakan perangkat lunak ini masih versi demo yang hanya mampu menampung program maksimal 300 step (600 Byte).

4.3.2 Pengujian sistem keseluruhan

Pengujian sistem ini dilakukan dengan mengamati proses dari awal sampai akhir. Parameter keberhasilan dalam pengujian sistem ini adalah proses otomatisasi pembuatan tiramisu berjalan dengan baik dari awal sampai akhir kecuali jika kompresor mati terproteksi karena *overload* pada saat proses berjalan.

Tabel 4.6 Hasil pengujian sistem keseluruhan

No	Proses	Bahasa Grafis				Bahasa Tekstual	
		LD	FBD	SFC 1	SFC 2	IL	ST
1	Tahap 1	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
2	Tahap 2	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
3	Delay	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
4	Tahap 3	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
5	Selesai	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil

Setelah diimplementasikan dan diuji ternyata antara satu bahasa dengan bahasa lainnya, memiliki hasil dan tingkat keberhasilan yang sama untuk otomatisasi pembuatan tiramisu.

5. PENUTUP

5.1 Kesimpulan

- Dilihat dari segi kapasitas memori, implementasi dengan bahasa LD membutuhkan memori paling sedikit dibandingkan dengan bahasa lainnya yaitu 346 Byte. Sedangkan implementasi dengan bahasa SFC membutuhkan memori paling banyak yaitu 600 Byte.
- Dilihat dari tingkat kesulitannya, implementasi dengan bahasa FBD relatif lebih susah karena proses yang dikendalikan merupakan proses sekuensial sehingga harus memperhatikan kondisi masukan dan state-nya untuk setiap keluaran PLC. Sedangkan implementasi dengan bahasa LD, ST dan IL masih relatif mudah, ditambah metode kontrol yang digunakan hanya *on-off*.
- Implementasi dengan bahasa SFC sangat cocok untuk proses sekuensial seperti pada otomatisasi pembuatan tiramisu, baik dari segi kemudahan memprogram karena adanya fasilitas dekomposisi program, maupun dari segi kekompakan dan fleksibilitas, karena bagian-bagian SFC dapat diprogram dengan bahasa yang berbeda-beda kemudian digabungkan untuk melakukan tugas yang sama. Selain itu, pengawasan (*monitoring*) proses otomatisasi dengan SFC terlihat lebih jelas dibanding yang lain karena pengguna tinggal melihat step yang aktif dalam SFC.
- Setelah diimplementasikan dan diuji ternyata antara satu bahasa dengan bahasa lainnya, memiliki hasil dan tingkat keberhasilan yang sama untuk otomatisasi pembuatan tiramisu.
- Dengan IEC 61131-3, *project* yang sama dapat diprogram dengan bahasa pemrograman yang berbeda-beda, dan juga bisa diprogram oleh orang yang memiliki latar belakang pendidikan dan kemampuan berbeda-beda.

5.2 Saran

- Jika diperlukan masukan dan keluaran yang lebih banyak bisa menggunakan modul ekspansi dari PLC Nais FP0-C16T yaitu FP0-E16T.
- Sebaiknya dipasang sensor pada kompresor untuk mengetahui apakah kompresor dalam kondisi *overload* atau tidak, dimana kondisi *overload* tersebut bisa dianggap sebagai gangguan proses.
- Agar waktu proses otomatisasi pembuatan tiramisu lebih cepat bisa digunakan kompresor dan pemanas dengan daya yang lebih besar.
- Konsep pengandaan FB maupun PRG (*FB/PRG instances*) bisa digunakan untuk mengendalikan beberapa plant yang identik.

6. DAFTAR PUSTAKA

- [1] Coughlin, Robert and Federick Driscoll, *Penguat Operasional dan Rangkaian Terpadu Linear edisi kedua*, Erlangga, Jakarta.,1985.
- [2] Jacob M. Ph.D, C.C. Halkias, Ph.D, *Elektronika Terpadu*, Erlangga, Jakarta, 1990.
- [3] Lukas, Michael P, *Distributed Control System: Their Evaluation and Design*, Van Nostrand Reinhold Company, New York, 1986.
- [4] Ogata, Katsuhiko, *Teknik Kontrol Automatik (Sistem Pengaturan)*, Jilid 1, Erlangga, Jakarta, 1990.
- [5] Supriyadi, Subiyakto, *Otomatisasi Pembuatan Tiramisu Berbasis Mikrokontroler AT89S51*, Penelitian, Universitas Diponegoro, Semarang, 2006.
- [6] Stoecker, Wilbert F. Jones, Jerold W. Hara, Supratman, *Refrigerasi dan Pengkondisian Udara Edisi II*, Erlangga, Jakarta., 1989.
- [7] Tiegelkamp, Michael and John, Karl-Heinz, *IEC 61131-3: Programming Industrial Automation Systems*, Springer-Verlage BH, New York, 1995.
- [8] ---, *Data Sheet PLC FP0, Matsushita Electric Work, Ltd*
- [9] ---, <http://www.naisplc.com>
- [10] ---, <http://www.plcopen.org>