

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pada pelaksanaan program kerja magang pada perusahaan PT. Patara Teknik Solusindo, penulis diberikan posisi *SCADA engineer* untuk membantu *SCADA engineer* lainnya, yakni terutama pada pekerjaan *softcom*, kemudian selain itu, penulis juga membantu *staff* lainnya dalam pekerjaan seperti pembuatan HMI.

Selain itu, pada program kerja magang penulis diberikan sebuah pekerjaan proyek magang, yakni proyek pembuatan *logic* untuk *warning system* komponen *rectifier* yang dapat dimonitor secara *remote*. Pada proyek yang diberikan, penulis diberikan tugas untuk membuat sebuah *logic* yang dapat digunakan oleh RTU untuk melakukan *monitoring* dan menerima peringatan yang diberikan oleh *rectifier* secara *remote* yang akan digunakan oleh pihak PLN untuk melihat lokasi - lokasi terjadinya *error*.

3.2 Tugas dan Uraian Kerja Magang

Selama melakukan pekerjaan pada PT. Patara Teknik Solusindo, penulis berbagai tugas, tugas - tugas tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kegiatan Kerja Magang

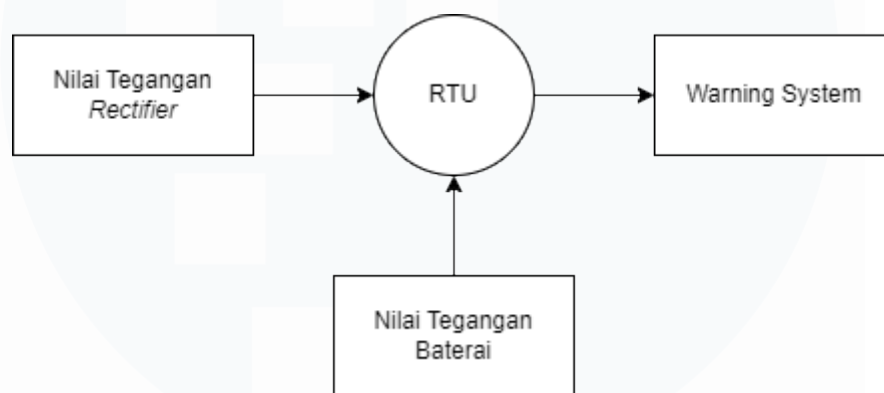
Bulan	Kegiatan
Juni	<ul style="list-style-type: none">● Mempelajari mengenai SCADA.● Mempelajari mengenai komunikasi pada SCADA.
Juli	<ul style="list-style-type: none">● Mempelajari aplikasi - aplikasi SCADA yang digunakan pada perusahaan.● Melakukan pembuatan HMI menggunakan aplikasi RapidScada.● Mempelajari aplikasi untuk

	melakukan <i>softcom</i> .
Agustus	<ul style="list-style-type: none"> ● Membuat dan merakit kabel serial dan kabel RJ-45. ● Melakukan <i>setup router</i> yang akan digunakan pada panel. ● Melakukan <i>softcom</i> pada panel RTU.
September	<ul style="list-style-type: none"> ● Membuat dan merakit kabel serial dan kabel RJ-45. ● Melakukan <i>softcom</i> pada panel RTU. ● Melakukan konsultasi proyek magang kepada supervisor. ● Memulai pengerjaan proyek magang.
Oktober	<ul style="list-style-type: none"> ● Melakukan <i>softcom</i> pada panel RTU. ● Melanjutkan pengerjaan proyek magang serta konsultasi laporan magang kepada supervisor. ● Melakukan dokumentasi untuk proyek magang. ● Berpamitan dengan <i>staff</i> pada perusahaan PT. Patara Teknik Solusindo.

3.3 Uraian Kerja Magang

Proyek awal merupakan sebuah panel MP (*Middle Point*) 20 kV yang terdapat pada sebuah gardu distribusi. Pada panel digunakan sebuah *rectifier* yang digunakan untuk menstabilisasi sistem panel melalui pemberian daya dengan bantuan komponen baterai agar panel tidak mengalami kematian sistem jika terjadi gangguan atau pemutusan tegangan pada panel. Panel MP 20 kV sendiri digunakan untuk mengendalikan kubikel pada gardu yang mengendalikan aliran pembagian listrik pada berbagai distrik.

