BAB IV
HASIL DAN PEMBAHASAN

Setelah perancangan dan pembuatan perangkat keras dan perangkat lunak terselesaikan, maka pada Bab 4 ini penulis akan membahas data hasil pengujian dan analisa berdasar pengamatan perangkat keras dan perangkat lunak yang telah dirancang.

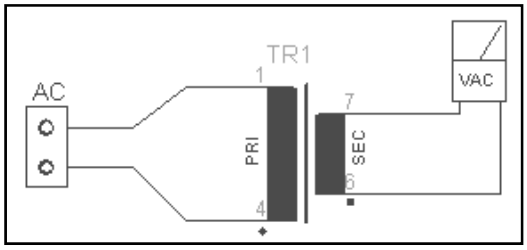
4.1 Pengujian dan Analisa Perangkat Keras

4.1.1 Pengujian Rangkaian Catu Daya

Pengujian rangakaian catu daya sangat penting untuk mengetahui kualitas sumber tegangan yang dihasilkan oleh catu daya, sumber tegangan ini akan digunakan untuk men-*supply* perangakat keras lainnya. Pengukuran rangkaian catu daya dilakukan dengan beberapa tahap yaitu sebagai berikut:

a. Pengukuran tegangan AC pada kumparan sekunder transformator.

   Kumparan sekunder akan menghasilkan tegangan AC, tegangan ini merupakan tegangan induksi yang dihasilkan oleh kumparan primer jika kumparan primer dihubungkan dengan sumber tegangan PLN. Pada Gambar 4.1 merupakan pengukuran tegangan pada kumparan sekunder transformator.



Gambar 4.1. Pengukuran Tegangan Sekunder Transformator

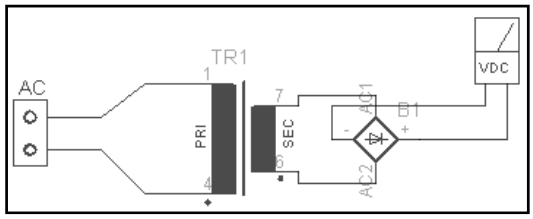
Berdasar data hasil pengukuran didapat tegangan sekunder sebesar 6,3 Volt, tegangan ini merupakan tegangan rata-rata atau disebut dengan *Voltage Root Mean Square* (VRMS). Dengan demikian tegangan puncak (V_{peak}) dapat ditentukan dengan cara sebagai berikut:

$$V_{peak} = \frac{VRMS}{0.7}$$
$$= 6.3V/0.7$$
$$= 9V$$

Maka tegangan puncak yang dihasilkan oleh kumparan sekunder transformator yaitu 9 Volt.

b. Pengukuran tegangan DC setelah melewati dioda *bridge*.

Penggunaan dioda penyearah yang disusun dengan metode *bridge* bertujuan untuk memperoleh sinyal gelombang penuh (*full wave*). Pada Gambar 4.2 merupakan pengukuran tegangan DC setelah melewati dioda *bridge*.



Gambar 4.2. Pengukuran Tegangan DC

Berdasar data hasil pengukuran diperoleh tegangan DC sebesar 5.49 Volt, data hasil pengukuran ini menandakan terjadi penurunan tegangan pada dioda *bridge* sebesar:

$V_d = V_{in} - V_o$

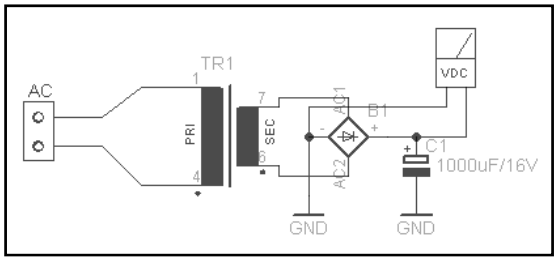
$V_d = 6.3V - 5.49V$

$= 0.81V$

Berdasar hasil perhitungan secara matemais penurunan tegangan pada dioda sebesar 0.81V mendekati tegangan ideal *forward* dioda Germanium yaitu 0.70V.

c. Pengukuran tegangan DC dengan tapis kapasitor.

Untuk memperoleh tegangan DC rata maka digunakan sebuah tapis (*filter*) berupa kapasitor. Pada Gambar 4.3 merupakan pengukuran tegangan DC setelah menggunakan tapis kapasitor.

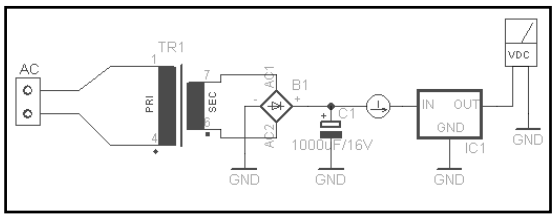


Gambar 4.3. Pengukuran DC dengan Tapis Kapasitor

Data hasil pengukuran tegangan DC setelah menggunankan tapis kapasitor sebesar 8V, hal ini menandakan bahwa tegangan DC mendekati tegangan puncak yang dihasilkan oleh kumparan sekunder transformator.

d. Pengukuran tegangan DC setelah IC regulator.

Rangkaian catu daya bertujuan untuk menghasilkan tegangan DC 5 Volt, maka digunakan IC regulator 7805 agar tegangan yang dihasilkan oleh rangkaian catu daya sebesar 5 Volt. Pada Gambar 4.4 merupakan pengukuran tegangan yang dihasilkan oleh IC 7805.



Gambar 4.4. Pengukuran Tegangan DC pada IC Regulator

Hasil pengukuran tegangan keluaran pada IC 7805 yaitu 5.02 Volt, tegangan yang dihasilkan mengalami kesalahan sebesar 0.02V, besar kesalahan tegangan ini dapat ditoleransi oleh perangkat TTL. Selain pengukuran tegangan, pada tahap ini juga dilakukan pengukuran arus yang diperlukan oleh IC regulator. Berdasar pengukuran arus yang diperlukan oleh IC regulator yaitu 3.74mA. Dengan demikian besar beban (RL) IC regulator dapat ditentukan dengan cara sebagai berikut:

$$RL = \frac{V}{I}$$

$$RL = \frac{8V}{3.74mA}$$

$$= 2139\ Ohm$$

Setelah melakukan pengukuran dan perhitungan secara matematis maka rangkaian catu daya yang dirancang diperkirakan memiliki persenatase tegangan *ripple* (Vr) sebesar:

$$V_r = \frac{V_{dc}}{2fCRL}$$

Di mana nilai unsur-unsurnya adalah sebagai berikut:

$V_{dc} = 8V$

$f = 50Hz$

$C \quad = 1000uF$

$RL \ = 2139\ Ohm$

$$V_r \quad = \frac{8V}{2 \times 50Hz \times 1000uF \times 2139\ Ohm}$$

$$= 0.037V$$

$$\%Vr = \frac{0.073V}{8V} \times 100$$

=0.4625 %

Berdasar hasil pengukuran dan perhitungan secara matematis dapat disimpulkan bahwa perangkat keras catu daya bekerja sesuai dengan fungsinya yaitu sebagai sumber tegangan DC 5V dan aman digunakan untuk sumber tegangan yang diperlukan oleh perangkat lainnya, hal ini dikeranakan persentase tegangan *ripple* berdasar perhitungan lebih kecil dari 1%.

4.1.2 Pengujian Rangkaian Serial RS-232

Pengujian terhadap perangkat keras rangkaian komunikasi serial RS-232 sangat penting untuk mengetahui kemampuan komunikasi serial yang menggunakan IC MAX232, jika komunikasi serial berjalan dengan baik maka rangkaian RS-232 sudah benar. Sebaliknya jika komunikasi serial berjalan kurang baik, kemungkinan rangkaian RS-232 belum dirangkai dengan benar atau kondisi IC MAX232 dalam keadaan kurang baik atau rusak.

Untuk menguji komunikasi serial dari komputer ke perangkat luar dalam hal ini yaitu rangakaian RS-232 diperlukan perangkat lunak untuk mengirim dan menerima data secara serial. Adapun perangkat lunak yang akan digunakan yaitu perangkat lunak Hyper Terminal yang terdapat pada sistem operasi Windows XP, perangkat lunak ini sudah ter-*install* secara otomatis pada saat melakukan instalasi sistem operasi Windows XP di komputer, maka perangkat lunak ini dapat langsung digunakan.

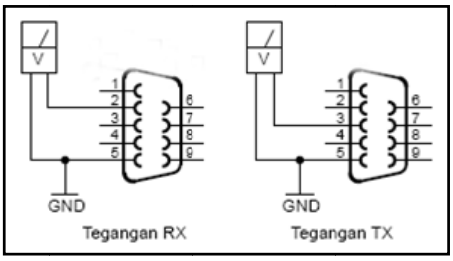
Pengujian komunikasi serial RS-232 yang akan dilakukan meliputi pengujian kondisi *port* serial komputer dan kondisi rangkaian RS-232. Adapun cara pengujian komunikasi serial RS-232 adalah sebagai berkut:

1. Pengamatan port serial komputer

   Pada pengujian *port* serial komputer bertujuan untuk mengetahui kondisi *port* serial komputer dalam keadaan baik atau dalam keadaan kurang baik, maka untuk memastikan kondisi *port* serial komputer dilakukan dengan cara mengukur tegangan *port* serial dan mengonfigurasi *port* serial untuk mengirim dan menerima data dari/ke komputer itu sendiri atau disebut juga dengan metode *loopback*.

   a. Pengukuran tegangan pada konektor DB-9 komputer.

