

BAB IV

HASIL DAN PEMBAHASAN

Pada bab keempat ini akan mengulas tentang bagaimana hasil dari alat yang telah dirancang pada bab sebelumnya. Pengujian alat dilaksanakan dengan tujuan untuk menganalisa apakah sistem yang telah dirancang sebelumnya dapat berjalan normal atau tidak. Dengan pengujian alat tersebut, dapat menghasilkan alat pemantau kesehatan *relay* dengan mengolah data yang dihasilkan oleh *memory counter* sehingga dapat terbentuknya *preventive maintenance*.

4.1 Hasil Rangkaian Eletrik

Pada sub bab ini akan dijelaskan hasil dari rangkaian yang sebelumnya dirangkai dalam bentuk PCB.



Gambar 4. 1 Proses solder komponen pada modul



Gambar 4. 2 Module PCB

Module PCB ini berisikan beberapa komponen seperti *driver bus* (74LS245N), *controller*, I/O Expander (MCP23017SP), *memory* (AT24CP), *driver* (ULN2803) dan *relay* (HRS2H-S-DC24V). Yang mana module tersebut akan dipasang pada alat *switching unit* seperti pada gambar dibawah ini :



Gambar 4.3 Module PCB terpasang pada alat *switching unit*

4.2 Pembuatan Program Arduino Sehingga Menjadi Alat Pemantau Kesehatan Relay Pada Alat *Switching Unit*

Module PCB yang telah dicetak akan dilakukan pemrograman pada arduino untuk mendapatkan hasil yang diinginkan yaitu dapat menunjukkan berapa banyak relay telah dipakai saat operasional. Data tersebut kemudian akan diolah dalam bentuk excel sehingga dapat menunjukkan berapa persen kesehatan relay pada alat *switching unit*.

```

void incCounter(int relay)
{
  relay = relay + 7;
  byte counterValue = memory.read(relay);
  counterValue++;
  memory.write(relay, counterValue);
}

void printCounterValues()
{
  for (int i = 0; i < 32; i++)
  {
    Serial.print("K");Serial.print(i+1);Serial.print(" ");Serial.println(memory.read(i+8));
    delay(50);
  }
}

```

Gambar 4. 4 Program Counter Pada Arduino

Pada **Gambar 4.4** di atas merupakan program arduino yang berfungsi untuk counter yang mana nantinya akan diolah menjadi sistem pemantauan kesehatan relay pada alat *switching unit*.

Dimulai dari increment counter, kemudian

Relay = relay + 7 , merupakan address dari relay 1, dikarenakan address 0-7 telah digunakan untuk penyimpanan serial number untuk memory dan arduino pada module BMM.

Kemudian, sistem akan membaca nilai counter yang tersimpan pada memory. Lalu, akan dilakukan increment nilai counter. Setelah di increment, maka data tersebut akan disimpan pada memory. Sehingga memory akan melanjutkan data setelah increment tersebut.

Fungsi printCounterValues berguna untuk menampilkan jumlah perhitungan dari memory. Kemudian, yang akan dimunculkan nantinya akan bersimbol K1 4, yang mana K adalah relay, 1 adalah relay ke berapa, dan 4 adalah jumlah perhitungan yang disimpan pada memory.

```

void commandRelay(String command)
{
  byte relay;
  if (command[0] == 'C')
  {
    // Connect
    command[1] = (command[1] - 48) * 10;
    command[2] = command[2] - 48;
    relay = command[1] + command[2];
    if (readRelay(relay) == LOW)
    {
      // Increment the counter
      incCounter(relay);
    }
    execRelay(relay, HIGH);
  }
  else if (command[0] == 'D')
  {
    // Disconnect
    command[1] = (command[1] - 48) * 10;
    command[2] = command[2] - 48;
    relay = command[1] + command[2];
    execRelay(relay, LOW);
  }
  else if (command[0] == 'M' && command[1] == 'E' && command[2] == 'M')
  {
    printCounterValues();
  }
  else
  {
    Serial.println("Command Error!");
  }
}