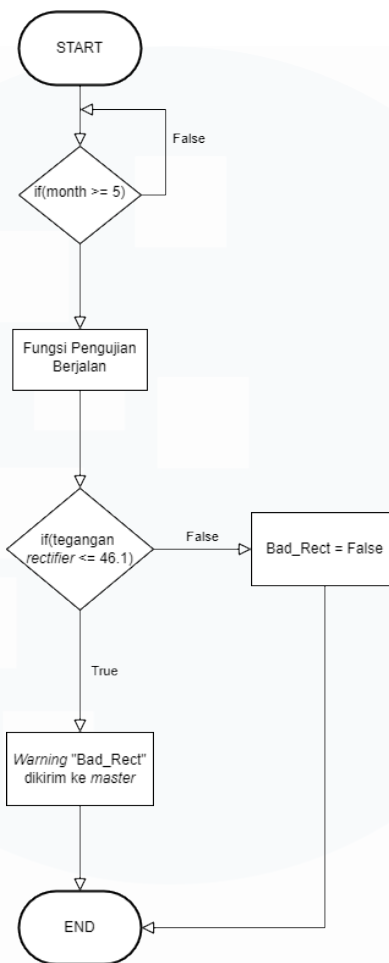
Pada proyek magang mengenai proyek pembuatan *warning system* komponen *rectifier* yang dapat dimonitor secara *remote*, sistem terdiri dari komponen RTU sebagai pengendali utama dan penghubung komunikasi antar panel dengan master, kemudian dibantu oleh komponen *I/O controller*, yakni *Patio 48* untuk mengendalikan berbagai komponen yang terdapat pada panel, dimana salah satunya merupakan komponen *rectifier*. Gambaran umum dari sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Sistem

Pada pengerjaan proyek magang digunakan *software* SEL RTAC untuk melakukan pemrograman dan pengujian akan *warning system* dan *logic* dengan komponen *rectifier*, diutamakan untuk dapat memberikan input jika *warning* pada komponen *rectifier* menyala. Pengerjaan proyek dilakukan melalui pembuatan dua buah *logic*, yakni *logic* mengenai yang berhubungan dengan komponen *rectifier* dan *logic* yang berhubungan dengan baterai yang terhubung dengan komponen *rectifier*.

Program yang dibuat tidak akan melakukan pengecekan jika seluruh sistem panel sudah terhubung kepada master secara *online*, sehingga program tetap akan jalan secara internal walau sistem sedang berada pada posisi *offline*. Program-program yang dibuat untuk sistem merupakan program “RECT_TEST” dan program “BATT_LOGIC”. Flowchart program ‘RECT_TEST’ dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Program “RECT_TEST”

Pada program, terdapat deklarasi atas setiap variabel pada fungsi yang dinamakan dengan “RECT_TEST” yang akan digunakan pada program. Variabel digunakan untuk mencatat tanggal yang digunakan pada fungsi pengujian status *rectifier*. Kemudian, terdapat variabel yang digunakan sebagai batas untuk menandakan telah terjadinya suatu kesalahan atau kerusakan jika data yang dibaca berada dibawah batas tersebut. Lalu, terdapat variabel *timer*, yakni TOF yang berfungsi untuk mengembalikan status atau kondisi *rectifier* kembali pada status atau kondisi awal. Gambar dari deklarasi variabel “RECT_TEST” dapat dilihat pada Gambar 3.3 dan Gambar 3.4.

```

1 PROGRAM RECT_TEST
2 VAR
3   CURRENT_TIME: DATE_AND_TIME;
4   LAST_RUN_TIME: DATE_AND_TIME := DT#2023-01-01-00:00:00; // Waktu awal (set sesuai kebutuhan)
5   TIME_DIFF: TIME;
6   THREE_MONTHS: LTIME := LTIME#90D; // 3 bulan dianggap 90 hari
7   TRIGGER_PERIODIC: BOOL := FALSE; // Trigger Secara otomatis setiap 3 Bulan
8   MANUAL_TRIGGER: BOOL; // Trigger di luar periodic 3 Bulan
9   TIM_TEST : TON := (PT:=T#60S); // Timer untuk pengujian selama 60 MENIT
10  BAD_RECT : BOOL;
11  MIN_CAP_BATT : REAL :=46.1; // Setpoint Minimum Battery Capacity
12  YEAR, MONTH, DAY, HOUR, MINUTE, SECOND: INT;
13  NEXT_RUN_TIME: DATE_AND_TIME;
14  FUNCTION_CALLED: BOOL := FALSE;
15  TIME_PERIODIC : INT := 3; // siklus 3 Bulan
16