      Pengukuran tegangan pada *port* serial komuputer yaitu dengan mengukur pin penerima (RX) yang terletak pada pin 2 dan pin pengirim (TX) yang terletak pada pin 3 dari konektor DB-9 komputer. Pada Gambar 4.5 merupakan cara pengukuran tegangan konektor DB-9 komputer.



Gambar 4.5. Pengukuran Tegangan pada Konektor DB-9 Komputer

Data hasil pengukuran tegangan pada konektor DB-9 *port* serial personal komputer seperti terlihat pada Tabel 4.1.

Tabel 4.1. Data Hasil Pengukuran Tegangan pada Konektor DB-9 Komputer

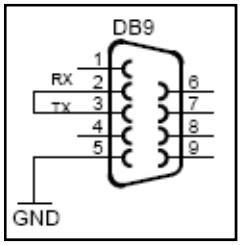
No	Bit perdetik (bps)	Port (Open/Close)	V(TX) (Volt)	V(RX) (Volt)
1	1200	Close	-11.45	0.08
		Open	-11.45	0.08
2	2400	Close	-11.45	0.08
		Open	-11.45	0.08
3	4800	Close	-11.45	0.08
		Open	-11.45	0.08
4	9600	Close	-11.45	0.08
		Open	-11.45	0.08
5	19200	Close	-11.45	0.08
		Open	-11.45	0.08

Berdasar hasil pengukuran pada Tabel 4.1 dapat disimpulkan bahwa *port* serial komputer pada kondisi *close* (tertutup) maupun *open* (terbuka) pin RX menghasilkan tegangan sebesar 0.08V, dan pin TX menghasilkan tegangan sebesar -11.45V.

b. Kirim dan terima data serial pada konektor DB-9 komputer.

Kondisi tegangan pada pin TX dari hasil pengukuran mengeluarkan tegangan sebesar -11.45V mengindikasikan bahwa bit yang dikeluarkan adalah satu (1), sedangkan pada pin RX mengeluarkan tegangan sebesar 0.08V, kondisi ini tidak dapat terbaca oleh *port* serial sebagai bit 0 maupun bit 1 karena tegangan pada pin RX berada pada *range* tidak terdefinisi.

Untuk dapat menguji pengiriman dan penerimaan data pada *port* serial yaitu dengan cara menghubung singkat pin TX dengan pin RX atau disebut dengan metode *loopback*, dengan demikian tegangan pada pin RX sama dengan tegangan pada pin TX, hal ini akan membuat kondisi pin RX dapat terbaca oleh *port* serial. Pada Gambar 4.6 merupakan pengkawatan pin RX dengan pin TX yang dihubung singkat.



Gambar 4.6. Pengkawatan Konektor DB-9 Komputer

Sumber data yang akan dikirimkan secara serial berasal dari penekanan *keyboard* yang dipresentasikan dalam bentuk kode ASCII sehingga data yang ditampilkan pada jendela Hyper Terminal juga dalam kode ASCII. Pada Tabel 4.2 merupakan data hasil pengiriman data serial.

Tabel 4.2. Data Hasil Pengamatan Kirim Terima Data Pada Port Serial

No	Baud Rate (bps)	Kirim (Keyboard)	Terima (Hyper Terminal)
1	1200	ABCDE01234	ABCDE01234
2	2400	ABCDE01234	ABCDE01234
3	4800	ABCDE01234	ABCDE01234
4	9600	ABCDE01234	ABCDE01234
5	19200	ABCDE01234	ABCDE01234

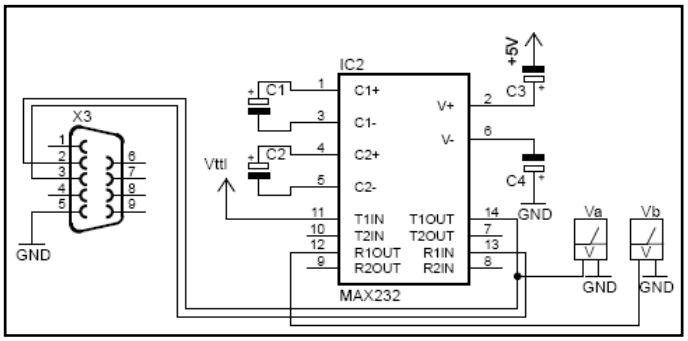
Dari data hasil pengujian dapat disimpulkan bahwa pengiriman dan penerimaan data pada *port* serial komputer bekerja dengan baik dengan pengaturan kecepatan sinyal mulai dari 1200bps sampai dengan 19200bps.

2. Pengamatan rangakaian RS-232 level TTL

   Setelah mengetahui kondisi *port* serial bekerja dengan baik, selanjutnya adalah menguji rangkaian RS-232, rangkaian ini digunakan untuk mengonversi RS-232 level EIA menjadi level TTL. Adapun pengujian serial RS-232 dijelaskan sebagai berikut:

   a. Pengukuran tegangan pada rangakaian RS-232.

      Pada pengujian rangkaian RS-232 yaitu dengan cara mengukur tegangan *intput output* pada pin IC MAX232 yang digunakan sebagai pengirim dan penerima data serial. Pada Gambar 4.7 merupakan pengukuran tegangan pin *input output* IC MAX232.



Gambar 4.7. Pengukuran Tegangan Input Output IC MAX232

Adapun data hasil pengukuran tegangan untuk rangkaian RS-232 seperti terlihat pada Tabel 4.3.

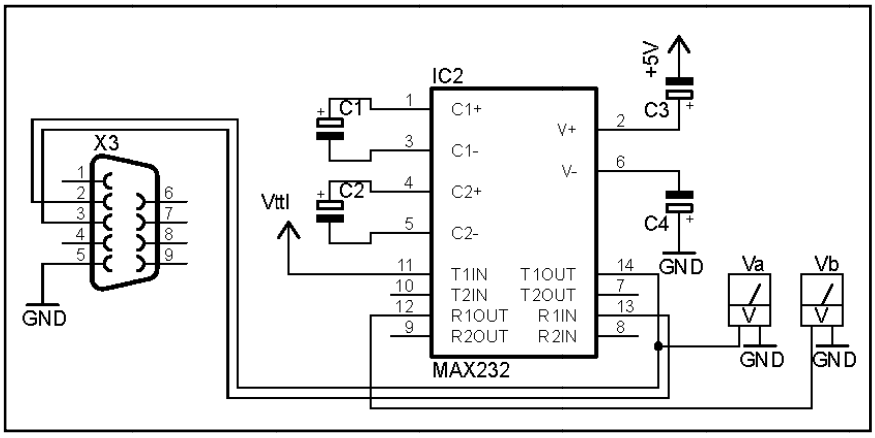
Tabel 4.3. Data Hasil Pengukuran Tegangan pada IC MAX232

No	Vttl (Volt)	Va (Volt)	Vb (Volt)
1	5	-7	5
2	0	8.2	5

Dari data hasil pengukuran pada Tabel 4.3, pada saat pin T1IN (Vttl) sama dengan 5V maka keluaran pada T1OUT (Va) menghasilkan tegangan sebesar -7V, tegangan level ini dapat dikenal sebagai bit 1 (*high*) oleh personal komputer. Sedangkan pada kondisi T1IN jika diberikan 0V, maka keluaran pada T1OUT (Va) menghasilkan tegangan sebesar 8.2V yang akan dibaca personal komputer sebagai bit 0 (*low*). Untuk pengukuran pada Vb merupan kondisi keluaran pin R1OUT yang menghasilkan tegangan sebesar 5V pada saat tegangan pada pin R1IN sebesar -11.45Volt, maka level tegangan R1OUT pada kondisi ini akan dibaca bit 1 (*high*) oleh perangkat penerima serial level TTL, dikarenakan personal komputer tidak mengirimkan data apapun maka nilai vb tetap 5V.

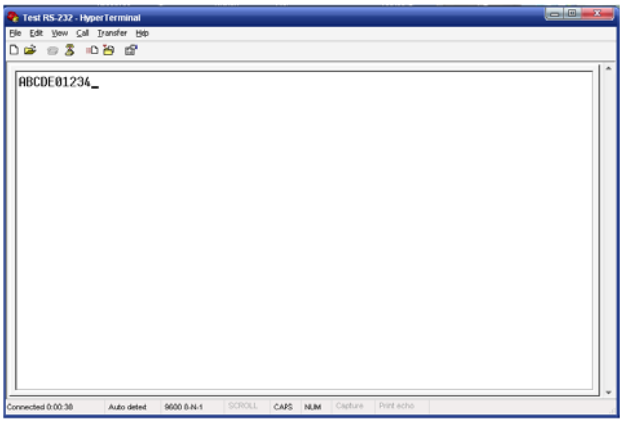
b. Pengamatan kirim dan terima data serial pada rangkaian RS-232.