```

Gambar 4. 5 Program Arduino

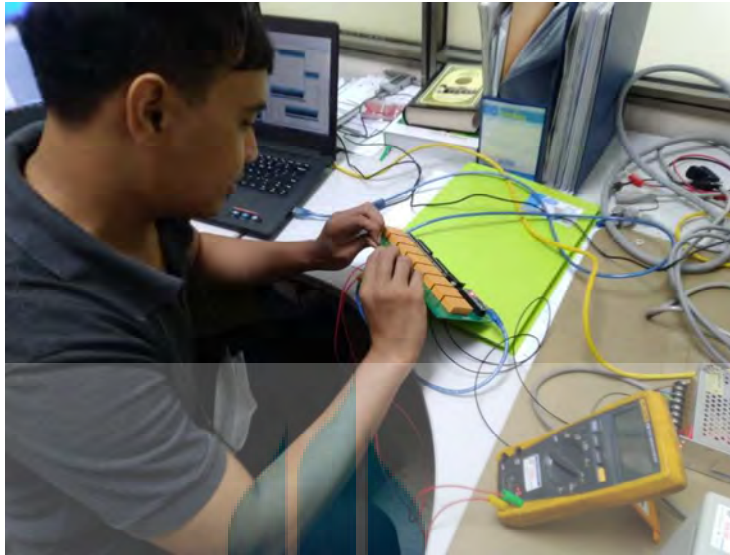
Pada **Gambar 4. 5** dibuat program perintah untuk menyalakan dan mematikan relay. Dengan mengetik C01 itu yang mana relay 1 akan menyala, dan seterusnya hingga C32. Dengan mengetik D01 itu yang mana relay 1 akan mati, dan seterusnya hingga D32. C melambangkan connect dan D melambangkan disconnect, yang akan digambarkan pada **Gambar 4.6**. Yang mana nantinya akan digunakan untuk self test secara manual untuk mengetahui program dengan module terhubung dan berjalan dengan yang diperintahkan.


```
COM8
Time Delay : 58.74 ms.
C32
Time Delay : 7.72 ms.
MEM
K1 21
K2 9
K3 5
K4 3
K5 4
K6 3
K7 4
K8 3
K9 3
K10 6
K11 3
K12 3
K13 3
K14 3
K15 3
K16 3
K17 4
K18 3
K19 3
K20 3
K21 3
K22 3
K23 3
K24 3
K25 2
K26 8
K27 3
K28 3
K29 6
K30 3
K31 3
K32 4
Time Delay : 2590.66 ms.
 Autoscroll  Show timestamp
```

Gambar 4. 7 Display Arduino

4.3 Self Test Pada Module BMM

Pada sub bab ini akan dilakukan *self test* pada module BMM. Pengujian ini berfungsi untuk mengetahui apakah komponen pada module sudah bekerja dengan baik dan program berjalan sebagaimana mestinya.



Gambar 4. 8 Self Test module BMM

UNIVERSITAS
MERCU BUANA

Tabel 4. 1 Tabel Hasil Self Test Module

No.	Relay	Fungsi	DMM		Hasil
			Hi	Lo	
1	K1	X1+,Y1+	X1-5	X1-18	PASS
2		X1-,Y1-	X1-9	X1-19	PASS
3	K2	X2+,Y1+	X1-6	X1-18	PASS
4		X2-,Y1-	X1-10	X1-19	PASS
5	K3	X3+,Y1+	X1-7	X1-18	PASS
6		X3-,Y1-	X1-11	X1-19	PASS
7	K4	X4+,Y1+	X1-8	X1-18	PASS
8		X4-,Y1-	X1-12	X1-19	PASS
9	K5	X1+,Y2+	X1-5	X1-20	PASS
10		X1-,Y2-	X1-9	X1-21	PASS
11	K6	X2+,Y2+	X1-6	X1-20	PASS
12		X2-,Y2-	X1-10	X1-21	PASS
13	K7	X3+,Y2+	X1-7	X1-20	PASS
14		X3-,Y2-	X1-11	X1-21	PASS
15	K8	X4+,Y2+	X1-8	X1-20	PASS
16		X4-,Y2-	X1-12	X1-21	PASS
17	K9	X1+,Y3+	X1-5	X1-22	PASS
18		X1-,Y3-	X1-9	X1-23	PASS
19	K10	X2+,Y3+	X1-6	X1-22	PASS
20		X2-,Y3-	X1-10	X1-23	PASS
21	K11	X3+,Y3+	X1-7	X1-22	PASS
22		X3-,Y3-	X1-11	X1-23	PASS
23	K12	X4+,Y3+	X1-8	X1-22	PASS
24		X4-,Y3-	X1-12	X1-23	PASS
25	K13	X1+,Y4+	X1-5	X1-24	PASS
26		X1-,Y4-	X1-9	X1-25	PASS
27	K14	X2+,Y4+	X1-6	X1-24	PASS
28		X2-,Y4-	X1-10	X1-25	PASS
29	K15	X3+,Y4+	X1-7	X1-24	PASS
30		X3-,Y4-	X1-11	X1-25	PASS
31	K16	X4+,Y4+	X1-8	X1-24	PASS
32		X4-,Y4-	X1-12	X1-25	PASS
33	K17	X1+,Y5+	X1-5	X1-4	PASS
34		X1-,Y5-	X1-9	X1-3	PASS
35	K18	X2+,Y5+	X1-6	X1-4	PASS
36		X2-,Y5-	X1-10	X1-3	PASS
37	K19	X3+,Y5+	X1-7	X1-4	PASS
38		X3-,Y5-	X1-11	X1-3	PASS
39	K20	X4+,Y5+	X1-8	X1-4	PASS
40		X4-,Y5-	X1-12	X1-3	PASS
41	K21	X1+,Y6+	X1-5	X1-2	PASS