```

Gambar 3.3 Deklarasi Variabel Program “RECT_TEST”

```

16
17  ON_RUNNING : BOOL;
18  SR1 : SR;
19  SR2 : SR;
20  TOF1 : TOF := (PT:=T#5S);
21  TOF2 : TOF := (PT:=T#5S);
22
23  RTRIG1:R_TRIG;
24  END_VAR
--

```

Gambar 3.4 Deklarasi Variabel Program “RECT_TEST”

Berdasarkan Gambar 3.5 hingga Gambar 3.9, dapat dilihat program yang akan menjalankan fungsi pengujian serta fungsi untuk menyalurkan status komponen *rectifier* secara *remote*. Sebelum dijalankan fungsi pengujian, program akan menjalankan fungsi untuk mengambil dan menentukan waktu sekarang dengan waktu yang ditentukan pada fungsi “NEXT_RUN_TIME”. Jika waktu sekarang telah mencapai batas, maka akan dijalankan fungsi pengujian yang akan memantau nilai tegangan yang terbaca pada komponen *rectifier* dan akan dibandingkannya dengan nilai batas yang telah ditentukan, jika nilai yang dipantau lebih kecil dibandingkan dengan nilai batas, maka program akan mengirimkan *output* bahwa terdapat masalah pada komponen *rectifier* melalui fungsi “BAD_RECT”. Kemudian untuk mencegah terjadinya pengoperasian pada saat dilakukannya pengujian, dibuat sebuah variabel bernama “ON_RUNNING” yang bertujuan untuk menandakan bahwa program sedang melakukan pengujian pada komponen *rectifier* tersebut.

```

Program
1 CURRENT_TIME := System_Time_Control_FOU.System_Time_DateAndTime ; // Ambil waktu saat ini
2
3 // Extract YEAR, MONTH, DAY from LAST_RUN_TIME
4 YEAR := Calendar.Year;
5 MONTH := Calendar.Month;
6 DAY := Calendar.Day;
7 HOUR := Calendar.Hour;
8 MINUTE := Calendar.Mins;
9 SECOND := Calendar.Sec;
10
11 // Tambahkan 5 bulan
12 MONTH := MONTH + TIME_PERIODIC;
13
14 // Periksa jika bulan lebih dari 12 (perlu mengubah tahun)
15 IF MONTH > 12 THEN
16     MONTH := MONTH - 12;
17     YEAR := YEAR + 1;
18 END_IF
19
20

```

Gambar 3.5 Program “RECT_TEST”

```

Program
20
21 // Hitung perbedaan waktu antara waktu saat ini dan waktu terakhir dijalanannya fungsi
22 TIME_DIFF := CURRENT_TIME - LAST_RUN_TIME;
23 // Buat NEXT_RUN_TIME
24 NEXT_RUN_TIME := STRING_TO_DT(Concat('DT#',Concat(INT_TO_STRING(YEAR),Concat('-',Concat(INT_TO_STRING(MONTH),Concat('-',Concat(INT_TO_STRING(DAY),Concat
25
26
27 IF CURRENT_TIME >= NEXT_RUN_TIME THEN
28     TRIGGER_PERIODIC := TRUE; // Jalankan fungsi
29     LAST_RUN_TIME := CURRENT_TIME; // Reset waktu terakhir fungsi dijalankan
30 ELSE
31     TRIGGER_PERIODIC := FALSE; // Fungsi tidak dijalankan
32 END_IF
33
34
35 //SRI (SET1 := Tags.RCD_RECTIFIER.operClear.ct1Val, RESET :=(TIM_TEST.ET = TIM_TEST.PT) OR BAD_RECT );
36 SRI (SET1 := Tags.RCD_RECTIFIER.operClear.ct1Val, RESET :=(TIM_TEST.ET = TIM_TEST.PT));
37
38 MANUAL_TRIGGER := SRI.Q1;
39
40 IF MANUAL_TRIGGER OR TRIGGER_PERIODIC THEN
41     TIM_TEST(IN:=TRUE);
42 ELSE
43     TIM_TEST(IN:=FALSE);
44 END_IF
45
46