Kondisi tegangan *input output* pada IC MAX232 telah memenuhi syarat untuk melakukan antarmuka komputer dengan perangkat luar yang memiliki tegangan level TTL. Untuk menguji fungsi kerja rangkaian RS-232 sebagai pengirim dan penerima data serial maka pin TXD dengan pin RXD harus dihubung singkat seperti pada pengujian *port* serial komputer. Pada Gambar 4.8 merupakan pin IC pada MAX232 yang dihubung singkat.



Gambar 4.8. Pengkawatan Komunikasi Serial IC MAX232

Pengujian kemampuan rangkaian RS-232 dapat diketahui dengan menggunakan program Hyper Terminal. Pada Gambar 4.9 merupakan hasil pengujian rangkaian RS-232 menggunakan Hyper Terminal.

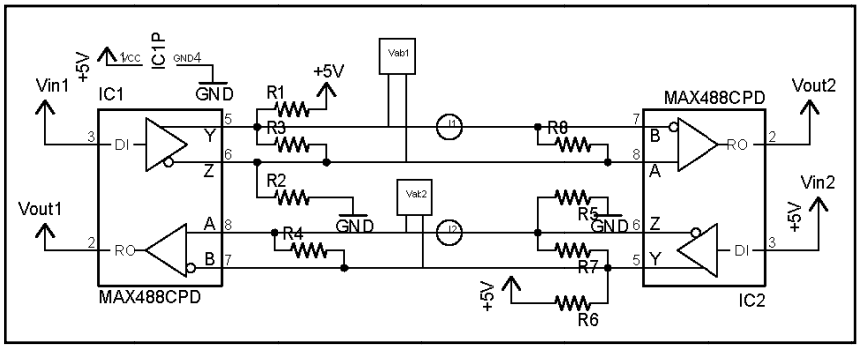


Gambar 4.9 Hasil Pengamatan Komunikasi Serial RS-232

Dari hasil pengamatan rangkaian RS-232 level TTL dapat disimpulkan bahwa rangakaian serial RS-232 dapat menerima data dan mengirimkan kembali data tersebut dengan benar meskipun karakteristik tegangan yang dihasilkan oleh IC MAX232 tidak memenuhi tegangan maksimum standar EIA/TIA yaitu -15V sampai dengan +15V.

4.1.3 Pengujian Rangkaian Serial RS-485

Pada subab ini penulis akan menjelaskan hasil pengujian dan pengamatan pada rangakaian RS-485. dilakukan dengan cara memberikan tegangan *input* pada pengirim dengan logika 1 dan logika 0 untuk mengetahui kondisi *output* pada penerima. Pada Gambar 4.10 merupakan titik pengukuran tegangan dan arus pada rangkaian RS-485.



Gambar 4.10. Pengukuran Tegangan Pada Rangkaian RS-485

Data hasil pengukuran tegangan pada rangkaian RS-485 seperti terlihat pada Tabel 4.4.

Tabel 4.4 Data Hasil Pengukuran pada Rangkaian RS-485

No	Vin		V_{YZ} (Volt)	I (mA)	Vout	
1	Vin1	5V	2.06	33.2	Vout2	5V
2		0V	-2.05	-33.4		0V
3	Vin2	5V	2.04	33.7	Vout1	5V
4		0V	-2.05	-33.4		0V

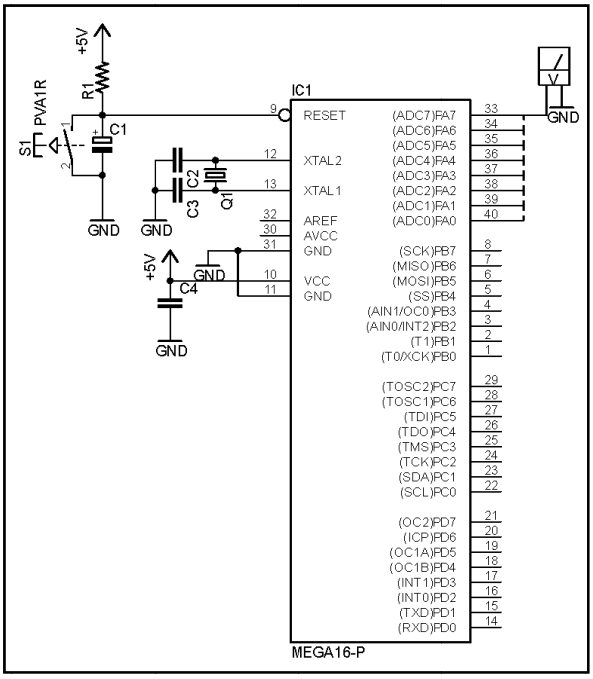
Dari hasil pengamatan dan pengukuran tegangan pada rangkaian RS-485, dapat disimpulkan bahwa pengiriman data untuk logika 1 pada level TTL dikonversi dengan perbedaan tegangan antara terminal Y dengan terminal Z pada sisi pengirim adalah 2.06 Volt, sedangkan untuk logika 0 memiliki perbedaan tegangan adalah -2.05 Volt, kondisi ini menggambarkan tegangan RS-485 telah memenuhi syarat yang distandardisasikan oleh EIA/TIA yaitu perbedaan tegangan minimal adalah 0.2 Volt.

4.1.4 Pengujian Rangkaian ATmega16

Fungsi *port* mikrokontroler ATmega 16 sebagai *input* harus memiliki karakteristik tegangan *high* (1) atau *low* (0), sedangkan fungsi *port* sebagai *output* harus dapat men-*drive* perangkat lainnya, pada perancangan ini *port output* digunakan untuk men-*drive* relay. Karakterisitik tegangan *input output* pada masing-masing *port* sangat penting diketahui, oleh karena itu Penulis melakukan pengukuran tegangan dan arus pada *port input output* yang akan digunakan. Adapun pengukuran tegangan dan arus pada masing-masing *port* dijelaskan sebagai berikut.

a. Port input.

   Port yang digunakan sebagai *input* adalah *port* A, cara pengukuran tegangan dan arus pada *port* ini dilakukan seperti terlihat pada Gambar 4.11.



Gambar 4.11. Pengukuran Tegangan pada Port A Sebagai Input

Agar *port* A berfungsi sebagai *input* maka perlu dikonfigurasi register *port* A yang dilakukan oleh *software*, potongan program di bawah ini berfungsi untuk mengonfigurasi *port* A sebagai *input*.

```
LDI     R16,0X00
OUT     DDRA,R16
LDI     R16,0XFF
OUT     PORTA,R16
```

Dari potongan program di atas maka didapat hasil pengukuran pada *port* A seperti terlihat pada Tabel 4.5.

Tabel 4.5. Data Hasil Pengukuran Tegangan Port A

No.	PORT	Terukur	
		Volt	mA
1	PA0	4.48	0.12
2	PA1	4.47	0.12
3	PA2	4.47	0.12
4	PA3	4.42	0.12
5	PA4	4.42	0.11
6	PA5	4.46	0.11
7	PA6	4.50	0.12
8	PA7	4.42	0.11

Berdasar hasil pengukuran pada Tabel 4.5 besar tegangan *output* kondisi *high* pada *port* A lebih besar dari keterangan tegangan pada lembar data

ATmega16, maka tegangan ini dapat dibaca sebagai level *high*. Sedangkan hasil pengukuran besar arus dapat digunakan untuk menghitung besar nilai resistansi *pull up* internal yang dimiliki oleh *port* A dengan persamaan sebagai berikut:

$$R_P = \frac{V}{I}$$

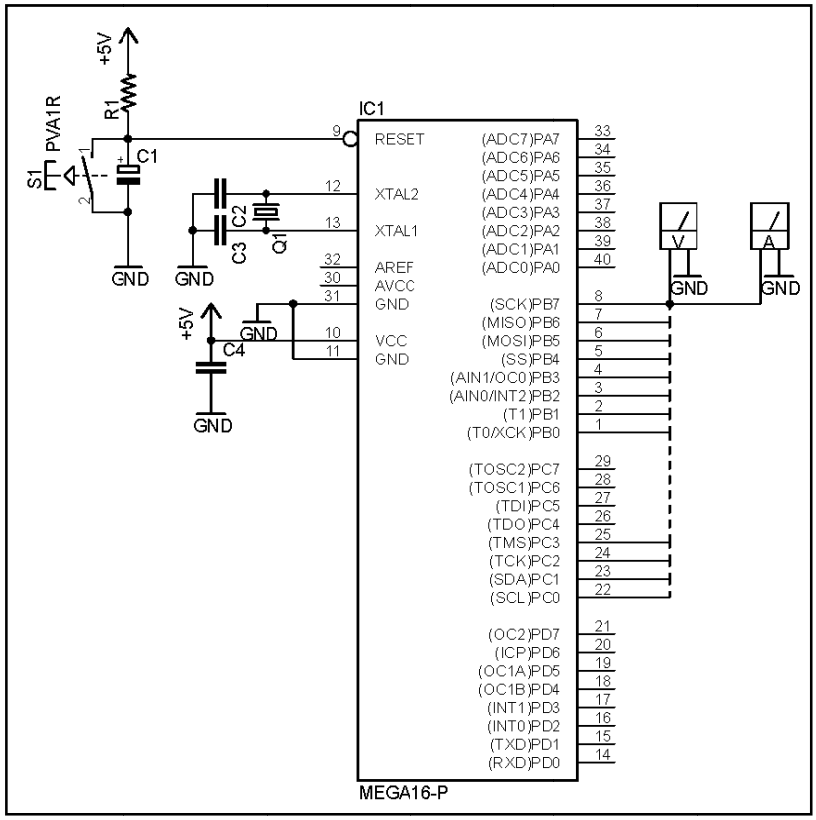
$$= \frac{4.47V}{0.1mA}$$

$$= 37.25\ K\ Ohm$$

Besar nilai resistansi resistor *pull up* internal berada pada range antara 20 K Ohm dan 50 K Ohm, kondisi ini sesuai dengan keterangan pada lembar data ATmega16.

b. Port output.

Untuk *port output* yaitu menggunakan *port* B dan *nibble low* pada *port* C, pengukuran tegangan pada *port* ini dilakukan dilakukan dengan cara seperti terlihat pada Gambar 4.12.



Gambar 4.12. Pengukuran Tegangan pada Port B dan Port C Sebagai Output

Agar *port* B dan *port* C dapat berfungsi sebagai *output* maka perlu dikonfigurasi register *port* B dan *port* C oleh *software*. Pengukuran dilakukan dengan dua kondisi yaitu *port* pada saat kondisi *output high* dan pada saat kondisi *low*. Potongan program berikut berfungsi untuk mengonfigurasi *port* B dan *port* C sebagai *output* dengan kondisi *high*.

```
LDI     R16,0B11111111
OUT     DDRB,R16
OUT     PORTB,R16
LDI     R16,0B00001111
OUT     DDRC,R16
OUT     PORTC,R16
```

Dari potongan program tersebut maka didapat hasil pengukuran tegangan dan arus pada *port* B dan *port* C seperti terlihat pada Tabel 4.6.

Tabel 4.6. Data Hasil Pengukuran Tegangan pada Port B dan Port C

Kondisi Output High

No.	PORT	Terukur	
		V	mA
1	PB0	4.84	71.2
2	PB1	4.84	71.6
3	PB2	4.82	71.6
4	PB3	4.85	71.6
5	PB4	4.78	71.5
6	PB5	4.84	71.6
7	PB6	4.82	71.6
8	PB7	4.79	71.6
9	PC0	4.84	70.8
10	PC1	4.84	70.6
11	PC2	4.82	70.4
12	PC3	4.82	70.4

Berdasar data hasil pengukuran pada tabel 4.6 maka nilai resistor pull up internal dengan kondisi output high dapat ditentukan dengan perhitungan matematis sebagai berikut:

$$R_P = \frac{V}{I}$$

$$= 4.84/71.6mA$$

$$= \ 67.6\ K\ Ohm.$$

Nilai resistansi resistor *pull up* internal sebesar 67.6 K Ohm, kondisi ini dikarenakan pada *port* dikonfigurasi sebagai *output* yang menghasilkan tegangan mendekati tegangan VCC yaitu 5 V.

Pengukuran tegangan *port* B dan *port* C dilakukan dengan cara mengubah potongan program *port output* untuk kondisi *high* menjadi kondisi *low*, potongan program berikut adalah cara untuk menghasilkan *port* dalam kondisi *output low*.

```
LDI     R16,0B11111111
OUT     DDRB,R16
LDI     R16,0B00000000
OUT     PORTB,R16
LDI     R16,0B00001111
OUT     DDRC,R16
LDI     R16,0B00000000
OUT     PORTC,R16
```

Adapun data hasil pengamatan *port* B dan *port* C dengan *output low* seperti terlihat pada Tabel 4.7.

Tabel 4.7. Data Hasil Pengukuran Tegangan pada Port B dan Port C Kondisi Output Low

No.	PORT	Terukur	
		Volt	mA
1	PB0	0.004	0.11
2	PB1	0.003	0.12
3	PB2	0.004	0.11
4	PB3	0.004	0.11
5	PB4	0.004	0.08
6	PB5	0.005	0.12
7	PB6	0.004	0.11
8	PB7	0.004	0.11
9	PC0	0.002	0.10
10	PC1	0.004	0.11
11	PC2	0.004	0.11
12	PC3	0.004	0.11

Berdasar hasil pengukuran pada Tabel 4.8, kondisi port yang dikonfigurasi sebagai output dengan kondisi low, maka nilai resistansi resistor pull up dapat ditentukan dengan persamaan sebagai berikut:

$$R_P = \frac{V}{I}$$

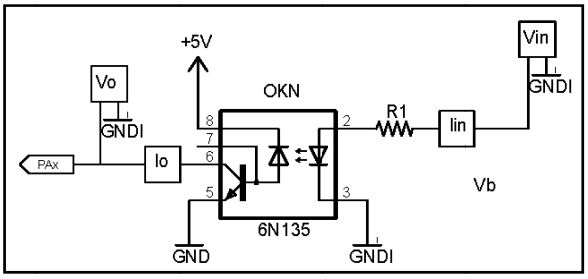
$$= \frac{0.004V}{0.11mA}$$

$= 36.36\ K\ Ohm$

Simpulan dari hasil pengamatan port *input/output* mikrokontroler ATmega16 yaitu nilai resistansi resistor *pull up* internal dapat berubah sesuai dengan konfigurasi register DDR.

4.1.5 Pengujian Rangkaian Optocoupler 6N135

Sesuai dengan perancangan fungsi *optocoupler* sebagai sakelar logika, maka pengamatan dilakukan dengan cara mengukur tegangan *output* pada pin kolektor untuk logika *high* (1) dan logika *low* (0). Pada Gambar 4.13 merupakan pengukuran tegangan *output* pada *optocoupler*.



Gambar 4.13. Pengukuran Tegangan Output Optocoupler 6N135

Adapun data hasil pengukuran tegangan *output* pada *optocoupler* seperti terlihat pada Tabel 4.8.

Tabel 4.8. Data Hasil Pengukuran Tegangan Ouput Optocoupler 6N135

No	Vin (Volt)	Iin (mA)	Vo (Volt)	Io (mA)
1	0	0	4.46	0
2	24	10.18	0.14	0.2

Dari data hasil pengukuran pada Tabel 4.8 maka besar tegangan yang tertahan oleh R1 adalah sebagai berikut:

$V_{R1} = V_{in} - V_F$

Di mana nilai unsur-unsurnya yaitu sebagai berikut:

$Vin = 24V$

$V_F = 1.3V\ (datasheet)$

$V_{R1} = 24V - 1.3V$

$= 22.7\ Volt$

Besar persentase nilai CTR (*Current Transfer Ratio*) adalah sebagai berikut:

$$IF = \frac{I_o}{CTR}$$

$$\%CTR = \frac{I_o}{I_F} x100$$

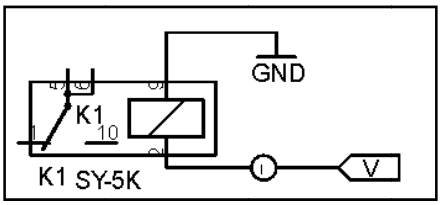
$$\%CTR = \frac{0.2mA}{10.18mA} x100$$

$$CTR = 1.96\%$$

Dari hasil pengujian *optocoupler* maka kondisi *optocoupler* dalam keadaan baik dan dapat difungsikan sebagai sakelar logika.

4.1.6 Pengujian Rangkaian Relay SY-5-K

Pengujian relay SY-5-K yaitu dengan memberikan tegangan pada kumparan dan pengamatan plat kontak pada relay. Pada Gambar 4.14 merupakan pengukuran tegangan pada relay SY-5-K.



Gambar 4.14. Pengukuran Relay SY-5-K

Adapun hasil pengamatan kondisi relay seperti terlihat pada Tabel 4.9.

Tabel 4.9. Data Hasil Pengamatan Relay SY-5-K

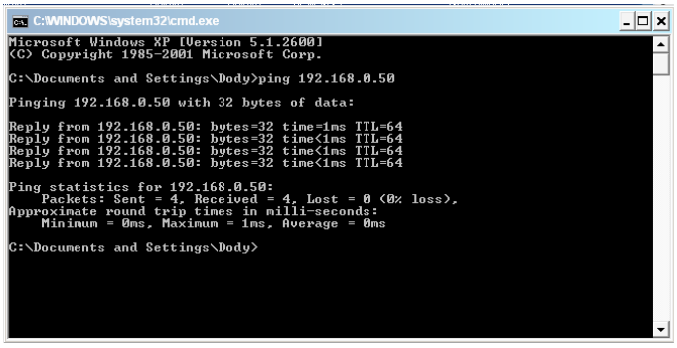
No	V	I	Perubahan Kontak
1	4.46V	24.9mA	NC→NO
2	0.004V	0mA	NO→NC

Dari data hasil pengamatan pada Tabel 4.9 maka kondisi relay dalam keadaan baik dan berkerja sesuai dengan fungsinya sebagai sakelar elektromagnetik.

4.1.7 Pengujian Acces Point

Pengujian perangkat keras *access point* yaitu dengan cara melakukan *pinging* IP *access point* dengan menggunakan personal komputer. Nomor IP pada *access point* D-LINK DWL-2100AP adalah 192.168.0.50 dan *subnetmask* 255.255.255.0 maka konfigurasi IP pada komputer harus dalam *range* IP yang dimiliki oleh *access point*, dalam pengujian ini TCP/IP komputer yaitu

192.168.0.1. Pada Gambar 4.15 merupakan hasil pengujian koneksi personal komputer dengan *access point*.

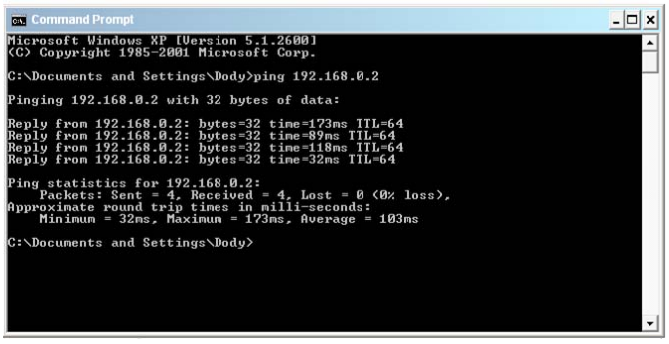


Gambar 4.15. Hasil Pinging IP Access Point

Dari hasil pengujian *access point*, maka kondisi *access point* dalam keadaan baik dan dapat terhubung dengan komputer.

4.1.8 Pengujian Tablet PC

Pengujian koneksi perangkat tablet PC dengan personal komputer yaitu dengan *pinging* IP seperti pada pengujian *acess point*. TCP/IP pada tablet PC sebelumnya dikonfigurasi dengan *range* IP sesuai dengan *range* IP komputer. Pada pengujian ini tablet PC memiliki IP *Address* 192.168.0.2. Pada Gambar 4.16 merupakan hasil *ping* IP dari personal komputer ke tablet PC.



Gambar 4.16. Hasil Pinging IP Tablet PC

Dari hasil pengamatan, Tablet PC dapat terkoneksi dengan personal komputer melalui media transmisi *wireless*.

4.2 Analisa Perangkat Lunak

Setelah melakukan pengujian terhadap perangkat keras, pada subab ini akan menjelaskan mengenai perangkat lunak sistem. Penjelasan perangkat lunak hanya mencakup subprogram untuk program utama.

4.2.1 Program CCR Simulator

Program CCR Simulator memiliki subprogram untuk mengonfigurasi *port input output* dan *port* komunikasi serial. Adapun subprogram tersebut yaitu sebagai berikut:

a. Konfigurasi port sebagai input dan output.

```
 1  LDI    R16,0x00
 2  OUT    DDRA,R16
 3  LDI    R16,0xff
 4  OUT    PORTA,R16
 5  LDI    R16,0x00
 6  OUT    DDRD,R16
 7  LDI    R16,0b11110000
 8  OUT    PORTD,R16
 9  LDI    R16,0xff
10  OUT    DDRB,R16
11  LDI    R16,0x00
12  OUT    PORTB,R16
13  LDI    R16,0x0f
14  OUT    DDRC,R16
15  LDI    R16,0x00
16  OUT    PORTC,R16
```

Dari potongan program di atas dapat dijelaskan yaitu, baris program ke-1 sampai dengan baris ke-4 digunakan untuk konfigurasi *port* A sebagai *input*. Pada baris program ke-5 sampai dengan baris ke-8 digunakan untuk konfigruasi *port* D, dimana *nibble high* digunakan sebagai *input*. Program pada baris ke-9 sampai dengan baris ke-16 digunakan untuk konfigurasi *port* B dan *nibble port* C sebagai *output*.

b. Konfigurasi register komunikasi serial.

```
 1  LDI    R16,HIGH(UBRRVAL)
 2  OUT    UBRRH,R16
 3  LDI    R16,LOW(UBRRVAL)
 4  OUT    UBRRL,R16
 5  LDI     R16,(1<<RXEN)|(1<<TXEN)|(1<<RXCIE)
 6  OUT    UCSRB,R16
 7  LDI    R16,(1<<URSEL)|(3<<UCSZ0)
 8  OUT    UCSRC,R16
 9  LDI    R16,(1<<RXC)|(1<<MPCM)
10  OUT    UCSRA,R16
11  SEI
12  RET
```

Potongan program di atas merupakan subprogram yang digunakan untuk inisialisasi *port* serial, dengan mengisi register-register serial maka *port* D0 dan port D1 tidak lagi berfungsi sebagai *port input output*, *port* ini akan berfungsi sebagai kirim terima data serial.

Program pada baris ke-1 sampai dengan baris ke-4 merupakan pengisian register UBRR dengan pengisian nilai sesuai dengan variabel UBRRVAL. Sedangkan program pada baris 5 dan 6 digunakan untuk pengaturan register UCSRB, dimana bit RXEN bernilai 1 untuk membolehkan penerimaan data serial, bit TXEN bernilai 1 untuk membolehkan pengiriman data serial, dan bit RXCIE bernilai 1 untuk mengaktifkan interupsi penerimaan data serial.

Baris program 7 dan 8 merupakan pengisian register UCSR dengan bit URSEL bernilai 1 yaitu untuk mengakses register UBRR dengan register UCSRC, dan pada bit UCSZ bernilai 3 untuk memilih mode dan format data serial yaitu mode asinkronus dengan format data 1 bit start, 8 bit data, 1 bit stop, dan tidak ada paritas. Baris program 9 dan 10 yaitu pengisian register UCSRA, bit RXC bernilai 1 sebagai *flag* penerimaan 8 bit data selesai diterima, sedangkan bit MPCM diset 1 agar mengaktifkan komunikasi serial multiprosesor.

c. Mengirim data serial.

```
1  SerialTx:
2  SBIS    UCSRA,UDRE
3  RJMP    SerialTx
4  OUT     UDR,Tx
5  RET
```

Prosedur di atas merupakan prosedur untuk pengiriman data serial, prosedur ini sering digunakan untuk mengirimkan data serial. Program akan menunggu bit UDRE pada register UCSRA bernilai 1, jika bit ini bernilai 1 maka program pada baris 3 akan dilewatkan dan melanjukan ke program pada baris 4 untuk mengirimkan data serial.

d. Menerima data serial.

```
1  SerialRx:
2  SBIS    UCSRA,RXC
3  RJMP    SerialRx
4  IN      Rx,UDR
5  RET
```

Untuk penerimaan data serial prosedur di atas dapat digunakan jika menerima data serial. Program pada baris ke-2 merupakan program untuk menunggu selesainya penerimaan data serial, jika 8 bit data telah diterima, maka bit RXC akan di-*set* kemudian melewatkan satu baris program pada baris 3, kemudian membaca data serial yang dilakukan pada baris 4.

4.2.2 Program AFL Simulator

a. Membuka port serial.

```
1  Sub OpenSerialPort()
2  Dim LngNum As Integer
3  Dim PortNum As Integer
4  On Error GoTo ErrHandler
5  LngNum = Len(Me.ComboSett(0).Text)
6  LngNum = LngNum - 3
7  PortNum = Val(Right(Me.ComboSett(0).Text, LngNum))
8  With FrmAFL
9     With .MSComm1
10    .CommPort = PortNum
11    .Settings = Me.ComboSett(1).Text & "," & Me.ComboSett(4).Text & _
12    "," & Me.ComboSett(2).Text & "," & Me.ComboSett(3).Text
13    .PortOpen = True
14    End With
15 End With
16 Exit Sub
17 ErrHandler:
18 MsgBox Err.Description, vbInformation, "SERIAL PORT"
19 Err.Clear
20 End Sub
```

Untuk dapat menggunakan *port* serial komputer diperlukan prosedur untuk membuka *port* tersebut, pada program di atas merupakan prosedur untuk membuka *port* serial.

Program pada baris 2 dan 3 merupakan deklarasi variabel dengan tipe data *integer*, variabel ini digunakan untuk menentukan nomor *port* serial yang akan digunakan. Baris program 4 merupakan penanganan kesalahan yaitu ketika program membuka *port* serial terjadi kesalahan maka program akan menujuk ke subprogram ErrHandler. Baris program 5 samapai dengan 7 digunakan untuk mengambil nomor *port* pada objek ComboBox. Pada baris program 8 sampai 16 merupakan program untuk membuka *port* serial. Program pada baris ke-17 sampai dengan baris ke-20 berfungsi untuk penanganan kesalahan.

b. Menerima data melalui port serial.

```
1  Private Sub MSComm1_OnComm()
2          Select Case Me.MSComm1.CommEvent
3              Case comEvReceive
4                  If  Me.MSComm1.InBufferCount <> 0 Then
5                     DataSerial = Me.MSComm1.Input
6                  End If
7          …
8          End Select
   End Sub
```

Penerimaan data serial melalui *port* serial dilakukan dengan pembuatan prosedur seperti potongan program di atas, program ini akan dijalankan jika terjadi interupsi penerimaan data serial. Program basi ke-2 merupakan pemilihan *event* interupsi serial, sedangkan program baris ke-3 mememilih *event* penerimaan data serial. Pada baris ke-4 program memeriksa panjang *buffer* penerima data serial, jka *buffer* serial tidak sama dengan 0 maka program akan membaca data serial pada objek MsComm1 dan menampungnya di variabel DataSerial.

c. Mengirim data melalui port serial.

```
1  Private Sub TimerRequest_Timer()
2  Dim j As Integer
3  On Error Resume Next
4  If  j = 6 Then j = 0
5  Me.MSComm1.Output = Chr(Nstatus(j).id) & Chr(13)
   …
6  End Sub
```

Pengiriman data serial diperlukan prosedur pengiriman serial seperti pada prosedur program di atas, pada program baris ke-3 merupakan penangan kesalahan yaitu jika terjadi kesalahan pada baris program yang sedang dieksekusi maka program akan melewatkan baris program tersebut dan mengeksekusi baris program berikutnya. Pada baris program ke-4 merupakan penanganan variabel j, yaitu jika isi variabel j sama dengan 6 maka isi variabel j akan dinolkan kembali, variabel ini berfungsi untuk menunjukan alamat CCR Simulator. Baris program ke-5 merupakan program untuk mengirimkan data melalui *port* serial.

d. Membuka port TCP/IP.

```
 1  Sub ServerListening()
 2  With frmBrilliancy
 3     If .wnSck.State <> 0 Then
 4       MsgBox “LOCAL PORT ALREADY OPEN”
 5     Else
 6       .wnSck.LocalPort = Trim(Me.txtSocket(1).Text)
 7       .wnSck.Listen
 8       MsgBox "LOCAL PORT OPEN"
 9       Unload Me
10     End If
11  End With
12  End Sub
```

Komunikasi pada TCP/IP dilakukan oleh objek WinSock, prosedur untuk membuka *port* TCP/IP ditangani dengan prosedur di atas. Objek WinSock

diletakan pada form AFL sehingga pada prosedur ini dicantumkan frmAFL seperti pada baris program ke-2, pada baris ke-3 merupakan program untuk mengambil status *port* TCP/IP, jika status *port* dalam kedaan terbuka maka baris ke-4 akan dieksekusi yaitu dengan menampilkan sebuah pesan ke pengguna. Sebaliknya jika *port* dalam keadaan tertutup maka baris ke-6 akan dieksekusi yaitu dengan membuka nomor *port* sesuai dengan nilai objek EditText dan membukanya pada baris program ke-7. Pada baris ke-8, program menampilkan kotak dialog pesan bahwa *port* sudah terbuka.

e. Menerima data melalui port TCP/IP.

```
1  Private Sub wnSck_DataArrival(ByVal bytesTotal As Long)
2  Dim DataInstruktur As String
3  Me.wnSck.GetData DataInstruktur
4  Wait 0.1
7  End Sub
```

Prosedur di atas merupakan prosedur untuk mengangi penerimaan data melalui *port* TCP/IP yang telah terbuka. Pada baris ke-2 merupakan deklarasi variabel untuk menampung penerimaan data, sedangkan pada baris ke-3 yaitu memindahkan data yang berada pada objek WinSock ke dalam variabel DataInstruktur, kemudian program menunggu sampai penerimaan data selesai dengan memberikan waktu tunda 100 milisekon.

4.2.3 Program AFL Remote

a. Membuka port TCP/IP.

```
1  Int PORT=Integer.parseInt(strPort);
2  String ServerIP=getString(StrIP);
3  try{
4  InetAddress IP = InetAddress.getByName(ServerIP);
5  soket=new Socket(IP,PORT);
6  }catch(IOException e){
7  e.printStackTrace();
8  }
```

Komunikasi TCP/IP pada program aplikasi AFL Remote dilakukan oleh objek Socket, untuk menghubungkan tablet PC dengan personal komputer melalui *wireless* harus membuka nomor *port* dan TCP/IP yang sama dengan *server*. Pada potongan program di atas merupakan program untuk meminta koneksi ke *server*. Objek *Class* InetAdress merupakan *public class* untuk menampung nomor *port* dan TCP/IP tujuan. Pada baris program 1 sampai 2 merupakan

variabel yang menampung nomor *port* dan TCP/IP. Baris program ke-3 merupakan penanganan jika koneksi gagal. Baris program 4 yaitu mendeklarasikan variabel IP, variabel ini merupakan variabel *reference* dari *class* InetAddress yang diisi dengan variabel ServerIP. Pada baris ke-5, sebuah variabel *reference* yang diberinama soket merupakan variabel objek dari objek Socket, variabel ini kemudian diisi dengan nilai variabel IP dan variabel port agar terkoneksi. Baris program 6 dan 7 merupakan penanganan jika terjadi kesalahan koneksi atau nomor *port* dan nomor TCP/IP tujuan tidak ditemukan.

b. Mengirim data pada port TCP/IP.

```
1  class SendFault {
2  public void Transmit(String val){
3  try{
4  PrintWriter out=new PrintWriter(new BufferedWriter(new
5  OutputStreamWriter(soket.getOutputStream())),true);
6  out.println(val);
7  }catch(IOException e){
8  e.printStackTrace();}
9  }}}
```

Untuk mengirimkan data melalui objek Socket dibuatkan sebuah *class* baru dengan nama SendFault seperti pada tulisan program di atas, *class* ini sering dipanggil untuk mengirimkan data ke Socket. Pada baris 1 mendeklarasikan nama *class* dengan SendFault, pada baris 2 dibuatkan prosedur dalam class SendFault dengan nama Transmit. Baris program ke-3 merupakan penanganan kesalahan, pada baris program ke-4 merupakan *class* PrintWriter yang mereferensi pada variabel *out* untuk menampung data yang akan dikirimkan. Program mengirimkan data pada baris program ke-6. Untuk baris program ke-7 sampai dengan ke-8 merupakan baris penanganan kesalahan.

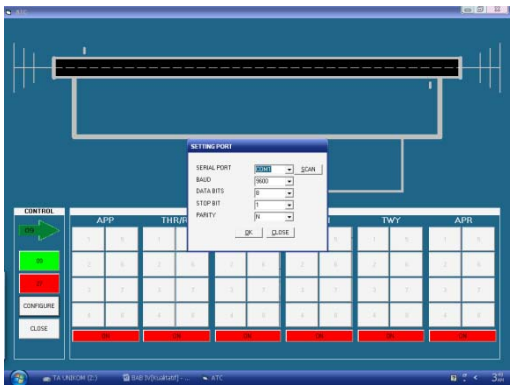
4.3 Pengujian Kerja Sistem Secara Keseluruhan

Untuk melakukan pengujian sistem yang telah dirancang yaitu dengan membuat program visual sederhana untuk *Air Traffic Controll* (ATC) dan perangkat PLC yang dibuat menggunakan mikrokontroler dengan *output* 8 bit data paralel, kedua perancangan perangkat ini tidak dibahas pada laporan Tugas Akhir. Pengujian sistem hanya menggunakan satu buah perangkat CCR Simulator yang digunakan untuk lampu AFL jenis *Approach*.

Media transmisi yang digunakan untuk komunikasi serial RS-485 antara personal komputer AFL Simulator dengan CCR Simulator menggunakan kabel UTP CAT5 dengan jarak 20 Meter. Sedangkan komunikasi antara PLC dengan CCR Simulator menggunakan kabel serabut, untuk terminal *input* ke-5 memiliki panjang kabel 100 Meter, sedangkan terminal *input* lainnya dengan panjang kabel 80 cm. Besar tegangan untuk simulasi yaitu 5V DC. Untuk IP address tablet PC dan personal komputer seperti pada pengujian perangkat keras yang telah dijelaskan sebelumnya. Berikut ini menjelaskan langkah-langkah konfigurasi pada masing-masing perangkat:

a. Konfigurasi pada program aplikasi ATC.

   Pada program aplikasi ATC yang perlu dikonfigurasi adalah nomor *port* serial yang akan digunakan. Pada Gambar 4.17 merupakan konfigurasi *port* serial.



Gambar 4.17. Konfigurasi Port Serial pada Program ATC

b. Konfigurasi pada program aplikasi AFL Simulator.

   Pada program aplikasi AFL Simulator memerlukan konfigurasi *port* serial dan nomor *port* TCP/IP yang akan digunakan.

   - Konfigurasi Port Serial

     Langkah-langkah untuk konfigurasi *port* serial adalah sebagai berikut:

     Pada menu Aksi pilih submenu Jalankan maka akan ditampilkan jendela AFL Simulator seperti terlihat pada Gambar 4.18.



Gambar 4.18. Tampilan Jendela AFL Simulator

Pada jendela AFL klik tombol Port Serial, maka jendela konfigurasi *port* serial akan ditampilkan seperti terlihat pada Gambar 4.19.



Gambar 4.19. Tampilan Konfigurasi Port Serial AFL Simulator

Dari jendela Port Serial, atur nomor *port*, *baud rate*, dan format data pada ComboBox seperti terlihat pada Gambar 4.20, untuk keperluan sistem maka format data *port* serial yang diperlukan adalah 1 bit start, 8 bit data, 1 bit stop, dan tidak ada paritas.

Untuk menentukan nomor *port* serial yang disediakan oleh personal komputer dapat dilakukan dengan menekan tombol *Scan*, maka pada ComboBox untuk nomor *port* hanya menampilkan nomor *port* yang disediakan seperti terlihat pada Gambar 4.20.



Gambar 4.20. Tampilan Pencarian Nomor Port pada AFL Simulator

Setelah konfigurasi *port* serial selesai klik tombol OK, maka pada jendela AFL Simulator akan mengubah status *port* serial dari *Close* menjadi *Open* seperti terlihat pada Gambar 4.21.



Gambar 4.21. Tampilan Status Port Serial

- Konfigurasi nomor port TCP/IP.

  Untuk komunikasi jaringan komputer pada AFL Simulator, diperlukan nomor *port* agar data yang diterima oleh *port* TCP/IP dapat berantarmuka dengan program aplikasi AFL Simulator. Adapun konfigurasi nomor *port* yaitu dengan cara mengklik tombol PORT TCP/IP, maka ditampilkan jendela TCP/IP seperti terlihat pada Gambar 4.22.



Gambar 4.22. Tampilan Konfigurasi Port TCP/IP

Pada EditText untuk IP secara otomatis program menampilkan TCP/IP lokal yang telah di *assign* sebelumnya, untuk nomor *port* dimasukkan pada objek EditText dengan *range* di atas 1023, hal ini untuk menghindari penggunaan nomor *port* yang termasuk dalam kategori *well-known port*. Adapun nomor *port* yang digunakan untuk pengujian yaitu *port* 5000, seperti terlihat pada Gambar 4.23.



Gambar 4.23. Tampilan Nomor Port TCP/IP yang Digunakan

Setelah menentukan nomor *port* yang digunakan kemudian klik tombol OK, jendeal AFL Simulator akan ditampilkan dan status *port* TCP/IP akan menampilkan pesan *Listen*, seperti terlihat pada Gambar 4.24.



Gambar 4.24. Tampilan Status Port TCP/IP

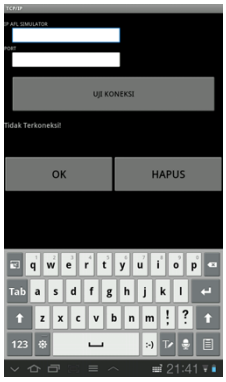
c. Konfigurasi pada program aplikasi AFL Remote.

Untuk mengendalikan secara *wireless* program aplikasi AFL Simulator dan CCR Simulator, maka pada AFL Remote diperlukan *assign* IP. Nomor IP tablet PC yang digunakan telah dijelaskan sebelumnya pada pengujian perangkat keras. Adapun langkah-langkah untuk assign IP remote dan nomor *port* yaitu dengan mengklik tombol menu, maka akan ditampilkan *context menu* sperti terlihat pada Gambar 4.25.



Gambar 4.25. Layout Utama AFL Remote

Pilih *context menu* Pengaturan, maka layout TCP/IP akan ditampilkan seperti terlihat pada Gambar 4.26.



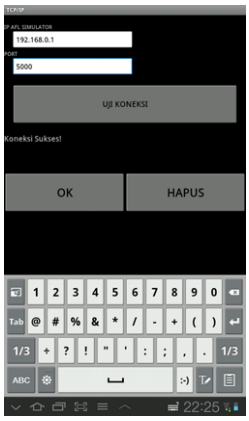
Gambar 4.26. Layout TCP/IP

Pada objek EditText IP AFL Simulator diisi dengan nomor IP yang sama dengan IP pada AFL Simulator, begitu juga dengan nomor *port* yang digunakan seperti terlihat pada Gambar 4.27.



Gambar 4.27. Assign IP address dan Nomor Port

Tekan Tombol Uji Koneksi, jika pengisian IP dan nomor *port* benar maka ditampilkan pesan “Koneksi Sukses” seperti terlihat pada Gambar 4.28.



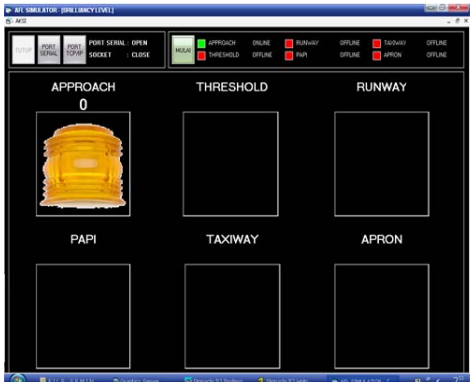
Gambar 4.28. Tampilan Pesan Status Koneksi

Setelah melakukan uji koneksi antara AFL Remote dengan AFL Simulator berhasil dilakukan, langkah selanjutnya adalah mengklik tombol OK, maka layout Utama akan ditampilkan kembali, dari layout utama tersebut klik menu kemudian pilih Jalankan maka akan ditampilkan layout Orange terlihat pada Gambar 4.29.



Gambar 4.29. Layout Orange

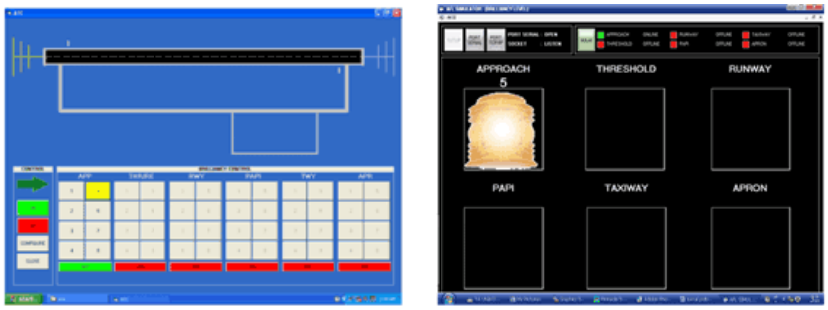
Setelah melakukan konfigurasi masing-masing perangkat, langkah selanjutnya adalah memulai mengoperasikan AFL Simulator dengan cara menekan tombol Mulai. Pada Gambar 4.30 merupakan AFL simulator menampilkan lampu jenis *Approach*.



Gambar 4.30. Tampilan Lampu Approach pada AFL Simulator

Enam jenis lampu yang disediakan oleh AFL Simulator hanya menampilkan satu jenis lampu yaitu lampu *Approach*, hal ini dikarenakan CCR Simulator yang terdeteksi oleh AFL Simulator hanya satu buah yaitu CCR Simulator untuk lampu *Apporach*.

Untuk Pengiriman data tingkat kecarahan lampu yaitu dengan mengklik tombol pada program aplikasi ATC, pada pengujian ini tombol yang dipilih adalah tombol 5. Pada Gambar 4.31 merupakan hasil pemilihan tombol 5 pada ATC dan lampu yang ditampilkan oleh AFL Simulator.



ATC AFL Simulator

Gambar 4.31. ATC Mengendalikan Lampu Approach pada Level 5

Data tingkat kecerahan lampu yang dikirimkan oleh ATC akan disimulasikan oleh AFL Simulator dengan tingkat kecerahan sama dengan data yang dikirimkan oleh ATC, hal ini mensimulasikan kondisi lampu AFL secara

visual dalam keadaan baik. Hasil pengukuran tegangan pada terminal *input* seperti terlihat pada Tabel 4.10.

Tabel 4.10. Data Hasil Pengukuran Tegangan pada Terminal Input CCR Simulator

No.	Terminal Input	Tegangan (V)
1	1	0 V
2	2	0 V
3	3	0 V
4	4	0 V
5	5	24 V
6	6	0 V
7	7	0 V
8	8	0 V

Dari Tabel 4.11 hanya terminal *input* ke-5 yang memiliki tegangan sebesar 24V, sedangkan tegangan pada terminal lainnya 0 V, hal ini mengindikasikan bahwa level kecerahan lampu yang dikendalikan oleh ATC yaitu pada level 5.

Sedangkan hasil pengukuran arus pada tiap terminal *output* CCR Simulator dilakukan dengan cara seperti terilihat pada Gambar 3.2.



Gambar 4.32. Pengukuran Arus pada Terminal Output CCR Simulator Lampu Approach

Data hasil pengukuran arus pada tiap terminal output dapat dilihat pada Tabel 4.11

Tabel 4.11. Data Hasil Pengukuran Arus Konstan pada Terminal Output CCR Simulator

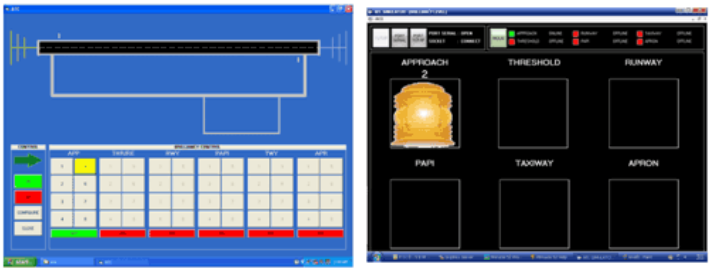
No.	Terminal Output	Arus (mA)
1	1	51.1
2	2	51.0
3	3	51.0
4	4	51.0
5	5	51.1
6	6	51.0
7	7	51.0
8	8	51.1

Berdasar data hasil pengukuran pada tabel 4.11 dengan mengabaikan selisih pengukuran sebesar 0.1 mA maka besar arus pada setiap terminal *output* sama besarnya yaitu 51mA, kondisi ini akan disimulasikan bahwa kondisi arus dalam keadaan konstan.

Untuk simulasi kerusakan yaitu dengan cara mengirimkan data simulasi kerusakan pada AFL Simulator yang dilakukan oleh AFL Remote. Simulasi kerusakan lampu AFL dikategorikan menjadi tiga jenis kerusakan yaitu *Low Current*, *Over Current*, dan *Open Circuit*. Pengujian mengenai simulasi kerusakan dijelaskan sebagai berikut:

a. Simulasi kerusakan jenis *Low Current*.

   Simulasi kerusakan lampu jenis *low current* yaitu CCR Simulator menghasilkan arus lebih rendah dari arus konstan pada nomor terminal sesuai dengan simulasi level kecerahan lampu yang ditampilkan oleh AFL Simulator. pada pengujian simulasi kerusakan *low current* yaitu dengan menekan tombol TIPE 2 pada AFL Remote, maka level kecerahan lampu akan seperti terlihat pada Gambar 4.33.



ATC AFL Simulator

Gambar 4.33.Simulasi Kerusakan Jenis Low Current

Terlihat pada Gambar 4.33 terjadi perubahan level kecerahan lampu menjadi level 2, sedangkan ATC mengatur level kecerahan lampu pada level 5. Untuk simulasi kerusakan secara teknis disimulasikan dengan adanya penurunan arus pada terminal *output* ke-2 CCR Simulator seperti terlihat pada Gambar 3.34..



Gambar 4.34. Pengukuran Arus dengan Kerusakan Jenis Low Current

Besar arus hasil pengukuran pada terminal *output* ke-2 yaitu sebesar 8.8 mA, arus ini lebih rendah dari arus konstan yaitu 51mA, maka kondisi ini sebagai simulasi kerusakan *low current*. Pada CCR yang sesungguhnya besar nilai peredaman arus dihitung dengan persentase, maka pada CCR Simulator jika peredaman arus dihitung dengan persentase dapat dilakukan dengan cara memberikan simbol matematis pada unsur-unsurnya yaitu sebagai berikut:

$Ic = \ Arus\ konstan\ (mA)$.

$It = Arus\ terukur(mA)$.

$Itr = Arus\ teredam\ (mA)$.

Dengan demikian dengan memasukan nilai hasil pengukuran terhadap unsur-unsurnya adalah sebagai berkut:

$Ic = 51mA$

$It = 8.8mA$

Sedangkan nilai Itr merupakan selisih nilai dari Ic dengan It yaitu sebagai berikut:

$Itr = Ic - Ii$

maka peredaman arus yaitu:

$Itr = 51mA - 8.8mA$

$= 42.2mA$

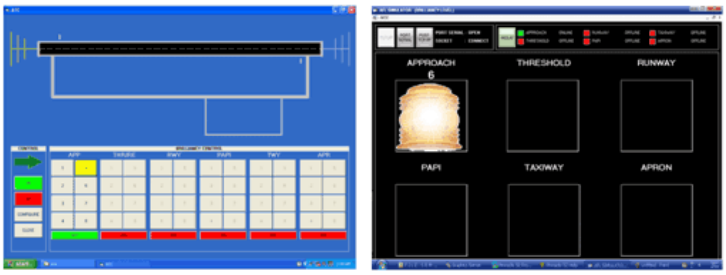
Dengan demikian arus teredam dalam persentase dapat dihitung dengan cara sebagai berikut:

$\%It = (42.2mA/51mA)x100 = 82.75\%$

Maka persentase peredaman arus sebesar 82.75%.

b. Simulasi kerusakan jenis *Over Current.*

Untuk simulasi kerusakan dengan kategori *over current* yaitu besar arus pada terminal output CCR Simulator lebih besar dari arus konstan pada normor terminal sesuai dengan simulasi level kecerahan lampu pada AFL Simulator. Pada pengujian ini simulasi kerusakan *over current* yaitu dengan mengklik tombol TIPE 6 pada AFL Remote, maka simulasi tingkat kecerahan lampu seperti terlihat pada Gambar 4.35.



ATC AFL Simulator

Gambar 4.35. Simulasi Kerusakan Jenis Over Current

Sedangkan secara teknis kerusakan ini disimulasikan pada CCR Simulator dengan mengukur besar arus terminal *output* ke-6 seperti terilihat pada Gambar 4.36.



Gambar 4.36. Pengukuran Arus dengan Kerusakan Jenis Over Current

Hasil pengukuran arus pada terminal *output* ke-6 yaitu 74 mA, arus ini lebih besar dari arus konstan. untuk perhitungan kenaikan arus dalam persentase dengan cara perhitungan yang sama dengan simulasi kerusakan jenis *low current*. Berikut adalah simbol matematis dari unsur-unsurnya:

$Ic = Arus\ konstan\ (mA)$

$It = Arus\ terukur\ (mA)$

$In = Kenaikan\ arus\ (mA)$

Untuk menghindari bilangan negatif maka perhitungan selisih arus dilakukan dengan cara sebagai berikut:

$$In = Ii - Ic$$
$$= 74mA - 51mA$$
$$= 23mA$$

Maka persentase kenaikan arus yaitu:

$$\%In = \frac{23mA}{51mA} \times 100 = 45\%$$

c. Simulasi kerusakan jenis *Open Circuit*.

Simulasi kerusakan dalam kategori *open circuit* yaitu terminal output CCR Simulator tidak mengeluarkan arus, sedangkan pada AFL Simulator akan ditampilkan tingkat kecerahan lampu level 0 seperti terlihat pada Gambar 4.37, simulasi kerusakan jenis ini dengan cara menekan tombol TIPE 9 pada AFL Remote.



ATC



AFL Simulator

Gambar 4.37. Simulasi Kerusakan Lampu Apporach Jenis Open Circuit

Hasil pengujian sistem di atas hanya mensimulasikan tiga jenis penekanan tombol pada AFL remote, oleh karena itu pada Tabel 4.12 menyajikan besaran arus CCR Simulator berdasar jenis simulasi kerusakan sesuai dengan penekanan tombol jenis kerusakan pada AFL Remote.

Tabel 4.12 Data Hasil Pengukuran Arus pada CCR Simulator Berdasar Penekan Tombol pada AFL Remote

Tombol Tipe	Besar Arus pada Terminal Keluaran (mA)								Simulasi Kerusakan
	1	2	3	4	5	6	7	8	
1	**8**	51.0	51.0	51.0	51.0	51.0	51.0	51.0	*Low Current*
2	51.1	**8.8**	51.0	51.0	51.0	51.0	51.0	51.1	
3	51.0	51.0	**15.8**	51.0	51.0	51.0	51.0	51.0	
4	51.0	51.0	51.0	**25.9**	51.0	51.0	51.0	51.0	
5	51.0	51.0	51.0	51.0	**63.6**	51.0	51.0	51.0	*Over Current*
6	51.0	51.0	51.0	51.0	51.0	**74**	51.0	51.0	
7	51.0	51.0	51.0	51.0	51.0	51.0	**107.5**	51.0	
8	51.0	51.0	51.0	51.0	51.0	51.0	51.0	**130.5**	
9	**0.0**								*Open Circuit*