No.	Relay	Fungsi	DMM		Hasil
			Hi	Lo	
42		X1-,Y6-	X1-9	X1-1	PASS
43	K22	X2+,Y6+	X1-6	X1-2	PASS
44		X2-,Y6-	X1-10	X1-1	PASS
45	K23	X3+,Y6+	X1-7	X1-2	PASS
46		X3-,Y6-	X1-11	X1-1	PASS
47	K24	X4+,Y6+	X1-8	X1-2	PASS
48		X4-,Y6-	X1-12	X1-1	PASS
49	K25	X1+,Y7+	X1-5	X1-14	PASS
50		X1-,Y7-	X1-9	X1-15	PASS
51	K26	X2+,Y7+	X1-6	X1-14	PASS
52		X2-,Y7-	X1-10	X1-15	PASS
53	K27	X3+,Y7+	X1-7	X1-14	PASS
54		X3-,Y7-	X1-11	X1-15	PASS
55	K28	X4+,Y7+	X1-8	X1-14	PASS
56		X4-,Y7-	X1-12	X1-15	PASS
57	K29	X1+,Y8+	X1-5	X1-16	PASS
58		X1-,Y8-	X1-9	X1-17	PASS
59	K30	X2+,Y8+	X1-6	X1-16	PASS
60		X2-,Y8-	X1-10	X1-17	PASS
61	K31	X3+,Y8+	X1-7	X1-16	PASS
62		X3-,Y8-	X1-11	X1-17	PASS
63	K32	X4+,Y8+	X1-8	X1-16	PASS
64		X4-,Y8-	X1-12	X1-17	PASS

Pada Tabel 4. 1 di atas menunjukkan module memiliki 32 relay. Setiap relaynya akan dilakukan continuity dengan digital multimeter untuk mengetahui apakah relay dan program sudah berjalan dengan baik. Berikut adalah cara pengujian *self test* pada module PCB :

1. Sambungkan output relay dengan digital multimeter untuk dilakukan continuity test.
2. Berikan perintah pada aplikasi arduino untuk menyalakan relay yang ingin diukur.
3. Apabila digital multimeter memberikan suara bip, maka relay itu dikatakan tersambung.
4. Berikan perintah pada aplikasi arduino untuk mematikan relay.
5. Berikan keterangan pada data relay tersebut dengan keterangan pass atau fail.

4.4 Pengujian Time Delay Pada Module

Pengujian kali ini digunakan untuk mengetahui waktu delay yang dibutuhkan relay untuk menyala saat diperintah dengan program arduino nano sebelumnya.

```

void loop()
{
  if (Serial.available())
  {
    timeDelay = micros();
    digitalWrite(13, HIGH);
    delay(5);
    serialCommand = "";
    serialCommand += char(Serial.read());
    serialCommand += char(Serial.read());
    serialCommand += char(Serial.read());
    Serial.println(serialCommand);
    while (Serial.available()){Serial.read();}
    commandRelay(serialCommand);
    digitalWrite(13, LOW);
    waktu = micros() - timeDelay;
    waktu = waktu / 1000;
    Serial.print("Time Delay : ");Serial.print(waktu);Serial.println(" ms.");
  }
}

```

Gambar 4. 9 Program time delay pada arduino

Pada **Gambar 4. 9** merupakan program perhitungan time delay yang saat relay dinyalakan. Sehingga setiap menyalakan relay pada perintah arduino, maka secara otomatis relay akan menyala dan memberikan keterangan time delay pada display arduino seperti **Gambar 4. 10** di bawah ini :

```

COM8
C01
Time Delay : 58.75 ms.
C01
Time Delay : 7.72 ms.
C02
Time Delay : 58.74 ms.
C02
Time Delay : 7.72 ms.
C03
Time Delay : 58.75 ms.
C03
Time Delay : 7.72 ms.
C04
Time Delay : 58.75 ms.
C04
Time Delay : 7.71 ms.
C05
Time Delay : 58.76 ms.
C05
Time Delay : 7.72 ms.
C06
Time Delay : 58.75 ms.
C06
Time Delay : 7.72 ms.
C07
Time Delay : 58.76 ms.
C07
Time Delay : 7.72 ms.
C08
Time Delay : 58.75 ms.
C08
Time Delay : 7.72 ms.
C09
Time Delay : 58.74 ms.
C09
Time Delay : 7.72 ms.
C10
Time Delay : 58.74 ms.
 Autoscroll  Show timestamp

```

Gambar 4. 10 Time delay module BMM

Pada **Tabel 4. 2** di bawah ini merupakan hasil dari time delay setiap relay yang ada pada module BMM. Delta T adalah perbedaan waktu antara relay pertama kali dinyalakan dan setelah dinyalakan kedua kalinya. Perbedaan waktu tersebut dikarenakan saat relay telah dinyalakan, maka program tidak perlu

menyalakan, cukup membaca apakah relay telah high saja. Apabila relay masih low maka program akan mengirimkan feedback dari relay menuju driver dan menyalakannya, disinilah terjadi perbedaan waktu yang disebut delay tersebut.