```

Gambar 3.6 Program “RECT_TEST”

```

Program
45
46 IF (TIM_TEST.ET = TIM_TEST.PT) AND (Tags.BATT_VOLT.instMag > MIN_CAP_BATT) THEN
47     BAD_RECT := FALSE;
48     //Tags.RCD_RECTIFIER.operSet.ct1Val := TRUE;
49
50 ELIF (Tags.BATT_VOLT.instMag < MIN_CAP_BATT) THEN
51     BAD_RECT:= TRUE;
52     //Tags.RCD_RECTIFIER.operSet.ct1Val := TRUE;
53
54 ELIF (TIM_TEST.ET = TIM_TEST.PT) AND (Tags.BATT_VOLT.instMag < MIN_CAP_BATT) THEN
55     BAD_RECT := TRUE;
56     //Tags.RCD_RECTIFIER.operSet.ct1Val := TRUE;
57
58 ELIF (BAD_RECT := TRUE) AND (Tags.BATT_VOLT.instMag > MIN_CAP_BATT) THEN
59     BAD_RECT := FALSE;
60     //Tags.RCD_RECTIFIER.operSet.ct1Val := TRUE;
61
62 //ELIF BATT_CAPACITY.V_BATT < MIN_CAP_BATT THEN
63     //BAD_RECT := TRUE;
64
65 END_IF
66

```

Gambar 3.7 Program “RECT_TEST”

```

66
67 IF ((TIM_TEST.ET < TIM_TEST.PT) AND TIM_TEST.IN AND (MANUAL_TRIGGER OR TRIGGER_PERIODIC)) THEN
68   ON_RUNNING := TRUE;
69   TOF1(IN:=TRUE);
70   SystemTags.OUT101.operClear.ct1Val := Tags.RCD_RECTIFIER.operSet.ct1Val ;
71 ELSE
72   TOF1(IN:=FALSE);
73   SystemTags.OUT101.operClear.ct1Val := TRUE OR Tags.RCD_RECTIFIER.operSet.ct1Val ;
74 END_IF
75   SystemTags.OUT101.operClear.ct1Val := Tags.RCD_RECTIFIER.operSet.ct1Val ;
76
77 IF (TOF1.ET<TOF1.PT) THEN
78   ON_RUNNING := TRUE;
79 ELSE
80   ON_RUNNING := FALSE;
81   SystemTags.OUT101.operClear.ct1Val := TRUE OR Tags.RCD_RECTIFIER.operSet.ct1Val ;
82   //SystemTags.OUT101.operClear.ct1Val := TRUE;
83 END_IF
84

```

Gambar 3.8 Program “RECT_TEST”

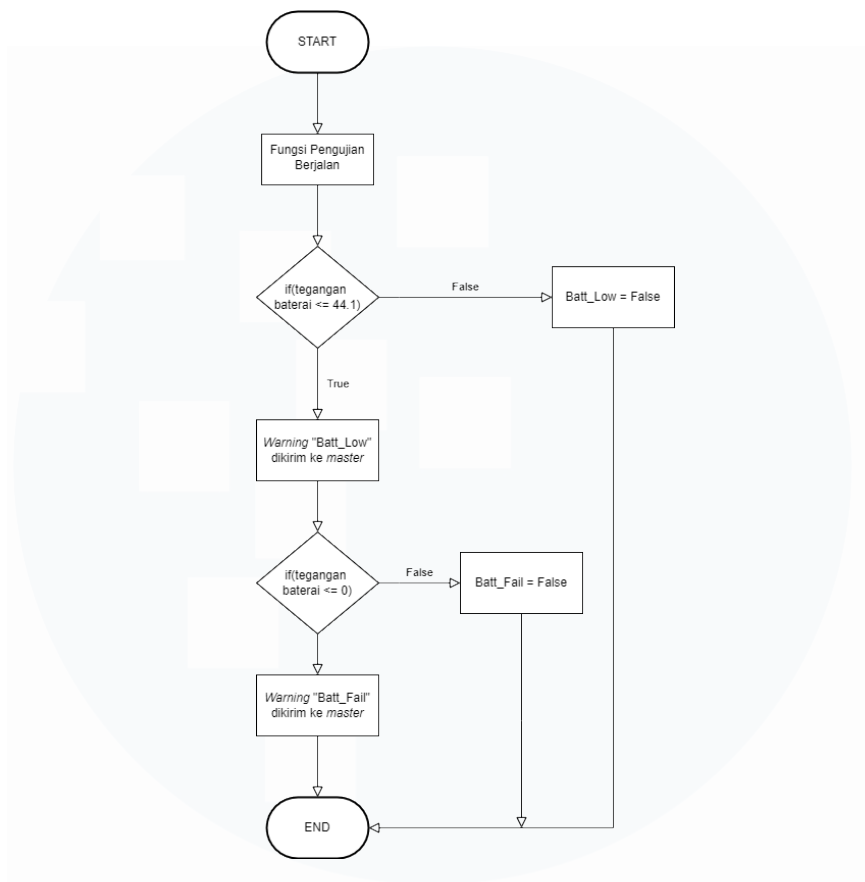
```

76
77 IF (TOF1.ET<TOF1.PT) THEN
78   ON_RUNNING := TRUE;
79 ELSE
80   ON_RUNNING := FALSE;
81   SystemTags.OUT101.operClear.ct1Val := TRUE OR Tags.RCD_RECTIFIER.operSet.ct1Val ;
82   //SystemTags.OUT101.operClear.ct1Val := TRUE;
83 END_IF
84
85 RTRIG1 (CLK:=BAD_RECT);
86 TOF2 (IN:= RTRIG1.Q);
87
88 IF TOF2.Q THEN
89   Tags.RCD_RECTIFIER.operSet.ct1Val := TRUE;
90 ELSE
91   Tags.RCD_RECTIFIER.operSet.ct1Val := FALSE;
92 END_IF
93
94 IF (TOF1.ET<TOF1.PT) THEN
95   ON_RUNNING := TRUE;
96 ELSE
97   ON_RUNNING := FALSE;
98 END_IF
99
100
101

```

Gambar 3.9 Program “RECT_TEST”

Lalu, untuk selanjutnya terdapat penjelasan program “BATT_LOGIC” yang mencakup deklarasi program, isi program dan flowchart dari program “BATT_LOGIC”. Flowchart dari program “BATT_LOGIC” dapat dilihat pada Gambar 3.10.



Gambar 3.10 Flowchart Program “BATT_LOGIC”

Pada Gambar 3.11 dapat dilihat variabel yang digunakan untuk menjalankan fungsi pengujian akan komponen baterai yang digunakan dan dihubungkan dengan komponen *rectifier*. Variabel yang digunakan merupakan variabel untuk menandakan status baterai dan variabel batas kapan status tersebut dipanggil.

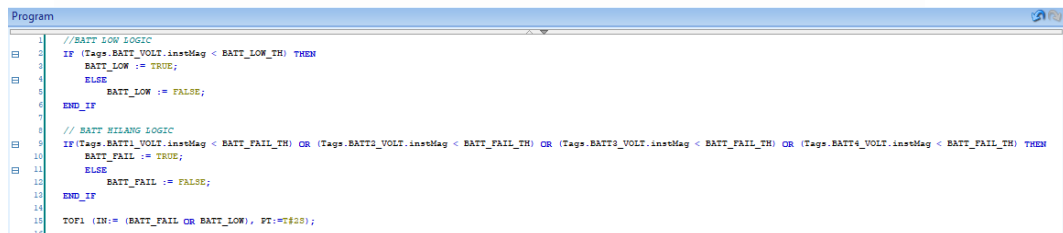
```

Program
1  PROGRAM BATT_LOGIC
2  VAR
3    BATT_FAIL : BOOL := FALSE;
4    BATT_LOW  : BOOL := FALSE;
5
6    BATT_FAIL_TH : REAL := 0;
7    BATT_LOW_TH  : REAL := 44.1;
8
9    TOF1 : TOF;
10
11 END_VAR
12
  