Tabel 4. 2 Tabel Hasil Time Delay Module BMM

No.	Relay	T0 (ms)	T1 (ms)	ΔT (Write EEPROM) ms
1	K1	58,75	7,72	51,03
2	K2	58,74	7,72	51,02
3	K3	58,74	7,72	51,02
4	K4	58,75	7,71	51,04
5	K5	58,76	7,72	51,04
6	K6	58,75	7,72	51,03
7	K7	58,76	7,72	51,04
8	K8	58,75	7,72	51,03
9	K9	58,74	7,72	51,02
10	K10	58,74	7,71	51,03
11	K11	58,75	7,72	51,03
12	K12	58,74	7,72	51,02
13	K13	58,74	7,72	51,02
14	K14	58,75	7,71	51,04
15	K15	58,76	7,72	51,04
16	K16	58,75	7,72	51,03
17	K17	58,76	7,72	51,04
18	K18	58,75	7,72	51,03
19	K19	58,74	7,72	51,02
20	K20	58,75	7,72	51,03
21	K21	58,74	7,72	51,02
22	K22	58,74	7,72	51,02
23	K23	58,75	7,71	51,04
24	K24	58,76	7,72	51,04
25	K25	58,75	7,72	51,03
26	K26	58,76	7,72	51,04
27	K27	58,75	7,72	51,03
28	K28	58,74	7,72	51,02
29	K29	58,75	7,72	51,03
30	K30	58,74	7,72	51,02
31	K31	58,74	7,72	51,02
32	K32	58,75	7,71	51,04
AVERAGE				51,0296875

4.5 Pemantau Kesehatan *Relay* Dari Alat *Switching Unit*

Pemantau kesehatan *relay* ini merupakan tindakan *preventive maintenance* yang mana mengambil data dari setiap *relay* yang digunakan dalam alat *switching unit*. Dengan menambahkan komponen EEPROM AT24C64 pada setiap *module* yang ada pada alat *switching unit*, yang mana EEPROM AT24C64 tersebut akan selalu menghitung setiap *relay* digunakan dan menyimpannya. Kemudian, EEPROM AT24C64 tersebut akan menerima perintah dari arduino nano untuk mengunduh hasil penyimpanan perhitungan saat *relay* setiap digunakan. Data yang diunduh tersebut akan disimpan dalam bentuk excel yang mana data inilah yang kita olah untuk melihat berapa persen kesehatan masing – masing *relay* pada setiap *module* yang ada di alat *switching unit*.

Pada **Tabel 4. 3** dan **Tabel 4. 4** di bawah ini adalah simulasi bagaimana data yang digunakan untuk memantau kesehatan *relay* pada setiap modulnya.

Tabel 4. 5 Tabel Pemantau Kesehatan Pada Alat *Switching Unit*

COUNT OF MODULE USED			
	MODULE 1	MODULE 2	MODULE 3
RELAY 1	56	54	52
RELAY 2	55	34	37
RELAY 3	62	63	47
RELAY 4	49	45	46
RELAY 5	49	62	49
RELAY 6	73	47	59
RELAY 7	49	50	43
RELAY 8	47	43	57
RELAY 9	56	42	58
RELAY 10	46	42	41
RELAY 11	68	38	52
RELAY 12	60	30	42
RELAY 13	59	49	59
RELAY 14	79	54	33
RELAY 15	75	64	32
RELAY 16	18	18	50
RELAY 17	18	31	16
RELAY 18	11	9	5
RELAY 19	21	28	22
RELAY 20	20	17	1
RELAY 21	17	31	13

COUNT OF MODULE USED			
	MODULE 1	MODULE 2	MODULE 3
RELAY 22	32	12	15
RELAY 23	11	2	10
RELAY 24	17	26	1
RELAY 25	2	12	14
RELAY 26	26	23	2
RELAY 27	11	19	32
RELAY 28	5	22	4
RELAY 29	3	7	26
RELAY 30	6	28	15
RELAY 31	23	22	31
RELAY 32	28	28	31
RATE	36	32,875	31,09375
RATE MODULE GENERAL	33,32291667		

Keterangan :

- = Penggunaan relay tersebut dibawah rata-rata operasional
■ = Penggunaan relay tersebut diatas rata-rata operasional

Tabel 4. 6 Tabel Persentase Pemantau Kesehatan Pada Alat *Switching Unit*

PERCENTAGE OF MODULE USED			
	MODULE 1 (%)	MODULE 2 (%)	MODULE 3 (%)
RELAY 1	0,00000373333333333333	0,00000360000000000000	0,00000346666666666667
RELAY 2	0,00000366666666666667	0,00000226666666666667	0,00000246666666666667
RELAY 3	0,00000413333333333333	0,00000420000000000000	0,00000313333333333333
RELAY 4	0,00000326666666666667	0,00000300000000000000	0,00000306666666666667
RELAY 5	0,00000326666666666667	0,00000413333333333333	0,00000326666666666667
RELAY 6	0,00000486666666666667	0,00000313333333333333	0,00000393333333333333
RELAY 7	0,00000326666666666667	0,00000333333333333333	0,00000286666666666667
RELAY 8	0,00000313333333333333	0,00000286666666666667	0,00000380000000000000
RELAY 9	0,00000373333333333333	0,00000280000000000000	0,00000386666666666667
RELAY 10	0,00000306666666666667	0,00000280000000000000	0,00000273333333333333
RELAY 11	0,00000453333333333333	0,00000253333333333333	0,00000346666666666667
RELAY 12	0,00000400000000000000	0,00000200000000000000	0,00000280000000000000
RELAY 13	0,00000393333333333333	0,00000326666666666667	0,00000393333333333333
RELAY 14	0,00000526666666666667	0,00000360000000000000	0,00000220000000000000
RELAY 15	0,00000500000000000000	0,00000426666666666667	0,00000213333333333333
RELAY 16	0,00000120000000000000	0,00000120000000000000	0,00000333333333333333
RELAY 17	0,00000120000000000000	0,00000206666666666667	0,00000106666666666667
RELAY 18	0,00000073333333333333	0,00000060000000000000	0,00000033333333333333
RELAY 19	0,00000140000000000000	0,00000186666666666667	0,00000146666666666667

PERCENTAGE OF MODULE USED			
	MODULE 1 (%)	MODULE 2 (%)	MODULE 3 (%)
RELAY 20	0,00000133333333333333	0,00000113333333333333	0,00000006666666666667
RELAY 21	0,00000113333333333333	0,00000206666666666667	0,00000086666666666667
RELAY 22	0,00000213333333333333	0,00000080000000000000	0,00000100000000000000
RELAY 23	0,00000073333333333333	0,00000013333333333333	0,00000066666666666667
RELAY 24	0,00000113333333333333	0,00000173333333333333	0,00000066666666666667
RELAY 25	0,00000013333333333333	0,00000080000000000000	0,00000093333333333333
RELAY 26	0,00000173333333333333	0,00000153333333333333	0,00000013333333333333
RELAY 27	0,00000073333333333333	0,00000126666666666667	0,00000213333333333333
RELAY 28	0,00000033333333333333	0,00000146666666666667	0,00000026666666666667
RELAY 29	0,00000020000000000000	0,00000046666666666667	0,00000173333333333333
RELAY 30	0,00000040000000000000	0,00000186666666666667	0,00000100000000000000
RELAY 31	0,00000153333333333333	0,00000146666666666667	0,00000206666666666667
RELAY 32	0,00000186666666666667	0,00000186666666666667	0,00000206666666666667
RATE	0,00000240000000000000	0,00000219166666666667	0,00000207291666666667
RATE MODULE GENERAL	0,0000022215277777778		

Keterangan :

- = Penggunaan relay tersebut dibawah rata-rata operasional
- = Penggunaan relay tersebut diatas rata-rata operasional

Pada **Tabel 4. 7** di atas, merupakan simulasi pada alat pemantau kesehatan yang datanya diambil dari komponen EEPROM AT24C64 lalu diolah. Data tersebut dihasilkan dari perhitungan yang disimpan oleh komponen EEPROM AT24C64 saat setiap penggunaan pada masing – masing *relay* pada modul. Data tersebut menjelaskan berapa banyak *relay* yang telah digunakan.

Sama halnya dengan **Tabel 4. 8** di atas, merupakan simulasi pada alat pemantau kesehatan yang datanya diambil dari komponen EEPROM AT24C64 lalu diolah, tetapi bukan dalam bentuk hitungan biasa melainkan adalah nilai berapa persen *relay* yang telah digunakan.

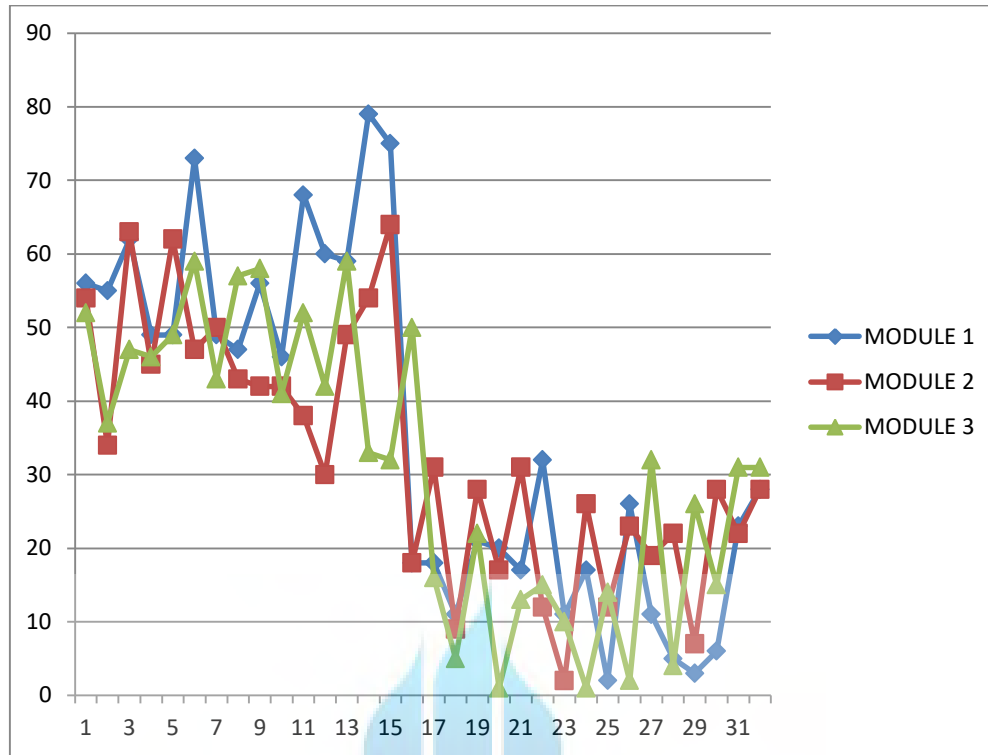
Agar pada **Tabel 4. 9** dan **Tabel 4. 10** lebih mudah untuk mengetahui relay mana saja yang telah di atas rata rata, kemudian berikan rules pada excel sehingga akan membuat display jumlah count dan persentase setiap relay melebihi rata rata penggunaan relay maka akan berwarna merah dan warna hijau untuk penggunaan relay dibawah rata rata.

Dari **Tabel 4. 11** di atas, kita dapat memperkirakan *relay* telah mencapai berapa persen kesehatannya. Semakin besar persen yang dihasilkan, maka *relay* tersebut semakin dekat dengan *lifetime* sehingga diperlukan penggantian *relay*. Dikarenakan pemesanan *relay* yang membutuhkan waktu yang lama, dengan persentase tersebut kita dapat memperkirakan estimasi datangnya *relay* dari vendor sehingga tidak mengganggu operasional dari *electrical workshop* sendiri, yang dapat kita sebut dengan *preventive maintenance*.

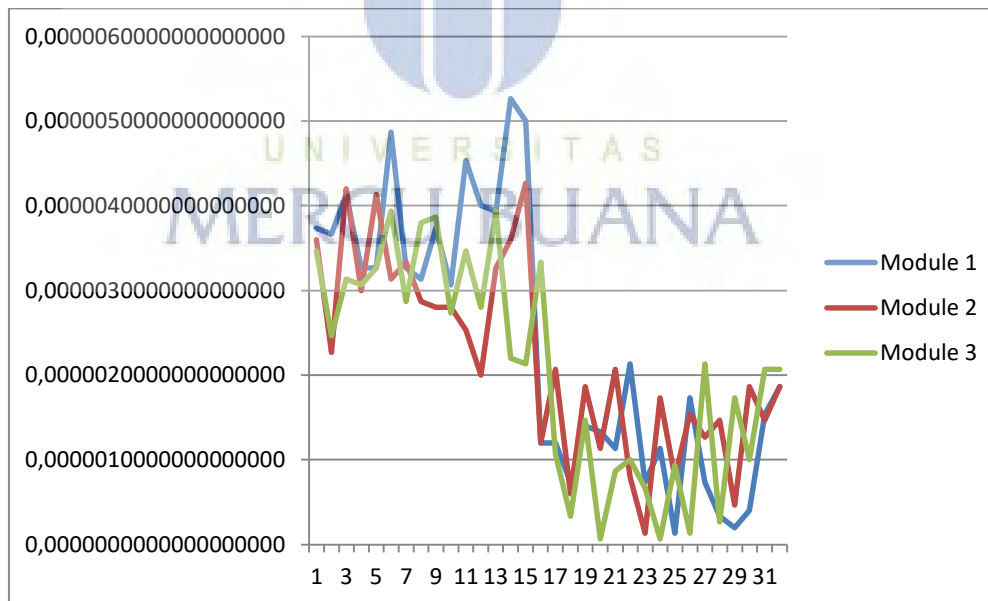
Berikut adalah perhitungan dari tabel persentase alat pemantau kesehatan *relay* dari setiap modulnya.

$$\text{Persentase} = \frac{\text{Jumlah Penggunaan Masing – Masing Relay}}{15.000.000} \times 100 \%$$

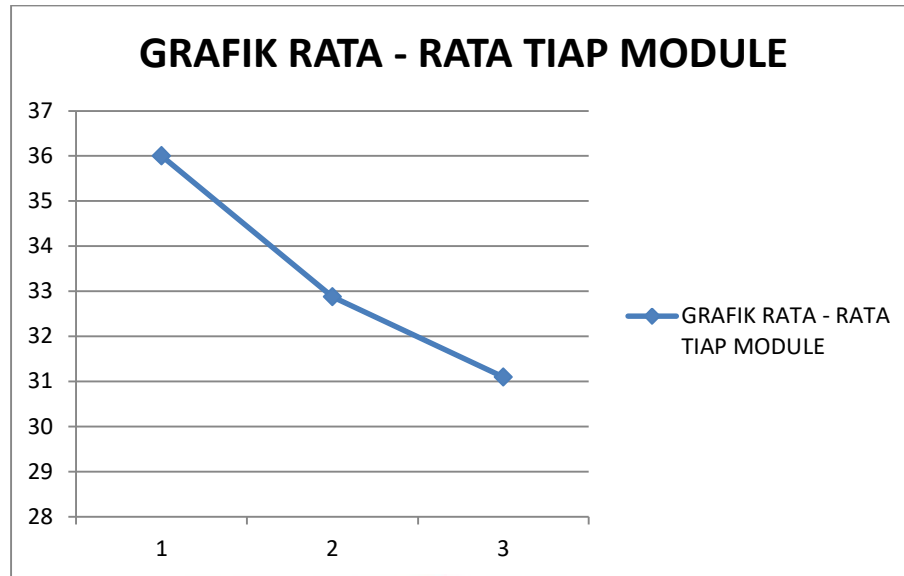
Pembagi 15 juta didapat pada datasheet yang mana merupakan life mechanical limit total pada *relay* jenis HRS2H-S-DC24V. Jadi, kita dapat lebih mudah untuk memantau berapa persen tiap riap *relay*, sehingga apabila *relay* telah mendekati life mechanical limit operasional sebaiknya kita segera memesan *relay* tersebut dan mengganti *relay* tersebut sebelum mengalami failure yang akan berakibat pada kerusakan alat *switching unit* maupun komponen yang akan dilakukan pengetesan.



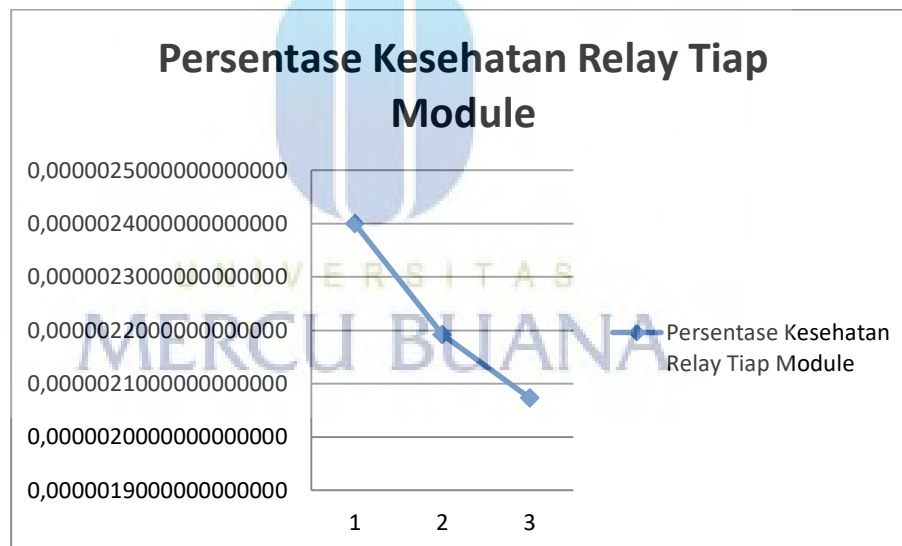
Gambar 4. 11 Grafik Masing - Masing *Relay* Pada Setiap *Module*



Gambar 4. 12 Grafik Persentase Masing - Masing *Relay* Pada Setiap *Module*



Gambar 4. 13 Grafik Rata – Rata Setiap Module Pada Alat *Switching Unit*



Gambar 4. 14 Grafik Persentase Rata – Rata Setiap Module Pada Alat *Switching Unit*

Pada gambar grafik di atas, menjelaskan seberapa besar persentase kesehatan masing masing relay.

Dengan mengetahui kesehatan pada tiap – tiap relay, kita dapat menghindari kerusakan pada alat *switching unit* maupun komponen yang akan dilakukan pengetesan, menghindari terhambatnya operasional akibat alat rusak,

serta dapat menghindari pemesanan relay baru yang mana estimasi waktu kedatangan yang lama.

4.6 Simulasi Korelasi Antara *Self Test* Dengan Alat Pemantau Kesehatan

Relay

Pada sub bab ini, kita akan mensimulasikan hubungan atau korelasi antara *self test* pada *switching unit* dengan data dari alat pemantau kesehatan *relay*, yang mana keduanya saling berhubungan satu sama lainnya.

Disini hanya simulasi data apabila antara *self test* pada *switching unit* terdapat hasil yang menyatakan *fail* pada *relay* yang berada di *switching unit*, maka kita lakukan observasi data pada alat pemantau kesehatan *relay*. Apakah *relay* yang *fail* tersebut telah digunakan terlalu sering saat operasional atau tidak.



Tabel 4. 12 Tabel Hasil *Self Test* Alat *Switching Unit*

SELF TEST REPORT			
	MODULE 1	MODULE 2	MODULE 3
RELAY 1	PASS	PASS	PASS
RELAY 2	PASS	PASS	PASS
RELAY 3	PASS	FAIL	PASS
RELAY 4	PASS	PASS	PASS
RELAY 5	PASS	FAIL	PASS
RELAY 6	FAIL	PASS	PASS
RELAY 7	PASS	PASS	PASS
RELAY 8	PASS	PASS	PASS
RELAY 9	PASS	PASS	FAIL
RELAY 10	PASS	PASS	PASS
RELAY 11	PASS	PASS	PASS
RELAY 12	PASS	PASS	PASS
RELAY 13	PASS	PASS	FAIL
RELAY 14	FAIL	PASS	PASS
RELAY 15	FAIL	FAIL	PASS
RELAY 16	PASS	PASS	PASS
RELAY 17	PASS	PASS	PASS
RELAY 18	PASS	PASS	PASS
RELAY 19	PASS	PASS	PASS
RELAY 20	PASS	PASS	PASS
RELAY 21	PASS	PASS	PASS
RELAY 22	PASS	PASS	PASS
RELAY 23	PASS	PASS	PASS
RELAY 24	PASS	PASS	PASS
RELAY 25	PASS	PASS	PASS
RELAY 26	PASS	PASS	PASS
RELAY 27	PASS	PASS	PASS
RELAY 28	PASS	PASS	PASS
RELAY 29	PASS	PASS	PASS
RELAY 30	PASS	PASS	PASS
RELAY 31	PASS	PASS	PASS
RELAY 32	PASS	PASS	PASS

Tabel 4. 13 Tabel Pemantau Kesehatan Pada Alat *Switching Unit*

COUNT OF MODULE USED			
	MODULE 1	MODULE 2	MODULE 3
RELAY 1	56	54	52
RELAY 2	55	34	37
RELAY 3	62	63	47
RELAY 4	49	45	46
RELAY 5	49	62	49
RELAY 6	73	47	59
RELAY 7	49	50	43
RELAY 8	47	43	57
RELAY 9	56	42	58
RELAY 10	46	42	41
RELAY 11	68	38	52
RELAY 12	60	30	42
RELAY 13	59	49	59
RELAY 14	79	54	33
RELAY 15	75	64	32
RELAY 16	18	18	50
RELAY 17	18	31	16
RELAY 18	11	9	5
RELAY 19	21	28	22
RELAY 20	20	17	1
RELAY 21	17	31	13
RELAY 22	32	12	15
RELAY 23	11	2	10
RELAY 24	17	26	1
RELAY 25	2	12	14
RELAY 26	26	23	2
RELAY 27	11	19	32
RELAY 28	5	22	4
RELAY 29	3	7	26
RELAY 30	6	28	15
RELAY 31	23	22	31
RELAY 32	28	28	31
RATE	36	32,875	31,09375
RATE MODULE GENERAL	33,32291667		

Keterangan :

- = Penggunaan relay tersebut dibawah rata-rata operasional
- = Penggunaan relay tersebut diatas rata-rata operasional

Pada **Tabel 4. 14** di atas, menjelaskan bahwa terdapat beberapa *relay* yang hasilnya adalah *fail* saat dilakukan *self test* pada *switching unit*. Dan pada **Tabel 4. 13** memaparkan seberapa banyak setiap *relay* pada *switching unit* telah digunakan saat operasional.

Saat kita lakukan perbandingan data, ketika beberapa *relay* yang hasilnya *fail* ternyata memiliki hasil penggunaan yang lebih besar daripada *relay* yang lainnya. Hal ini menyatakan, semakin besar penggunaan *relay* pada *switching unit* saat operasional, pastinya akan berdampak pada *relay* itu sendiri dengan mengurangi *lifetime* dari *relay* itu sendiri. Dengan adanya data untuk perbandingan tersebutlah yang ingin dicapai, agar alat pemantau kesehatan *relay* dapat menghindari terhambatnya proses operasional pada electrical workshop. Dengan mengetahui kesehatan pada *relay*, kita dapat membuat estimasi kapan harus memesan *relay* baru untuk mengganti *relay* yang rusak tanpa menghambat proses operasional pada electrical workshop.