```

Gambar 3.11 Deklarasi Variabel Program “BATT_LOGIC”

Kode untuk program yang melakukan pengecekan dan menjalankan fungsi pemberian *warning* secara *remote* untuk komponen baterai dapat dilihat pada Gambar 3.12. Kode untuk komponen baterai hanya menggunakan satu fungsi utama, yakni membandingkan nilai pembacaan atas tegangan baterai yang terbaca pada komponen *rectifier* pada saat ini dengan nilai batas yang telah ditentukan. Jika nilai bacaan tegangan baterai berada dibawah batas yang ditentukan maka akan dijalankan fungsi untuk memberikan *output error* yang akan menyalurkan *output* tersebut melalui variabel "BATT_LOW" yang akan diberikan kepada master secara *remote*.

Selain itu, digunakan fungsi perbandingan untuk melakukan pengujian atas keberadaannya atau status dari setiap baterai dengan melakukan pengujian akan setiap baterai. Jika terdapat slot baterai yang kosong atau terdapat baterai yang rusak maka hasil pembacaan akan menunjukkan nilai 0 yang kemudian akan menjalankan fungsi "BATT_FAIL" untuk menandakan ada kesalahan pada baterai.



```
1 //BATT LOW LOGIC
2 IF (Tags.BATT_VOLT.instMag < BATT_LOW_TH) THEN
3   BATT_LOW := TRUE;
4 ELSE
5   BATT_LOW := FALSE;
6 END_IF
7
8 // BATT HILANG LOGIC
9 IF (Tags.BATT1_VOLT.instMag < BATT_FAIL_TH) OR (Tags.BATT2_VOLT.instMag < BATT_FAIL_TH) OR (Tags.BATT3_VOLT.instMag < BATT_FAIL_TH) OR (Tags.BATT4_VOLT.instMag < BATT_FAIL_TH) THEN
10  BATT_FAIL := TRUE;
11 ELSE
12  BATT_FAIL := FALSE;
13 END_IF
14
15 TOP1 (IH:= (BATT_FAIL OR BATT_LOW), PT:=T42S);
16
```

Gambar 3.12 Program "BATT_LOGIC"

3.4 Hasil Kerja Magang

Setelah mengerjakan proyek magang dengan judul “Pembuatan *Logic* untuk *Warning System* Komponen *Rectifier* yang Dapat Dimonitor Secara *Remote*” telah didapatkan hasil bahwa didapatkan dua buah *logic* program yakni, *logic* untuk *rectifier* dan *logic* untuk baterai. Melalui proyek ini didapati manfaat dimana dapat mengurangnya waktu respon terhadap masalah yang dapat timbul pada lapangan. Selain itu, melalui proyek ini juga dapat mengurangi dibutuhkan tenaga kerja manual untuk melakukan tugas pengawasan secara lokal. Selain itu, melalui penggunaan sistem SCADA, pada pengembangannya selanjutnya sistem ini dapat dimasukkan pada sistem SCADA secara besar.

Melalui proyek ini didapati juga harapan akan pengembangan yang dapat dilakukan seperti melakukan pengintegrasian dengan sistem IoT, mengembangkan sebuah sistem, fungsi atau bahkan sebuah *artificial intelligence* (AI) yang dapat melakukan prediksi akan kemungkinan terjadinya kerusakan berdasarkan analisis data - data sebelumnya. Lalu, memungkinkan juga melakukan sebuah penambahan fungsi untuk melakukan pengecekan akan *battery health*.

3.5 Kendala yang Ditemukan

Jumlah referensi beserta dengan dokumentasi mengenai aplikasi SEL RTAC yang tersedia sangat sedikit atau hampir tidak ada. Sehingga penulis mengalami kesulitan dalam mempelajari aplikasi SEL RTAC.

3.6 Solusi atas Kendala yang Ditemukan

Solusi untuk kesulitan akan mempelajari aplikasi SEL RTAC adalah memperbanyak bertanya kepada supervisor dan SCADA *engineer* senior lainnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA