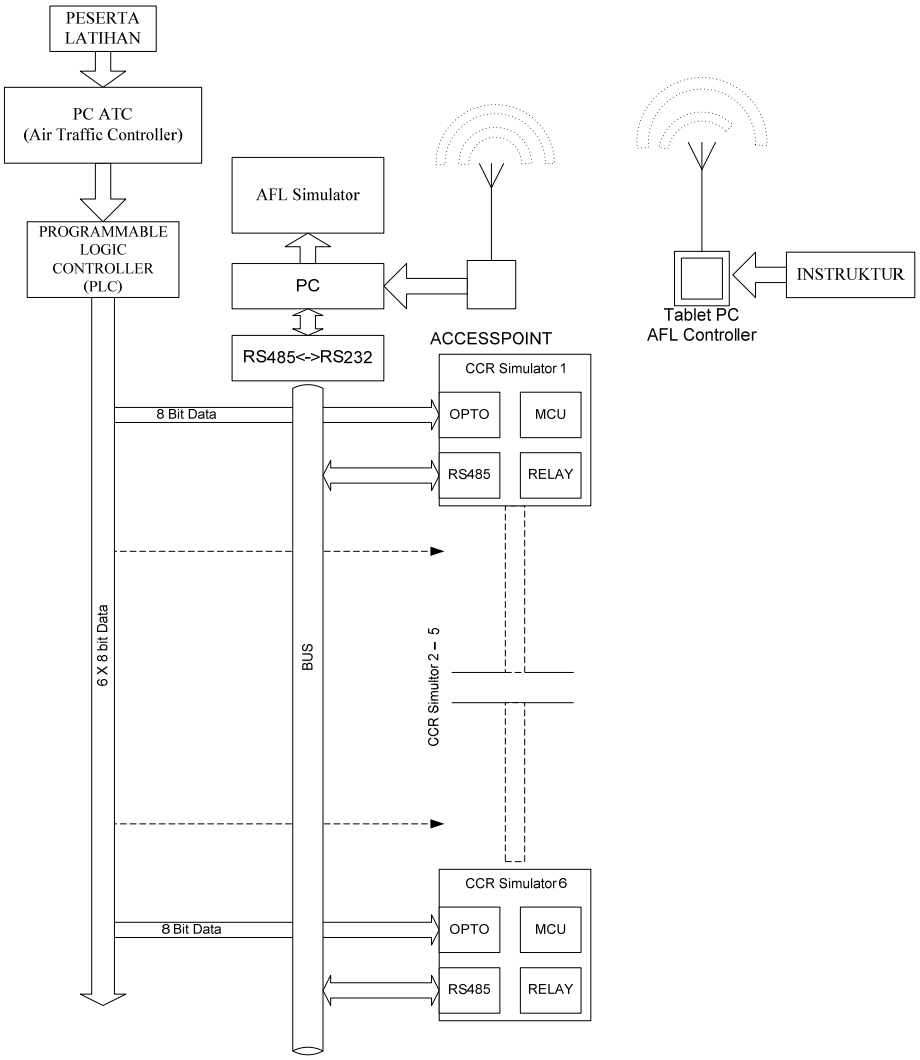
BAB III
PERANCANGAN

3.1 Perancangan Sistem

Setelah membahas mengenai teori-teori penunjang pada Bab 2 sebagai pendukung dalam pembuatan sistem, maka pada bab ini penulis akan membahas mengenai perancangan sistem yang akan dibuat. Secara garis besar pada bab ini berisi penjelasan perancangan blok diagram sistem, perangkat CCR Simulator, program aplikasi AFL Simulator, dan program aplikasi AFL Remote.

3.1.1 Blok Diagram Sistem Secara Keseluruhan



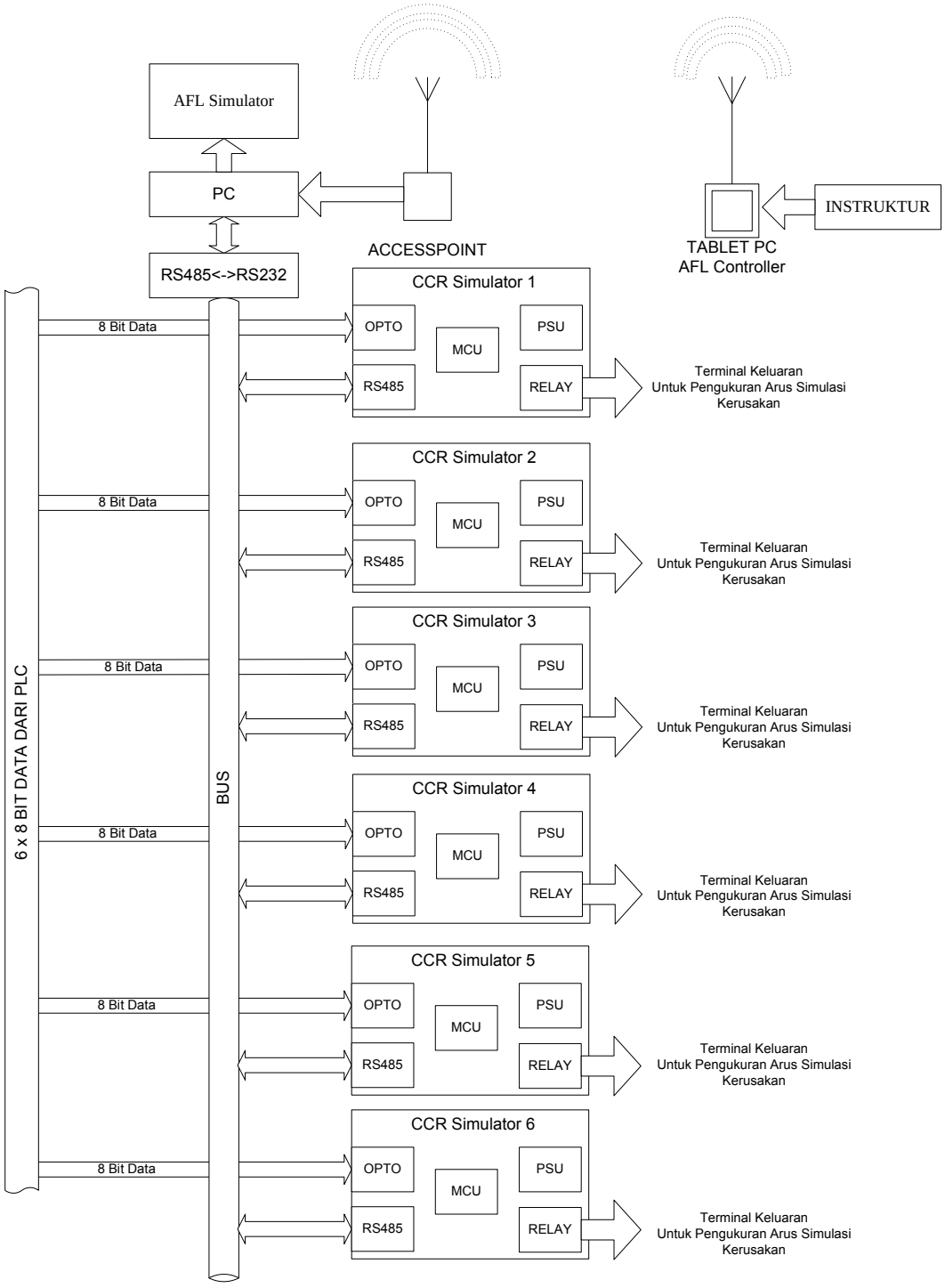
Gambar 3.1. Blok Diagram Sistem Secara Keseluruhan

Seperti terlihat pada Gambar 3.1 yaitu menggambarkan diagram blok sistem secara keseluruhan, adapun urutan proses kerja sistem yaitu sebagai berikut:

a. Peserta latihan operator AFL secara manual memasukan data tingkat kecerahan lampu melalui personal komputer ATC (*Air Traffic Controller*) yaitu dengan cara mengklik tombol navigasi tingkat kecerahan lampu.
b. Data tingkat kecerahan lampu tersebut akan dikeluarkan oleh personal komputer ATC melalui *port* serial dan dikirimkan ke PLC.
c. Data serial yang diterima oleh PLC akan dibaca dan dikeluarkan kembali dalam bentuk data paralel 8 bit kemudian dikirimkan ke salah satu CCR Simulator yang ditunjuk.
d. Jika AFL Simulator meminta status *input* CCR Simulator tertentu, maka CCR Simulator yang ditunjuk akan membaca data status *input* dan hasilnya akan dikirimkan ke AFL Simulator.
e. AFL Simulator akan mensimulasikan tingkat kecerahan lampu AFL sesuai dengan penerimaan data yang dikirimkan oleh CCR Simulator.
f. Jika instruktur memberikan nilai kesalahan terhadap AFL Simulator maka AFL Simulator akan menampilkan tanda adanya kesalahan pada CCR Simulator dan melanjutkan pengiriman data kesalahan tersebut ke CCR Simulator.

3.1.2 Blok Diagram Sistem yang Dirancang

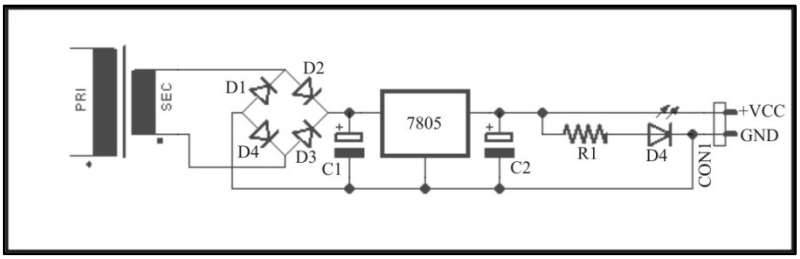
Komponen-komponen yang terdapat pada blok diagram sistem secara keseluran tidak seluruh blok komponen tersebut dirancang pada Tugas Akhir ini. Blok diagram sistem yang tidak dijelaskan dalam perancangan yaitu blok ATC dan blok PLC, kedua blok ini akan dibuatkan program sederhana yang nantinya akan digunakan sebagai pelengkap untuk menguji sistem yang telah dibuat. Pada Gambar 3.2 merupakan blok diagram sistem yang akan dirancang.



Gambar 3.2. Blok Diagram Sistem yang Dirancang

3.2 Perancangan Sumber Catu Daya

Catu daya merupakan perangkat keras yang berfungsi untuk menyediakan sumber tegangan yang diperlukan untuk men-*supply* perangkat keras agar perangkat tersebut dapat bekerja. Seperti terlihat pada Gambar 3.3 merupakan rangkaian catu daya yang akan dirancang, sumber tegangan yang akan dihasil oleh oleh rangkaian catu daya yaitu 5V DC.



Gambar 3.3. Rangkaian Catu Daya

Penggunaan transformator catu daya berfungsi untuk memperoleh tegangan AC kecil, tegangan tersebut merupakan tegangan induksi yang dihasilkan oleh kumparan sekunder, karena transformator yang digunakan adalah transformator catu daya maka keluaran tegangan sekunder lebih kecil dibanding dengan sumber tegangan primer.

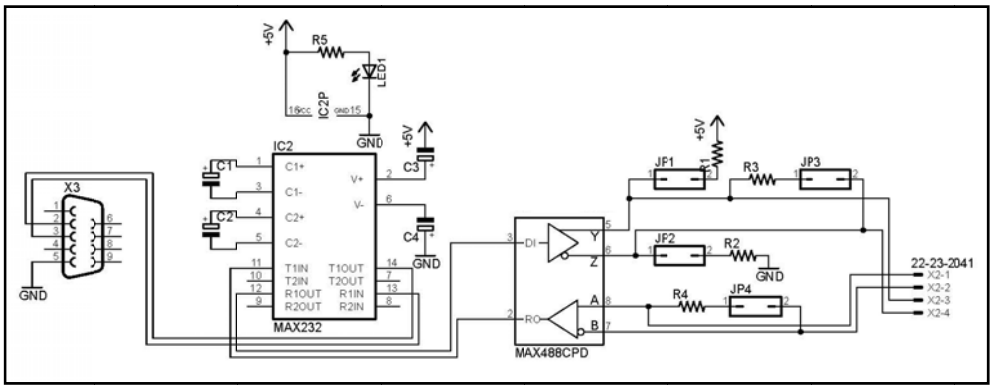
Tegangan yang dikeluarkan oleh kumparan sekunder masih dalam bentuk tegangan AC, sehingga diperlukan dioda penyearah untuk memperoleh tegangan DC. Penyusunan empat buah dioda yang disusun dengan metode *bridge* bertujuan untuk memperoleh tegangan DC gelombang penuh (*full-wave*).

Tegangan yang keluar pada dioda *bridge* tidak langsung menghasilkan tegangan DC rata, oleh karena itu diperlukan tapis kapasitor. Untuk memperoleh tegangan DC 5V stabil maka digunakan IC regulator 7805. Tegangan yang masuk pada IC 7805 harus lebih besar dari tegangan keluaran, IC 7805 merupakan salah satu IC regulator tegangan positif.

3.3 Perancangan Konverter RS-232 – RS-485

Komunikasi antara CCR Simulator dengan personal komputer akan digunakan komunikasi serial RS-485. Pada port serial personal komputer hanya

menyediakan komunikasi serial RS-232 sehingga diperlukan perangkat untuk mengonversi serial RS-232 menjadi RS-485. Pada Gambar 3.4 merupakan rangkaian konverter RS-232 menjadi RS-485 atau sebaliknya.



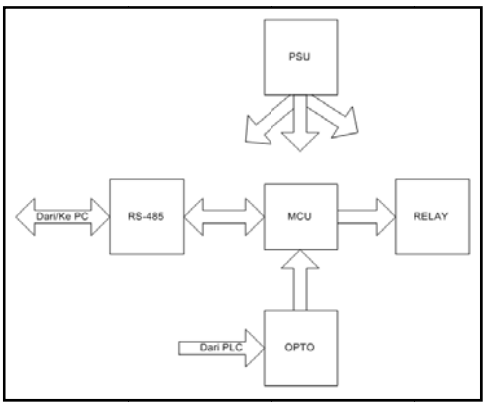
Gambar 3.4. Rangkaian Konverter RS-232 – RS-485

3.4 Perancangan CCR Simulator

Perancangan perangkatat CCR Simulator meliputi blok diagram sistem CCR Simulator, perancangan perangakat keras, dan perangkat lunak.

3.4.1 Blok Diagram CCR Simulator

Perangkat keras CCR Simulator berfungsi sebagai simulasi kerusakan yang mungkin terjadi pada lampu AFL, perangkat keras ini digunakan untuk melatih peserta calon operator AFL secara teknis dalam mengenal jenis kerusakan seperti *over current*, *low current*, dan *open circuit*. Perangkat keras CCR Simulator memiliki blok diagram seperti pada Gambar 3.5.



Gambar 3.5. Blok Diagram CCR Simulator

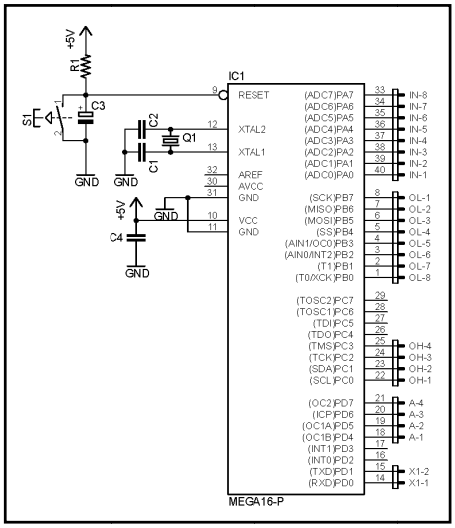
Fungsi dari masing-masing blok CCR Simulator pada Gambar 3.5 adalah sebagai berikut:

- Blok *Power Supply Unit* (PSU) atau dikenal dengan catu daya digunakan sebagai penyedia sumber tegangan DC untuk men-*supply* seluruh blok CCR Simulator.
- Blok RS-485 digunakan untuk mengonversi data serial standar RS-232 level TTL menjadi data serial standar RS-485 atau sebaliknya.
- Blok *Microcontroller Unit* (MCU) berfungsi memproses data yang diterima secara serial maupun data hasil pembacaan *port input*.
- Blok Relay berfungsi sebagai sakelar elektromagnetik yang di-*drive* oleh *port output* untuk menentukan jenis kerusakan dengan memilih nilai resistor.
- Blok *Optical Photo* (OPTO) berfungsi sebagai logika digital yang akan dibaca oleh *port input* mikrokontroler.

3.4.2 Perancangan Perangkat Keras CCR Simulator

Pada subbab ini menjelaskan perancangan rangkaian perangkat keras untuk CCR Simulator yang terdiri dari rangkaian mikrokontroler ATmega16 dan rangkaian lainnya yang berada pada blok CCR Simulator.

3.4.2.1 Rangkaian Mikrokontroler ATmega16



Gambar 3.6. Rangkaian ATmega16

Mikrokontroler yang akan digunakan yaitu ATmega16, pada Gambar 3.6 merupakan rangkaian mikrokontroler ATmega16 yang akan dirancang. Pada Tabel 3.1 merupakan penjelasan pin ATmega16 yang digunakan.

Tabel 3.1. Pin ATmega16 yang Digunakan

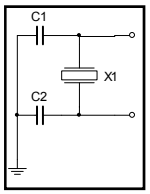
No	Pin	Nama	Fungsi	Keterangan
1	40	PA0	*Input*	*Input* 8 bit data paralel dari optocoupler.
2	39	PA1	*Input*	
3	38	PA2	*Input*	
4	37	PA3	*Input*	
5	36	PA4	*Input*	
6	35	PA5	*Input*	
7	34	PA6	*Input*	
8	33	PA7	*Input*	
9	1	PB0	*Output*	*Ouput* 12 bit data paralel untuk men-*drive* relay.
10	2	PB1	*Output*	
11	3	PB2	*Output*	
12	4	PB3	*Output*	
13	5	PB4	*Output*	
14	6	PB5	*Output*	
15	7	PB6	*Output*	
16	8	PB7	*Output*	
17	22	PC0	*Output*	
18	23	PC1	*Output*	
19	24	PC2	*Output*	
20	25	PC3	*Output*	
21	14	PD0	RXD	Menerima data serial
22	15	PD1	TXD	Mengirim data serial
24	18	PD4	*Input*	Identitas CCR simulator
25	19	PD5	*Input*	
26	20	PD6	*Input*	
27	21	PD7	*Input*	
28	9	RST	*Input*	*Reset*
29	13	XTAL1	*Input*	*Input clock*
30	12	XTAL2	*Output*	*Output clokc*
31	10	VCC	VCC	Tegangan VCC
32	11	GND	*Ground*	Tegangan *ground*

Seperti telihat pada rangkaian Gambar 3.6 terdapat dua rangkaian yang ada pada mikrokontroler ATmega16, kedua rangkaian tersebut dapat dijelaskan sebagai berikut:

1. Rangkaian osilator.

   Mikrokontroler ATmega16 dalam mengeksekusi instruksi memerlukan sumber detak (*clock*), pada Gambar 3.7 sumber *clock* yang digunakan berasal

dari kristal dengan frekuensi 11059200 Hz dan dua buah kapasitor sebagai resonator yang memiliki kapasitansi 22pF (lembar data). Adapun rangkaian osilator seperti terlihat pada Gambar 3.7 merupakan rangkaian osilator.

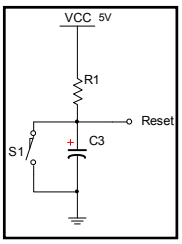


Gambar 3.7. Rangkaian Osilator

2. Rangkaian reset.

   Mikrokontroler yang beroperasi cukup lama akan mempengaruhi proses kerjanya, seperti terjadinya ketidaksesuaian antara proses kerja mikrokontroler dengan tujuan pembuatan program, sehingga sewaktu-waktu harus me-*reset* proses kerja mikrokontroler agar *Program Counter* (PC) kembali ke alamat awal program.

   Mikrokontroler akan melakukan *reset* jika pada pin 9 (*reset*) diberikan logika *low*. Adapun rangkaian *reset* mikrokontroler seperti yang terlihat pada Gambar 3.8.



Gambar 3.8. Rangkaian Reset.

Pada saat reset aktif besar arus yaitu 0.5 mA, maka nilai R1 ditentukan dengan persamaan sebagai berikut:

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

Sedangkan nilai C1 ditentukan dengan persamaan sebagai berikut:

$$C = \frac{t(s)}{R(Ohm)}(F)$$

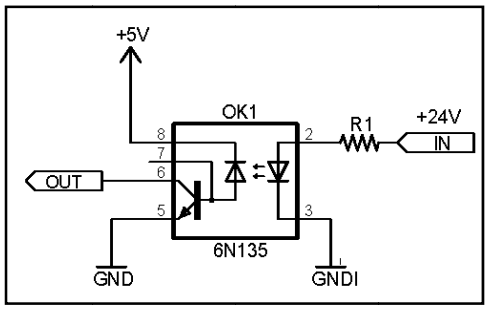
3.4.2.2 Rangkaian Optocoupler 6N135

Optocoupler yang digunakan dalam perancangan sistem yaitu 6N135, yang memiliki tegangan *forward* minimum 1.35V dan arus *forward* maksimum sebesar 25mA. Pada Tabel 3.2 merupakan pin *Optocoupler* 6N135 yang digunakan.

Tabel 3.2. Pin Optocoupler yang Digunakan

No Pin	Fungsi
2	Tegangan *input* LED
3	Tegangan *Ground* LED
5	Tegangan *Ground*
6	Tegangan *input photo transistor*
8	Tegangan positif

Maka rangkaian *Optocoupler* yang akan dirancang seperti terlihat pada Gambar 3.9 merupakan rangkaian *optocoupler*.



Gambar 3.9. Rangkaian Optocoupler 6N135

Untuk menentukan besar arus yang diperlukan oleh dioda LED dapat dicari dengan persamaan sebagai berikut:

$$IF = \frac{IC_{sat}}{CTR}$$

Di mana

IF : Arus forward yang diperlukan oleh LED.

CTR : Current Transfer Ratio dalam persen (Lihat datasheet).

ICsat : Arus saturasi yang diingkan.

Jika nilai IF telah diketahui maka besar resistansi dapat ditentukan dengan persamaan:

$$R_1 = \frac{V_s - V_{Led}}{IF}$$

Di mana

R_1 : Nilai resistansi.

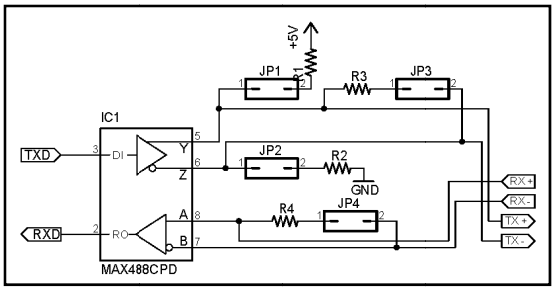
V_s : Tegangan sumber.

V_{Led}: Tegangan LED.

IF : Arus Forward.

3.4.2.3 Rangkaian Serial RS-485

Seperti telah dijelaskan pada subbab sebelumnya yaitu komunikasi antara CCR Simulator dengan personal komputer menggunakan komunikasi serial RS-485 maka pada rangkaian mikrokontroler ATmega16 juga harus memiliki rangakaian serial RS-485 hal ini dikarenakan komunikasi serial ATmega16 serial asinkronus level TTL. Pada Gambar 3.10 merupakan rangkaian RS-485 pada CCR Simulator.

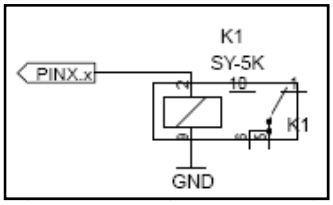


Gambar 3.10. Rangkaian Serial RS-485

3.4.2.4 Rangkaian Relay SY-5-K

Pada perancangan ini relay difungsikan untuk memilih besar resistansi yang dapat mempengaruhi besar arus sehingga dapat digunakan sebagai simulasi jenis kerusakan pada lampu AFL.

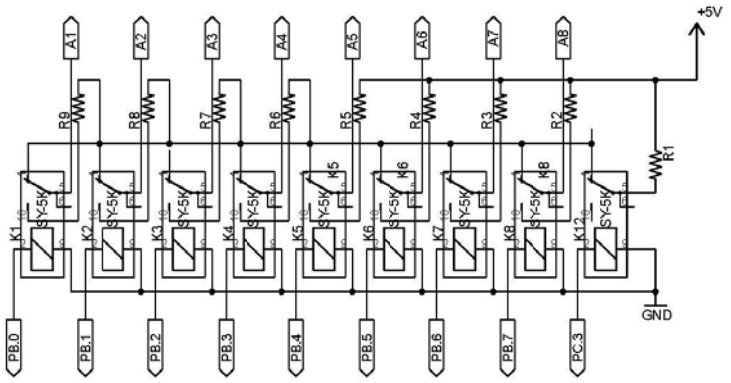
Salah satu pin kumparan pada relay dihubungkan dengan sebuah pin mikrokontroler ATmega16 dan pin lainnya dihubungkan ke *ground*, dengan metode pengkawatan pin relay tersebut maka relay akan aktif ketika pin mikrokontroler yang terhubung dengan relay memiliki tegangan *high* (1). Pada Gambar 3.11 merupakan hubungan setiap relay dengan pin mikrokontroler.



Gambar 3.11. Rangkaian Relay

3.4.2.5 Rangkaian Pembentuk Kerusakan

Rangkaian simulasi kerusakan menggunakan resistor tetap yang terhubung dengan tegangan 5 V DC, masing-masing resistor dihubungkan dengan relay yang telah dirancang pada perancangan rangakaian relay. Pengkawatan resistor pembentuk simulasi kerusakan dengan relay seperti terliihat pada Gambar 3.12.



Gambar 3.12. Rangkaian Simulasi Kerusakan

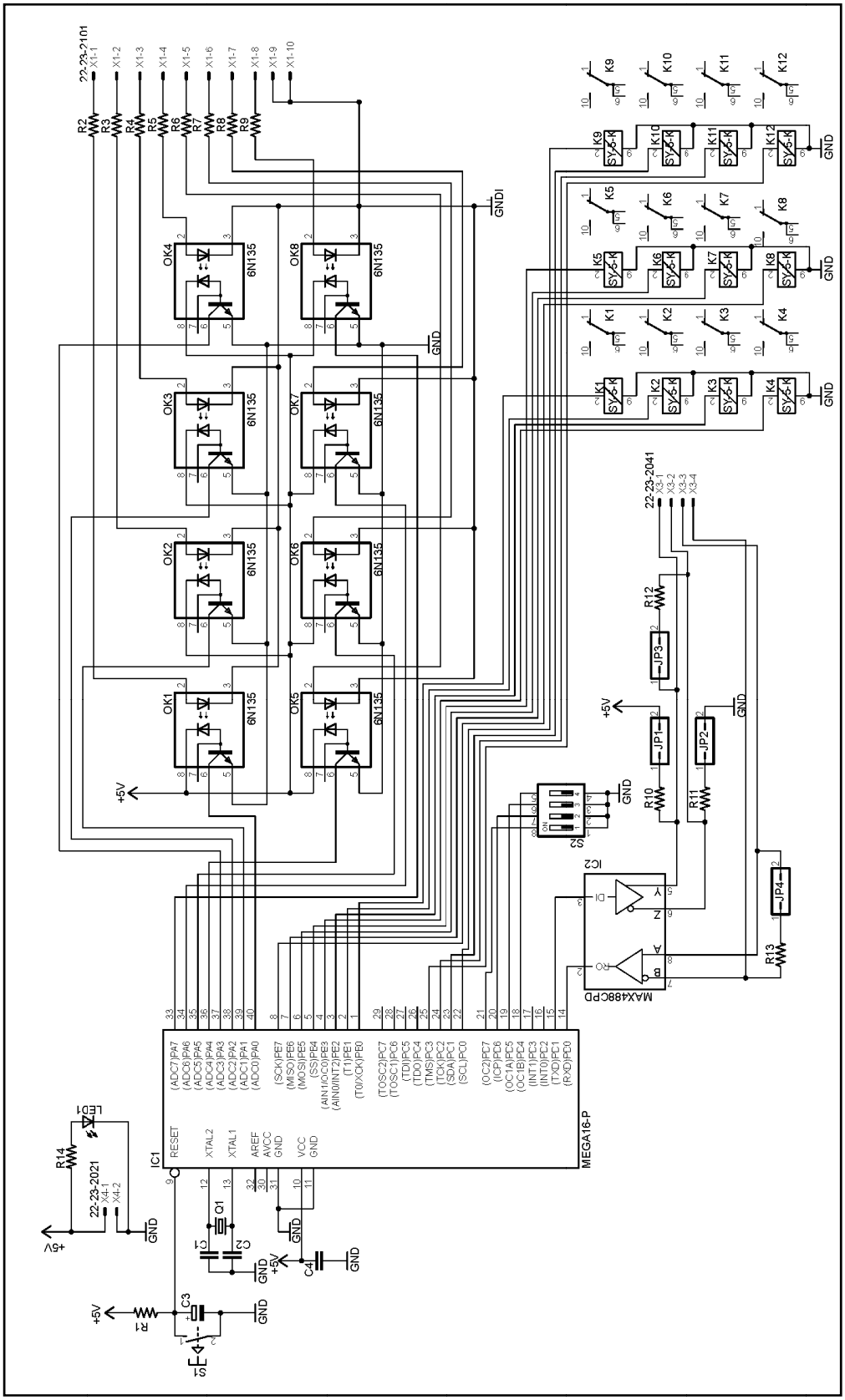
Besar nilai resistansi untuk masing-masing resistor ditentukan dengan persamaan:

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

Resistor R1 berfungsi untuk membatasi arus masuk pada saat pengukuran simulasi arus konstan, sedangkan resistor lainnya berfungsi untuk mensimulasikan kerusakan.

3.4.2.6 Rangkain CCR Simulator Secara Keseluruhan

Setelah merancang tiap-tiap bagian rangkaian yang terdapat pada perangkat CCR Simulator, maka jika rangkaian tersebut digabungkan akan terlihat seperti pada Gambar 3.13.



Gambar 3.13. Rangkaian CCR Simulator Secara Keseluruhan

3.4.3 Perancangan Perangkat Lunak CCR Simulator

Perancangan perangkat lunak ini membahas format perintah yang dikenal oleh CCR Simulator, pengaturan register *port input/output*, pengaturan register yang berhubungan dengan komunikasi serial, dan diagram alur program.

3.4.3.1 Perancangan Format Perintah

Pengiriman dan penerimaan data serial CCR Simulator satu dengan lainnya melalui sebuah bus yang sama, sehingga diperlukan identitas unik dari masing-masing CCR Simulator. Jumlah maksimum yang disediakan oleh CCR Simulator yaitu 2^4 identitas atau sama dengan 16 buah CCR Simulator.

CCR Simulator (*slave*) akan memberikan respon kepada master jika *master* mengirim identitas yang sesuai dengan identitasnya, dan mengabaikan permintaan master jika identitas yang diterima tidak sesuai. CCR Simulator yang ditunjuk oleh master akan melanjutkan komunikasi berikutnya yaitu dengan membaca perintah yang dikirimkan oleh *master*.

Adapun format perintah yang akan dirancang yaitu dengan format <ID><Perintah><Data><CR>, format perintah dari masing-masing tag ini memiliki arti sebagai berikut:

a. <ID> : merupakan nomor identitas CCR Simulator.
b. <Perintah> : merupakan data perintah berupa data karakter O dan karakter F.
c. <Data> : jika perintah yang digunakan adalah karakter O maka tag ini dibiarkan kosong, jika perintah berupa F maka tag ini diisi dengan data yang akan dikeluarkan oleh mikrokontroler dengan jumlah data 2 Byte dalam bentuk bilangan desimal.
d. <CR> : merupakan batas akhir dari pengiriman data.

Penjelasan mengenai penggunaan format perintah yang dikirimkan oleh *master* dan respon *slave* adalah sebagai berikut:

1. Master : <ID><CR>
   Slave : <ID><”:”>
2. Master : <ID><”O”><CR>
   Slave : <ID><”O”><Data>
3. Master : <ID><”F”><Data><CR>
   Slave : *High Byte* dikeluarkan pada PORT C
   Low Byte dikeluarkan pada PORT B

Panjang ID yang dikirimkan oleh *master* untuk CCR Simulator memiliki panjang data 1 Byte. Sedangkan panjang data yang dikirimkan oleh *master* yaitu 2 Byte, di mana 1 Byte tinggi akan dikeluarkan oleh CCR Simulator pada *port* C, sedangkan 1 Byte rendah akan dikeluarkan pada *port* B.

Panjang ID hasil dari respon CCR Simulator memiliki panjang data 2 Byte, dalam bentuk bilangan heksadesimal yang direpresentasikan dalam kode ASCII, begitu juga dengan panjang data yang dikirimkan oleh CCR Simulator memiliki ukuran data yang sama dengan ukuran ID yaitu 2 Byte.

3.4.3.2 Perancangan Register Port Input/Output

Perancangan ini menjelaskan mengenai konfigurasi register DDR untuk *port* yang difungsikan sebagai *input* dan *output*.

- *Port input*

  Sesuai dengan perancangan perangkat keras CCR Simulator, *port* yang berfungsi sebagai *input* yaitu *port* A yang digunakan untuk membaca *input* dari Optocoupler, dan *port* D4 sampai dengan port D7 digunakan untuk membaca identitas CCR Simulator dengan cara membaca *dip switch* 4 bit.
  Untuk konfigurasi port A sebagai *input* maka isi register DDRA bernilai 0, dan memiliki kondisi awal port A yaitu berlogika *high*, untuk mendapatkan kondisi awal bernilai high maka konfigurasi register DDRA yaitu sebagai berikut:

```
LDI   r16,0x00
OUT   DDRA,r16
LDI   r16,0xff
OUT   PORTA,r16
```

Sedangkan untuk *port* D konfigurasi registernya adalah sebagai berikut:

```
LDI    r16,0x00
OUT    DDRD,r16
LDI    r16,0xf0
OUT    PORTD,r16
```

- *Port Output*

  Port output pada CCR simulator yaitu terletak pada *port* B dan *port* C. pada *port* C hanya 4 bit rendah yang difungsikan sebagai *output* yaitu *port* C0 sampai dengan *port* C3.

  P*ort* B dan port C digunakan untuk men-*drive* relay, di mana kondisi relay aktif *high*, maka kondisi port B dan port C harus memiliki kondisi awal *low* , dengan demikian konfigurasi register DDRB adalah sebagai berikut:

```
LDI    r16,0xff
OUT    DDRB,r16
LDI    r16,0x00
OUT    PORTB,r16
```

  Sedangkan untuk *port* C yaitu sebagai berikut:

```
LDI    r16,0x0f
OUT    DDRB,r16
LDI    r16,0x00
OUT    PORTB,r16
```

3.4.3.3 Perancangan Register Komunikasi Serial

Pada *port* D selain dikonfigurasi sebagai *input*, *port* D juga digunakan sebagai jalur komunikasi serial yang terletak pada *port* D0 dan *port* D1. Pada *port* D0 digunakan sebagai jalur penerimaan data serial, sedangkan *port* D1 digunakan sebagai jalur pengiriman data serial.

Agar komunikasi serial pada *port* D0 dan *port* D1 dapat bekerja, maka diperlukan konfigurasi register-register yang berhubungan dengan komunikasi serial. Adapun perancangan kecepatan modulasi dan frame data komunikasi serial yaitu sebagai berikut:

- *Baud rate* 9600 bps.
- Frame data yaitu 1 bit start, 8 bit data, 1 bit stop dan tidak ada paritas.
- Dapat melakukan mengirim dan menerima data serial.

Untuk memenuhi komunikasi serial ini maka isi register serial dilakukan dengan langkah-langkah sebagai berikut:

- Mengisi nilai register UBRR (USART Baud Rate Register) dengan persamaan:

  $$UBRR = \frac{f_{osc}}{16 \times baud\ rate} - 1$$

  Berdasar perancangan maka nilai unsur-unsurnya adalah sebagai berikut:

  Fosc=11059200

  Baud rate=9600

  Maka nilai UBRR adalah:

  $$UBRR = \left(\frac{11059200}{16 \times 9600}\right) - 1 = 71_{10} = 47_{16}$$

- Men-*set* bit pada register UCSRB (USART *Control Status Register* B) agar dapat mengirim dan menerima data serial. Adapun bit pada register UCSRB yang di-*set* yaitu sebagai berikut:

  TXEN=1

  RXEN=1

- Men-*set* bit pada register UCSRC (USART *Control Status Register* C) yaitu untuk mengatur ukuran karakter. Adapun register yang harus di-*set* yaitu sebagai berikut:

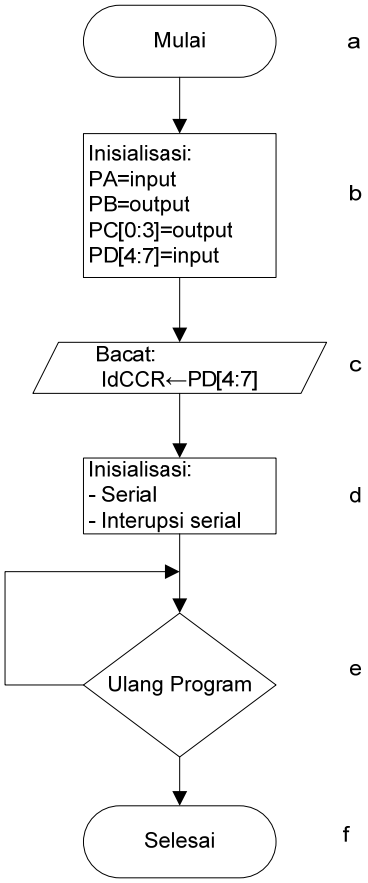
  UCSZ2:UCSZ1:UCSZ0: =3

  UPM=0

  URSEL=1

3.4.3.4 Diagram Alur CCR Simulator

Diagram alur dari program utama CCR Simulator seperti terlihat pada Gambar 3.14.



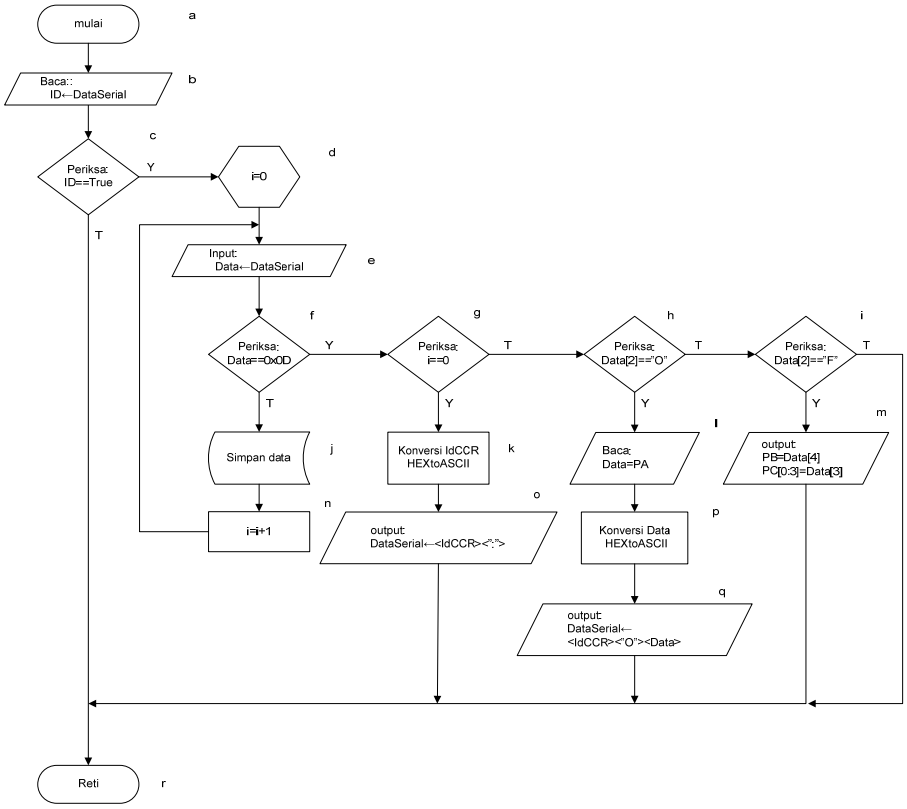
Gambar 3.14. Diagram Alur Program Utama CCR Simulator

Berikut adalah penjelasan mengenai alur program utama CCR Simulator.

a. Awal program dijalankan.
b. Program melakukan proses inisialisasi yang meliputi definisi variabel, pemetaan memori, konfigurasi *port input/output*.
c. Program membaca alamat sebagai identitas dirinya (*address based*) pada *port* D yang telah dikonfigurasi pada proses inisialisasi, dengan demikian jika terjadi perubahan alamat maka CCR Simulator harus di-*reset*.
d. Program malakukan proses inisialisasi komunikasi serial yaitu dengan mengonfigurasi register-register yang berhubungan dengan komunikasi serial, pada langkah ini juga interupsi serial diaktifkan.

e. Program melakuakan perulangan secara terus menerus di alamat program dengan lokasi yang sama.
f. Akhir program.

Program akan menajalankan diagram alur pada Gambar 3.15 ketika terjadi interupasi serial dan kembali lagi ke diagram alur program utama ketika proses interupsi selesai dijalankan.



Gambar 3.15. Diagram Alur Program Interupsi Serial

Diagram Alur program interupsi serial dijelaskan sebagai berikut:

a. Awal program.
b. Program membaca identitas yang dikirimkan oleh master.
c. Jika identitas sama dengan identitas yang dimiliki oleh CCR Simulator maka program akan melanjutkan ke langkah d, jika identitas tidak sama maka program melanjutkan ke langkah r.
d. Program mengisi nilai awal i dengan nol untuk menghitung panjang data yang diterima, kemudian dilanjutkan ke langkah e.

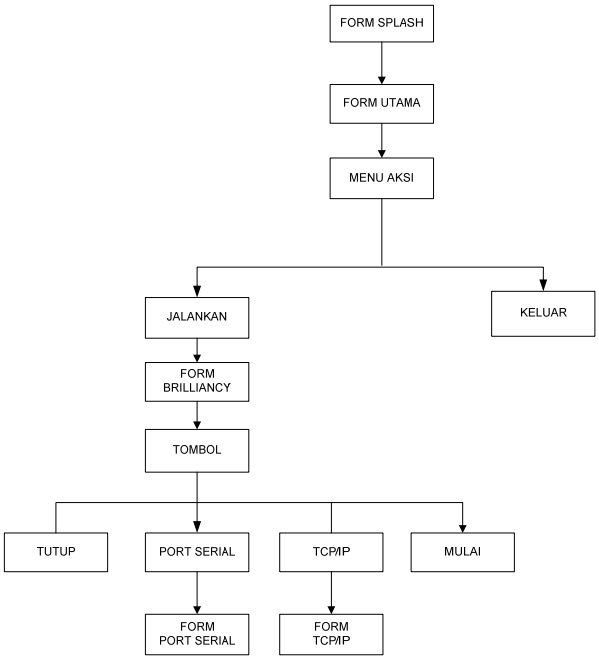
e. Program menunggu penerimaan data yang dikirimkan oleh master kemudian memeriksa data tersebut pada langkah f.
f. Program memeriksa data serial yang diterima, jika data tersebut adalah data ASCII 13 maka program melanjutkan ke langkah g, jika data tersebut bukan data ASCII 13 maka program melanjutkan ke langkah j.
g. Program memeriksa panjang data serial yang diterima, jika panjang data sama dengan nol maka program melanjutkan ke langkah k, jika tidak sama dengan nol maka program melanjutkan ke langkah h.
h. Program memeriksa perintah pada nomor data urutan ke dua, jika perintah sama dengan “O” maka program akan melanjutkan ke langkah l, jika tidak maka program akan memeriksa perintah pada langkah i.
i. Program memeriksa perintah pada nomor data urutan ke dua, jika perintah sama dengan “F” maka program akan melanjutkan ke langkah m, jika tidak maka program melanjutkan ke langkah r.
j. Program menyimpan penerimaan data serial dan melanjutkan ke langkah n.
k. Program melakukan konversi identitas dirinya dari bilangan heksa menjadi kode ASCII, dan melanjutkan ke langkah o.
l. Program membaca status *input port* A.
m. Program mengeluarkan data serial yang diterima ke port B untuk low byte dan port C untuk high byte, dan melanjutkan ke langkah r.
n. Program menaikkan penghitung penerimaan panjang data dengan satu, kemudian program kembali ke langkah e.
o. Program mengirimkan identitas yang sudah dikonversi dalam kode ASCII diikuti dengan karakter colon “:”, kemudian program melanjutkan ke langkah r.
p. Program mengonversi hasil pembacaan input padat port A menjadi kode ASCII dan melanjutkan ke langkah q.
q. Program mengirimkan status port A melalui komunikasi serial dan melanjutkan ke langkah r.
r. Program mengaktifkan kembali interupsi serial dan kembali ke program utama.

3.5 Perancangan Program Aplikasi AFL Simulator

Simulasi tingkat kecerahan lampu AFL akan ditampilkan pada program aplikasi AFL Simulator, maka pada subbab ini menjelaskan mengenai perancangan dalam membuat program aplikasi AFL Simulator. Adapun pembahasan perancangan meliputi struktur navigasi form aplikasi, tampilan masing-masing form, dan diagram alur program.

3.5.1 Struktur Navigasi AFL Simulator

Perancangan program aplikasi AFL Simulator akan dirancang dengan menggunakan bahasa pemrograman Visual Basic 6.0, adapun struktur navigasi pada program aplikasi AFL Simulator seperti terlihat pada Gambar 3.16.



Gambar 3.16. Struktur Navigasi AFL Simulator

Adapun penjelasan dari fungsi masing-masing struktur navigasi form AFL Simulator adalah sebagai berikut:

- Form Splash, merupakan form yang hanya ditampilkan pada saat program aplikasi AFL Simulator pertama kali dijalankan.

- Form Utama, merupakan form induk dari seluruh form pada program aplikasi AFL Simulator. Form Utama hanya memiliki sebuah menu yaitu menu Aksi.
- Menu Aksi memiliki dua buah submenu yaitu menu Jalankan dan menu Keluar.
- Submenu Jalankan digunakan untuk menampilkan form Brilliancy.
- Submenu Keluar digunakan untuk menutup program aplikasi AFL Simulator.
- Form Brilliancy merupakan form yang digunakan untuk menampilkan simulasi tingkat kecerahan lampu AFL, pada form ini terdapat objek kontrol tombol untuk menampilkan form Serial, form TCP, menutup form Brilliancy, dan memulai proses simulasi lampu AFL.
- Form Serial berfungsi untuk melakukan konfigurasi port serial yang akan digunakan.
- Form TCP/IP berfungsi untuk asign TCP/IP yang akan digunakan.

3.5.2 Perancangan Tampilan Form AFL Simulator

Setelah perancangan struktur navigasi AFL Simulator, langkah selanjutnya yaitu merancang tampilan dari masing-masing form untuk interaksi antara program aplikasi dengan pengguna.

1. Tampilan form Splash.

   Form Splash hanya ditampilkan sekali yaitu ketika program aplikasi AFL Simulator pertama kali dijalankan, perancangan tampilan form Splash seperti yang terlihat pada Gambar 3.17.



Gambar 3.17. Perancangan Tampilan Form Splash

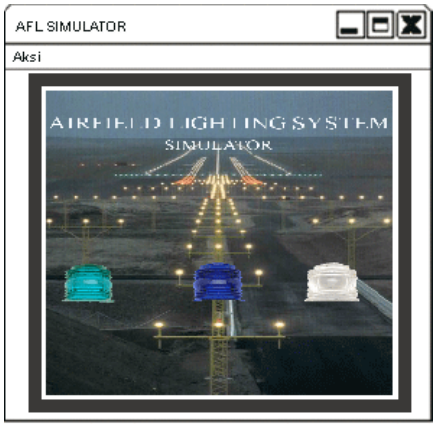
Form Splash akan disembunyikan atau ditutup ketika pengguna menekan sebarang tombol pada *keyboard*, objek gambar dan/atau ketika *interval* waktu pada objek timer telah terpenuhi. Adapun objek yang terdapat pada form Splash diperlihatkan pada Tabel 3.3.

Tabel 3.3. Objek dan Perubahan Properti pada Form Splash

Objek	Properti	Perubahan
Image	Name	ImgFlash
	Picture	ATC.gif
	Stratch	True
Label	Name	Label1
	Index	0
	Caption	SELAMAT DATANG
Label	Name	Label1
	Index	1
	Caption	DI AFL SIMULATOR
Timer	Name	Timer1
	Interval	3000

2. Tampilan form Utama.

   Perancangan form Utama seperti yang terlihat pada Gambar 3.18, form ini akan ditampilkan ketika form Splash ditutup.



Gambar 3.18. Perancangan Tampilan Form Utama

Pada form Utama terdapat satu buah menu yaitu menu Aksi, beberapa objek yang terdapat pada form Utama dan perubahan properti seperti pada Tabel 3.4.

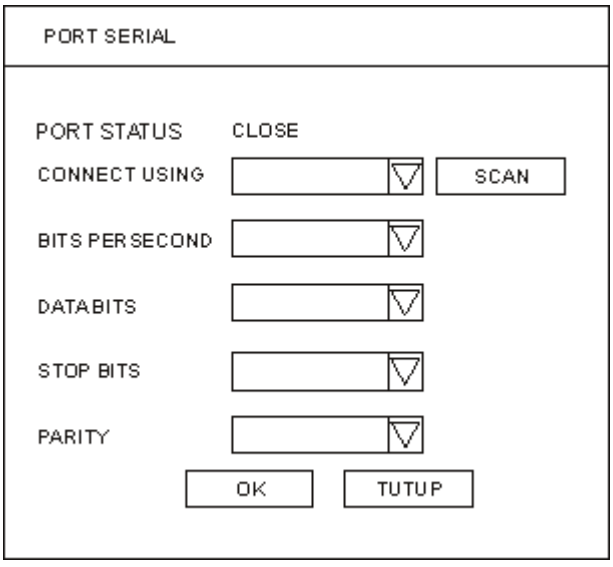
Tabel 3.4. Objek dan Perubahan Properti pada Form Utama

Objek	Properti	Perubahan
Picture	Name	Picture1
	Picture	BgMain.gif
	Stratch	True
Menu	Name	MnAksi
	SubMenu	Jalankan
		Keluar

Menu Aksi pada form Utama memiliki dua submenu yaitu submenu Jalankan dan submenu Keluar. Submenu Jalankan akan menampilkan form Brilliancy, sedangkan submenu Keluar berfungsi untuk keluar dari program aplikasi AFL Simulator.

3. Tampilan form Serial.

   Perancangan form Serial seperti terlihat pada Gambar 3.19 akan ditampilkan ketika objek tombol "PORT SERIAL" pada form Brilliancy terpilih oleh pengguna, form ini digunakan untuk konfigurasi *port* serial.



Gambar 3.19. Perancangan Tampilan Form Serial

Pada Tabel 3.5 merupakan perubahan properti objek dari standar Visual Basic yang terdapat pada form Serial.

Tabel 3.5. Objek dan Perubahan Properti pada Form Serial

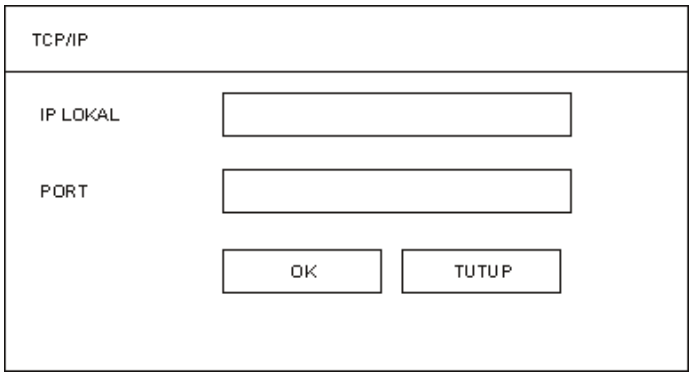
Objek	Properti	Perubahan
Label	Index	0
	Name	Label1
	Caption	PORT STATUS
	Index	1
	Name	Label1
	Caption	CLOSE
	Index	2
	Name	Label1
	Caption	CONNECT USING
	Index	3
	Name	Label1
	Caption	BITS PERSECOND
	Index	4
	Name	Label1
	Caption	DATA BITS
	Index	5
	Name	Label1
	Caption	STOP BITS
	Index	6
	Name	Label1
	Caption	PARITY
ComboBox	Index	0
	Name	ComboSett
	Text	-
	Index	1
	Name	ComboSett
	Text	-
	Index	2
	Name	ComboSett
	Text	-
	Index	3
	Name	ComboSett
	Text	-
	Index	4
	Name	ComboSett
	Text	-
Command	Index	0
	Name	Command1
	Caption	&SCAN
	Index	1
	Name	Command1
	Caption	&OK
	Index	2
	Name	Command1
	Caption	&CLOSE

Konfigurasi format data yang diperlukan oleh *port* serial terdapat pada objek ComboBox(0) sampai dengan ComboBox(4), sedangkan objek Label(0) sampai dengan objek Label(6) berfungsi sebagai keterangan dari konfiguasi *port* serial.

Objek Command(0) berfungsi sebagai pencarian nomor serial *port* secara otomatis yang disediakan oleh personal komputer, sedangkan objek Command(1) berfungsi sebagai persetujuan konfigurasi *port* serial yang disesuaikan dengan pemilihan pada objek ComboBox, penekanan terhadap command(1) akan menghubungkan dengan objek MsComm pada form Brilliancy, jika konfigurasi didukung oleh objek MsComm maka form komunikasi serial akan disembunyikan secara otomatis, sebaliknya jika tidak didukung akan ditampilkan pesan kesalahan. Pada objek Command(2) berfungsi untuk keluar dari form komunikasi dan menutup *port* serial.

4. Tampilan form TCP/IP.

   Program aplikasi AFL Simultor juga mendukung komunikasi jaringan komputer yang didukung dengan protokol TCP/IP. Untuk asign nomor port TCP ditampilkan pada form TCP/IP. Adapun perancangan form TCP/IP seperti terlihat pada Gambar 3.20.



Gambar 3.20. Perancangan Tampilan Form TCP/IP

Pada Tabel 3.6 merupakan objek perubahan properti yang terdapat pada form TCP/IP.

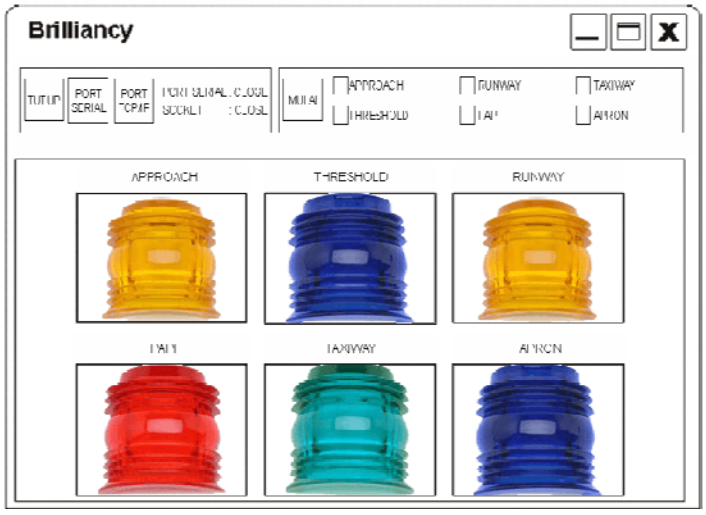
Tabel 3.6. Obejk dan Perubahan Properti pada Form TCP/IP

Objek	Properti	Perubahan
Label	Index	0
	Name	Label1
	Caption	IP LOKAL
	Index	1
	Name	Label1
	Caption	PORT
TextBox	Index	0
	Name	TextBox1
	Text	-
	Index	1
	Name	TextBox1
	Text	-
Command	Index	0
	Name	Command1
	Caption	&OK
	Index	1
	Name	Command1
	Caption	&TUTUP

Objek Label(0) dan Label(1) merupakan keterangan dari masing-masing objek TextBox, di mana TextBox(0) akan menampilkan IP address lokal pada saat form komunikasi TCP/IP ditampilkan, sedangkan objek TextBox(1) merupakan nomor *port* akan digunakan.

5. Tampilan form Brilliancy.

   Form Brilliancy merupakan form yang mensimulasikan tingkat kecarahan lampu AFL, form ini merupakan inti dari aplikasi AFL Simulator. Penerimaan data *input/output* baik dari *port* serial maupun *port* TCP dilakukan pada form Brilliancy. Perancangan tampilan form Brilliancy seperti terlihat pada Gambar 3.21.



Gambar 3.21. Perancangan Tampilan Form Brilliancy

Pada form Brilliancy menyediakan enam jenis lampu AFL yaitu lampu *Approach*, *Threshold*, *Runway*, PAPI, *Taxiway*, dan *Apron*. Objek gambar lampu akan menampilkan simulasi tingkat kecarahan lampu AFL sesuai dengan data yang diterima dari *port* serial maupun *port* TCP. Kondisi *port* serial jika dalam keadaan terbuka label keterangan port serial akan menampilkan teks “OPEN”, begitu juga dengan port TCP jika dalam keadaan terbuka akan menampilkan label keterangan “LISTEN” dan jika terjadi koneksi dengan *host* lainnya label keterangan akan menampilkan teks “CONNECT”. Pada Tabel 3.7 merupakan objek dan perubahan properti dari masing-masing objek yang terdapat pada form Brilliancy.

Tabel 3.7. Objek dan Perubahan Properti pada Form Brilliancy

Objek	Properti	Perubahan
Label	Index	0
	Name	lblSett
	Caption	PORT SERIAL:
	Index	1
	Name	lblSett
	Caption	SOCKET:
	Index	0
	Name	lblStatusCCR
	Caption	APPROACh
	Index	1
	Name	lblStatus CCR
	Caption	THRESHOLD

Tabel 3.8. Lanjutan Objek dan Perubahan Properti pada Form Brilliancy

Objek	Properti	Perubahan
Label	Index	2
	Name	lblStatusCCR
	Caption	RUNWAY
	Index	3
	Name	LblStatusCCR
	Caption	PAPI
	Index	4
	Name	lblStatusCCR
	Caption	TAXIWAY
	Index	5
	Name	lblStausCCR
	Caption	APRON
	Index	-
	Name	lblApprocah
	Caption	APPROACH
	Index	-
	Name	lblThreshold
	Caption	THRESHOLD
	Index	-
	Name	lblRunway
	Caption	RUNWAY
	Index	-
	Name	lblPapi
	Caption	PAPI
	Index	-
	Name	lblTaxiway
	Caption	TAXIWAY
	Index	-
	Name	lblApron
	Caption	APRON
Shape	Index	0
	Name	shpStsCCR
	FillColor	&H00FFFFFF&
	Index	1
	Name	shpStsCCR
	FillColor	&H00FFFFFF&
	Index	2
	Name	shpStsCCR
	FillColor	&H00FFFFFF&
	Index	3
	Name	shpStsCCR
	FillColor	&H00FFFFFF&
	Index	4

Tabel 3.9. Lanjutan Objek dan Perubahan Properti pada Form Brilliancy

Objek	Properti	Perubahan
Shape	Name	shpStsCCR
	FillColor	&H00FFFFFF&
	index	5
	Name	shpStsCCR
	FillColor	&H00FFFFFF&
Command	Index	0
	Name	cmdSett
	Caption	TUTUP
	Index	1
	Name	cmdSett
	Caption	PORT SERIAL
	Index	2
	Name	cmdSett
	Caption	PORT TCP/IP
	Index	3
	Name	cmdSett
	Interval	MULAI
Timer	Enable	False
	Name	TimerRequest
	Interval	100
	Enable	False
	Name	TmrTimeOut
	Interval	1500
	Enable	False
	Name	TimerRequestCCR
	Interval	200
	Enable	False
	Name	TimerFlipRed
	Interval	1000
	Enable	False
	Name	TimerFlipGreen
	I nterval	1000
Image	Index	0
	Name	ImgLight
	Stretch	True
	Picture	Yellow.gif
	Index	1
	Name	ImgLight
	Stretch	True
	Picture	Blue.gif
	Index	2
	Name	ImgLight
	Stretch	True

Tabel 3.10. Lanjutan Objek dan Perubahan Properti pada Form Brilliancy

Objek	Properti	Perubahan
Image	Picture	Yellow.gif
	Index	3
	Name	ImgLight
	Stretch	True
	Picuture	Red.gif
	Index	4
	Name	ImgLight
	Stretch	True
	Picture	Green.gif
	Index	5
	Name	ImgLight
	Stretch	True
	Picture	Blue.gif
Winshock1	Name	wnSck
MsComm1	Name	MsComm1

Objek Label(0) berfungsi untuk menampilkan pesan terhadap status koneksitas *port* serial, sedangkan objek Label(1) berfungsi untuk menampilkan pesan status *port* TCP/IP, yaitu *port* dalam keadaan terbuka, tertutup atau dalam keadaan terkoneksi dengan komputer lain.

Fungsi dari objek Label(3) sampai dengan objek Label(7) yaitu untuk menampilkan teks yang menginformasikan hasil dari dari status CCR Simulator, jika CCR Simulator dalam keadaan aktif maka objek Label lblStatusCCR pada index ke-n akan menampilkan teks *Online* sedangkan CCR yang tidak aktif maka objek Label lblStatusCCR akan menampilkan pesan *Offline*.

Untuk mengindikasikan CCR yang sedang *Online* maupun *Offline* digunakan enam buah objek Shape dengan nama shpStsCCR, objek ini juga berfungsi sebagai indikator terjadinya kesalahan terhadap CCR. Objek akan dikendalikan oleh objek Timer agar menampilkan warna yang bergantian sesuai dengan *interval* waktu yang telah diprogram.

Enam buah objek Image berindeks akan menampilkan simulasi lampu AFL sesuai dengan pembacaan data dari mikrokontroler, gambar simulasi lampu

AFL akan ditampilkan jika CCR Simulator dalam keadaan aktif dan disembunyikan jika CCR Simulator dalam keadaan mati atau *Offline*.

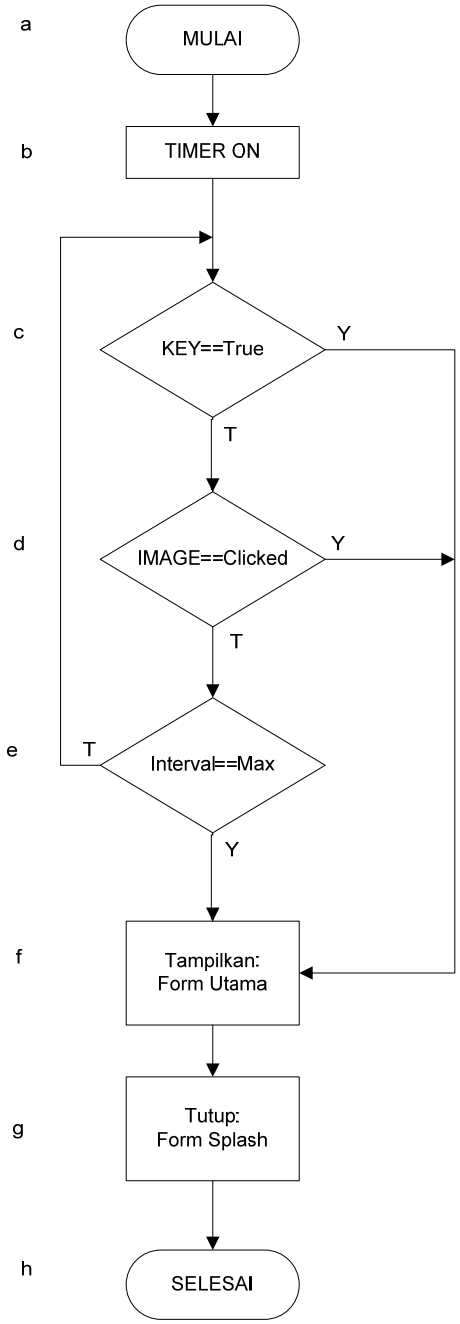
Objek Timer dengan nama TimerRequest pada form Brilliancy memiliki fungsi untuk meminta identitas CCR yang dimulai dari 0 sampai dengan 5, objek TmrTimeOut akan menghitung sampai *interval* waktu yang telah ditentukan, jika CCR Simulator yang diminta tidak memberikan respon maka CCR Simulator dengan alamat yang ditunjuk dianggap tidak aktif. Sedangkan dua buah objek Timer (TimerFlipRed dan TimerFlipGreen) berfungsi untuk menggerakan objek Shape(0) sampai dengan Shape(5) jika diterima data kesalahan oleh AFL Simulator.

Sebuah objek MsComm berfungsi untuk mengirim dan menerima data ke/dari CCR Simulator, data yang diterima oleh objek MsComm akan ditampilkan oleh objek Image. Sedangkan objek Winsock berfungsi untuk mengonfigurasi nomor *port* TCP/IP yang akan digunakan untuk terhubung dengan jaringan.

3.5.3 Diagram Alur Program AFL Simulator

Subbab ini menjelaskan diagram alur program yang terdapat pada program aplikasi AFL Simulator. Diagram alur tersebut meliputi diagram alur form Splash, form Utama, form Serial, form TCP, dan form Brilliancy

1. Diagram alur program pada form Splash.



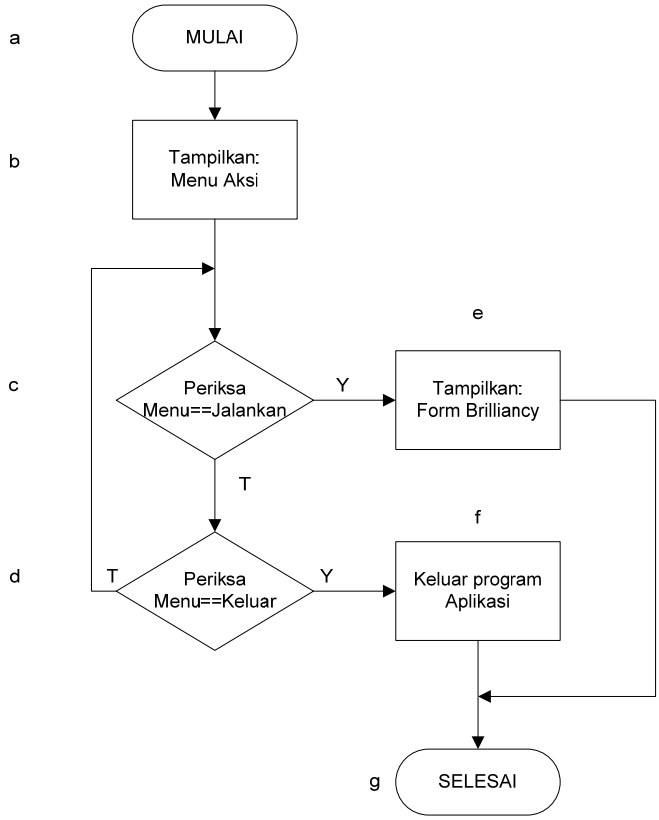
Gambar 3.22. Diagram Alur Form Splash

Pada Gambar 3.22 merupakan diagram alur program yang terdapat pada form Splash, alur program ini dapat dijelaskan sebagai berikut:

a. Awal program dijalankan.
b. Timer diaktifkan, timer ini berfungsi untuk waktu tunda form Splash ditampilkan.

c. Program menunggu penekanan sebarang tombol keyboard, jika terjadi penekanan maka program akan melanjutkan ke langkah f.
d. Program juga menunggu pengguna mengklik objek Image pada form Splash, jika objek tersebut diklik maka program akan melanjutkan ke langkah f.
e. Jika timer telah mencapai nilai interval maksimum maka program akan melanjutkan ke langkah f.
f. Program menampilkan form utama.
g. Program menutup form dirinya sendiri.
h. Akhir program.

2. Diagram alur program pada form Utama.

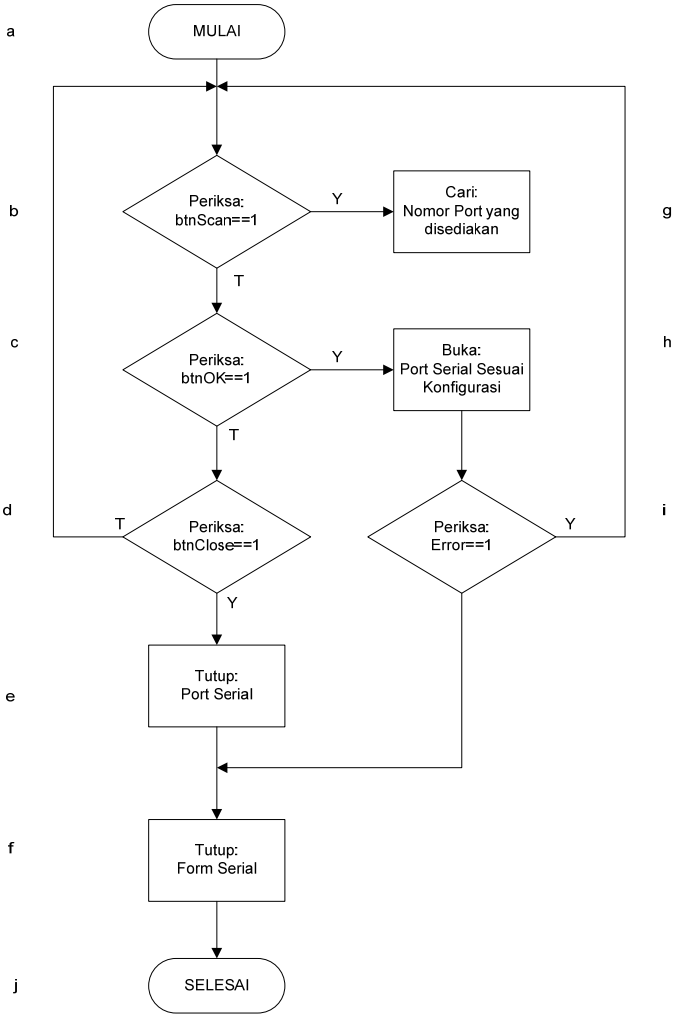


Gambar 3.23. Diagram Alur Form Utama

Form Utama merupakan form yang menyediakan menu-menu untuk menjalankan form Brilliancy dan untuk keluar dari program aplikasi. Berikut adalah penjelasan alur program form Utama.

a. Awal program form utama.
b. Program menyediakan menu Aksi
c. Program menunggu pemilihan salah satu menu menu yang dipilih oleh pengguna. Jika pengguna memilih menu Jalankan maka Program akan melanjutkan kelangkah e, jika tidak maka program akan memeriksa menu pada langkah d.
d. Progam mejalankan langkah f jika menu Keluar terpilih oleh pengguna
e. Program menampilkan form Brilliancy.
f. Program menutup aplikasi.
g. Akhir program.

3. Diagram alur program pada form Serial.

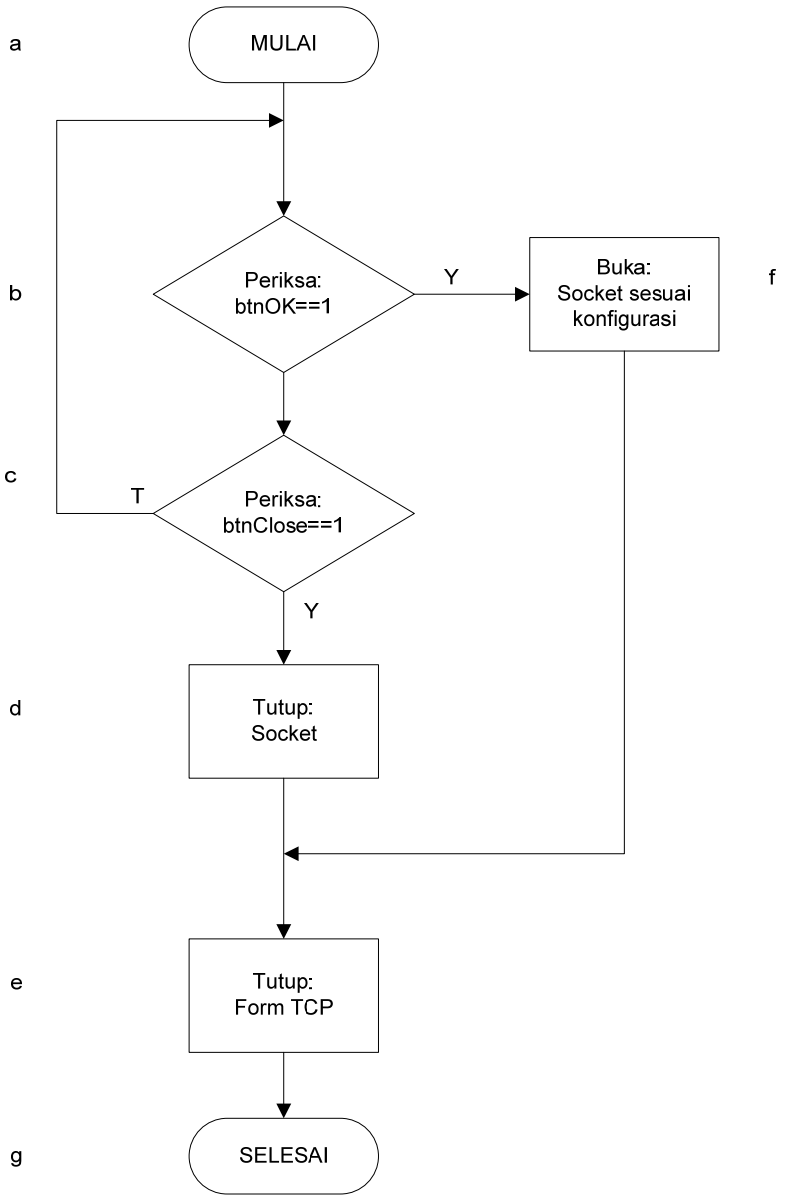


Gambar 3.24. Diagram Alur Form Serial

Pada form Serial terdapat tiga buah objek kontrol CommandButton, masing-masing objek CommandButton menunggu pengguna melakukan pengklikan pada salah satu objek tersebut. Berikut penjelasan mengenai alur program yang terdapat pada form Serial.

a. Awal program dijalankan.
b. Program menjalankan langkah g jika pengguna mengklik tombol Scan.
c. Program Menjalankan langkah h jika pengguna mengklik tombol OK.
d. Program menjalankan Langkah e jika pengguna mengklik tombol Close.
e. Program menutup *port* serial
f. Program menutup form dirinya sendiri dan melanjutkan ke langkah j.
g. Program menjalankan pencarian nomor *port* yang disediakan oleh personal komputer.
h. Program membuka port serial.
i. Jika *port* serial yang dibuka terjadi *error* maka program kembali ke langkah b, jika *port* serial sukses maka program melanjutkan ke langkah f.
j. Akhir program.

4. Diagram alur program pada form TCP/IP



Gambar 3.25. Diagram Alur Form TCP/IP

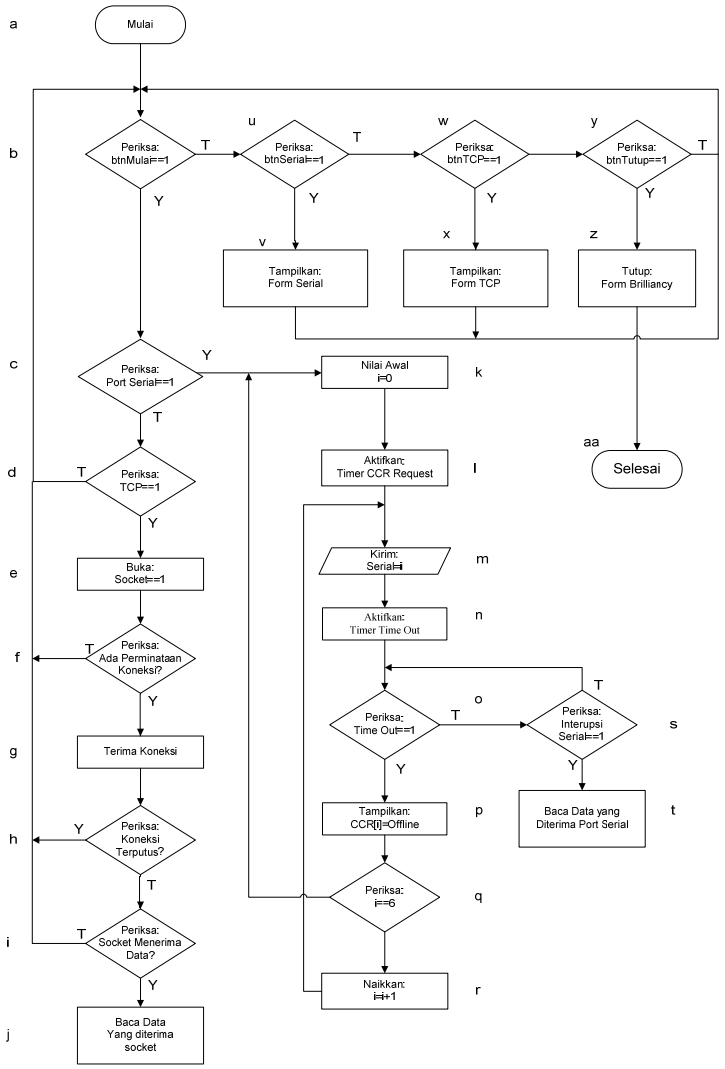
Form TCP/IP digunakan agar aplikasi AFL Simulator dapat berkomunikasi melalui jaringan sesuai dengan konfigurasi nomor *port* yang digunakan oleh pengguna, form ini memiliki alur program seperti yang terlihat pada Gambar 3.25. Alur programnya dijelaskan sebagai berikut:

a. Awal program dijalankan.
b. Program akan menjalankan langkah f jika pengguna memilih btnOK.

c. Program akan menjalankan langkah d jika pengguna memilih btnClose.
d. Program menutup objek Socket dan melanjutkan ke langkah e.
e. Program menutup form dirinya sendiri dan melanjutkan ke langkah g.
f. Program membuka objek Socket dan melanjutkan ke langkah e.
g. Akhir program.

5. Diagram alur program pada form Brilliancy.

Form Brilliancy merupakan form inti dari aplikasi AFL Simulator, form ini mensimulasikan tingkat kecerahan lampu AFL dan menampilkan terjadinya kesalahan yang terjadi pada CCR Simulator. Pada Gambar 3.26 merupakan alur progam yang akan dirancang untuk form Brilliancy.



Gambar 3.26. Diagram Alur Form Brilliancy

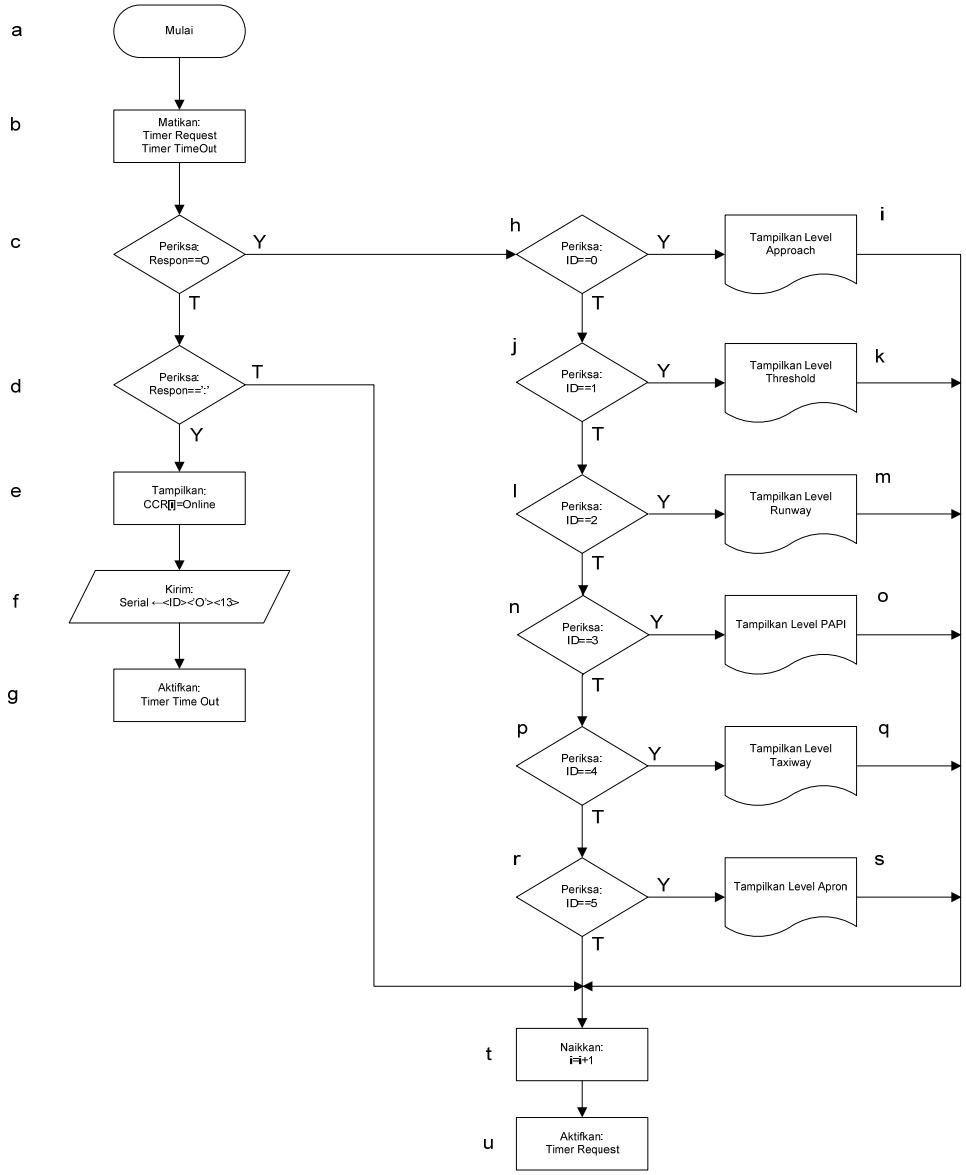
Alur program pada form Brilliancy dijelaskan sebagai berikut:

a. Awal program utama dijalankan.
b. Jika tombol btnMulai dipilih oleh pengguna maka program menjalankan langkah c, jika tidak program akan memeriksa tombol btnSerial pada langkah u.
c. Program memeriksa *port* serial, jika *port* serial sudah terbuka maka program memeriksa objek *port* serial dan *socket* pada langkah c dan langkah d, jika *port* serial terbuka maka program menjalankan langkah k.
d. Program memeriksa objek Socket, jika *port* sudah terbuka maka program melanjutkan ke langkah e, jika belum maka program kembali ke langkah a.
e. Program mendengarkan permintaan koneksi dari komputer lain.
f. Jika ada permintaan koneksi maka program melanjutkan ke langkah g.
g. Program menerima koneksi.
h. Program memeriksa status koneksi, jika koneksi terputus maka program kembali ke langkah b.
i. Jika objek Socket menerima data maka program akan melanjutkan ke langkah j.
j. Program membaca data yang diterima oleh objek Socket dan menjalankan alur program interupsi port TCP/IP.
k. Program menolkan variabel i, sebagai identitas CCR Simulator yang ditunjuk.
l. Program menghidupkan objek Timer yang berfungsi sebagai pencarian CCR Simulator dan menjalankan langkah m.
m. Program mengirimkan identitas CCR Simulator yang ditunjuk oleh variabel i dan melanjutkan ke langkah n.
n. Program menghidupkan timer yang berfungsi sebagai waktu tunda untuk menunggu respon dari CCR Simulator dan melanjutkan ke langkah o.
o. Program membandingkan nilai objek Timer dengan nilai *interval*, jika nilai waktu tunda sama dengan nilai *interval* maka program menjalankan

langkah p, jika nilai waktu tunda tidak sama dengan nilai *interval* maka program memeriksa interupsi pada langkah s.

p. Program menampilkan CCR Simulator dengan identitas yang ditunjuk oleh variabel i sedang tidak aktif (offline), dan menjalankan ke langkah q.
q. Program memeriksa nilai identitas yang disimpan pada variabel i, jika i=5 maka program kembali ke langkah k, jika tidak sama dengan 5 maka program menjalankan langkah r.
r. Program menambahkan variabel i dengan 1 untuk menunjuk identitas CCR Simulator berikutnya dan kembali ke langkah l.
s. Jika terjadi interupsi *port* serial maka program melanjutkan ke langkah t.
t. Program menjalankan pembacaan penerimaan data pada alur program interupsi *port* serial.
u. Jika pengguna memilih tombol btnSerial maka program menjalankan langkah langkah v, jika tidak maka program memeriksa objek tombol btnTCP pada langkah w.
v. Program menampilkan form komunikasi Serial.
w. Jika pengguna memilih tombol btnTCP maka program melanjutkan ke langkah x, jika tidak maka program memeriksa tombol btnTutup pada langkah y.
x. Program menampilkan form komunikasi TCP/IP.
y. Jika pengguna memilih tombol btnTutup maka program melanjutkan ke langkah z, jika tidak maka program kembali kelangkah b.
z. Program menutup form dirinya sendiri dan melanjutkan ke langkah aa.
aa. Akhir program.

Pada komunikasi *port* serial, objek MsComm dengan properti RThrishold di-*set* 1, hal ini akan mengaktifkan interupsi *port* serial. Pada saat AFL Simulator menerima data dari port serial maka program akan menjalankan program interupsi *port* serial. Adapun diagram alur program interupsi serial seperti terlihat pada Gambar 3.27.



Gambar 3.27. Diagaram Alur Interupsi Port Serial

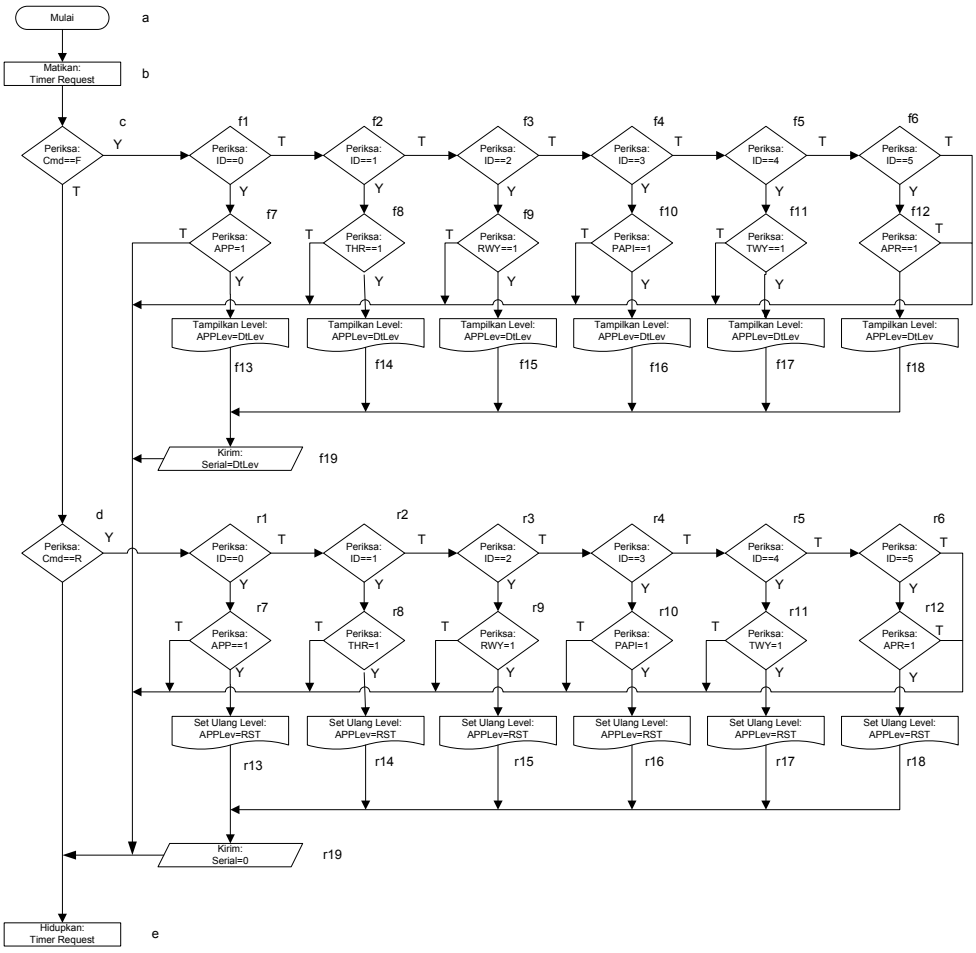
Penjelasan digaram alur interupsi *port* serial yaitu sebagai berikut:

a. Awal program interupsi *port* serial dijalankan.
b. Program mematikan objek Timer yang berfungsi sebagai pencarian CCR Simulator dan melanjutkan ke langkah c.
c. Program membandingkan data hasil dari respon CCR Simulator, jika diterima karakter ‘O’ maka program memeriksa identitas CCR Simulator pada langkah h, jika bukan maka program memeriksa karakter pada langkah d.

d. Jika program menerima karakter *colon* ‘:’ maka program menjalankan ke langkah e, jika bukan maka program menjalankan langkah s.
e. Program menampilkan status CCR dengan identitas i sedang aktif, dan melanjutkan ke langkah f.
f. Program mengirimkan perintah status CCR Simulator dengan identitas i, dan melanjutkan ke langkah g.
g. Program menjalankan waktu tunda, untuk menunggu respon dari CCR Simulator yang dijalankan di alur program utama.
h. Jika identitas 0 maka program menjalankan langkah i, jika bukan maka program memeriksa identitas CCR Simulator pada langkah j.
i. Program menampilkan level lampu *Approach*.
j. Jika identitas yang diterima adalah 1, maka program menjalankan langkah k, jika bukan maka program melakukan pemeriksaan identitas pada langkah l.
k. Program menampilkan level lampu *Threshold*.
l. Jika identitas yang diterima adalah 2 maka program menjalankan langkah m, jika tidak sesuai maka program memeriksa identitas pada langkah n.
m. Program menampilkan level lampu *Runway*.
n. Jika identitas yang diterima adalah 3 maka program menjalankan langkah o, jika tidak sama maka program memeriksa alamat pada langkah p.
o. Program menampilkan level lampu PAPI.
p. Jika identitas sama dengan 4 maka program menjalankan langkah q, jika tidak sama maka program memeriksa alamat pada langkah r.
q. Program menampilkan level lampu *Taxiway*.
r. Jika identitas sama dengan 5 maka program menjalankan langkh s, jika tidak sama maka program menjalankan langkah t.
s. Program menampilkan level lampu *Apron*.
t. Program menambahkan variabel i dengan 1, untuk menunjuk identitas CCR Simulator berikutnya, dan menjalankan langkah u.

u. Program mengaktifkan kembali objek Timer Request untuk meminta status CCR Simulator dengan identias berikutnya.

Selain interupsi *port* serial, form Brilliancy juga melayani interupsi port TCP/IP, program interupsi ini akan dijalankan jika *port* TCP/IP menerima data. Pada Gambar 3.28 merupakan diagram alur interupsi *port* TCP/IP.



Gambar 3.28. Diagram Alur Interupsi Port TCP/IP

Berikut adalah penjelasan diagram alur interupsi *port* TCP/IP.

a. Awal program.
b. Program mematikan objek Timer Request.
c. Program membandingkan perintah yang diterima oleh objek Winsock, jika perintah berupa karakter F maka program akan menjalankan langkah f1 sampai dengan f19 untuk mensimulasikan kerusakan lampu

AFL, sesuai dengan identias CCR Simulator yang ditunjuk. Jika perintah yang diterima bukan karakter ‘F’ maka program akan memeriksa perintah pada langkah d.

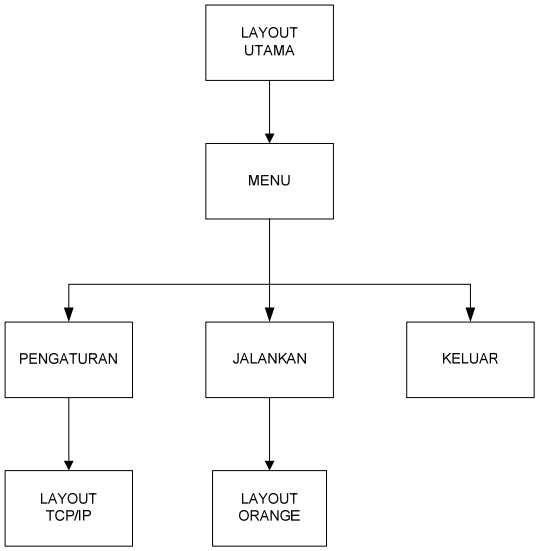
d. Program memeriksa perintah yang diterima dengan karakter ‘R’, jika perintah sama, maka program akan menjalankan langkah r1 sampai dengan r19. Jika perintah tidak sama dengan karakter ‘R’, maka program akan menjalankan langkah e.
e. Program mengaktifkan kembali objek Timer Request untuk meminta status CCR Simulator berikutnya, langkah ini akan membuat program menjalankan kembali program utama (keluar dari interupsi *port* TCP/IP. Untuk penerimaan perintah dengan karakter ‘F’ akan membuat program mengirimkan data kerusakan lampu pada CCR Simulator, dimana data kerusakan lampu akan dikirimkan sesuai dengan kesamaan identitas CCR Simulator pada langkah f1 sampai dengan f6. Sedangkan penerimaan perintah ‘R’ akan membuat program me-*reset* tampilan kerusakan lampu AFL dan me-reset CCR Simulator dengan mengirimkan data 00 dengan identitas CCR Simulator akan dijalankan pada langkah r1 sampai dengan r6.

3.6 Perancangan Program Aplikasi AFL Remote

Pada subbab ini menjelaskan perancangan program aplikasi AFL Remote pada tablet PC, aplikasi digunakan untuk mengirimkan data simulasi kerusakan lampu AFL. Adapun perancangannya meliputi struktur navigasi, perancangan layout, dan diagram alur masing-masing layout.

3.6.1 Perancangan Struktur Navigasi AFL Remote

Struktur navigasi program aplikasi AFL Remote yang akan dirancang seperti terlihat pada Gambar 3.29.



Gambar 3.29. Perancangan Struktur Navigasi AFL Remote

Layout Utama merupakan layout pertamakali ditampilkan, layout ini memiliki menu utama untuk menampilkan *contex* menu. Terdapat tiga buah *contex* menu yaitu menu Pengaturan, Jalankan, dan Keluar. Menu Pengaturan akan menampilkan layout TCP/IP yang berfungsi untuk assign IP yang digunakan melakukan remote terhadap AFL Simulator.

Sedangkan menu Jalankan akan menampilkan layout Orange, layout ini merupakan layout untuk mengirimkan data simulasi kerusakan dengan memilih tombol-tombol tipe kerusakan. Menu Keluar digunakan untuk keluar dari program aplikasi AFL Remote.

3.6.2 Perancangan Layout AFL Remote

1. Tampilan layout Utama.

   Perancangan tampilan *layout* Utama pada aplikasi AFL Remote seperti yang terlihat pada Gambar 3.30.



Gambar 3.30. Perancangan Tampilan Layout Utama

Pada layout Utama terdapat tiga buah menu yang berfungsi sebagai berikut:

a. Pengaturan: berfungsi untuk menampilkan *layout* TCP/IP.
b. Jalankan: berfungsi untuk menampilkan *layout* Orange.
c. Keluar: berfungsi untuk keluar dari program aplikasi AFL Remote.

Pada Tabel 3.11 merupakan objek dan atribut yang terdapat pada *layout* Utama.

Tabel 3.11. Objek dan Nilai Atribut Layout Utama

Objek	Atribut	Nilai
LinearLayout	Layout_Width	Fill_parent
	Layout_Heigh	Fill_parent
Label	Name.Id	@+id/Lbl1
	Text	AFL Remote
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
Image	Layout_Width	Fill_Parent
	Layout_Heigh	Fill_Parent
	Name	-
Menu	Index	0
	Name	Menu
	Text	PENGATURAN
	Index	1
	Name	Menu
	Text	JALANKAN
	Index	2
	Name	Menu
	Text	KELUAR

Objek LinearLayout merupakan objek yang menjadi dasar objek gambar ditampilkan, sedangkan objek label merupakan judul dari porgram aplikasi AFL Remote.

2. Tampilan layout TCP/IP.

   Layout TCP/IP merupakan *layout* yang berfungsi untuk mengonfigurasi TCP/IP disesuaikan dengan IP yang dimiliki oleh AFL Simulator. Pada Gambar 3.31 merupakan perancangan *layout* TCP/IP.



Gambar 3.31. Perancangan Layout TCP/IP

Objek yang terdapat pada *layout* Pengaturan dan atribut dari masing-masing objek seperti terlihat pada Tabel 3.12.

Tabel 3.12. Objek dan Nilai Atribut Layout TCP/IP

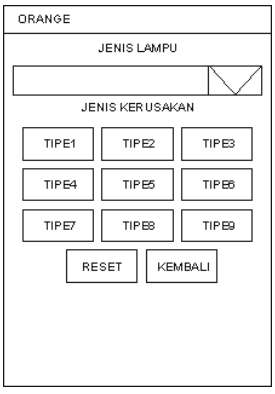
Objek	Atribut	Nilai
LinearLayout	Layout_Width	Fill_parent
	Layout_Heigh	Fill_parent
Label	Name	@+id/LblConfig
	Text	Pengaturan
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
EditText	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+/etIP
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+/etPORT
Button	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content

Tabel 3.13. Lanjutan Objek dan Nilai Atribut Layout TCP/IP

Objek	Atribut	Nilai
Button	Name	@+/btnTestConn (Button1)
	Text	Test Connection
	Layout_Width	Wrap_Content
	Layout_Heigh	Wrap_Content
	Name	@+id/btnOK (Button2)
	Text	OK
	Layout_Width	Wrap_Content
	Layout_Height	Wrap_Content
	Name	@+id/btnClose (Button3)
	Text	Close
TextView	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/tvConRes
	Text	-

3. Tampilan layout Orange.

   Layout Orange akan ditampilkan ketika pengguna memilih menu Jalankan, *layout* ini terdiri dari sembilan jenis kerusakan. Sebuah objek Spinner digunakan untuk kumpulan nama-nama jenis lampu AFL.. Pada Gambar 3.32 merupakan perancangan *layout* Orange yang akan dirancang.



Gambar 3.32. Perancangan Layout Orange

Untuk mengirimkan data simulasi kerusakan lampu AFL, maka nama dari jenis lampu tersebut harus dipilih terlebih dahulu, pemilihan jenis lampu dilakukan pada objek Spinner. Sedangakan untuk mengirimkan jenis kerusakan dilakukan dengan penekanan tombol mulai dari TIPE 1 sampai dengan TIPE 9, sedangkan tombol RESET digunakan untuk mengembalikan kondisi lampu AFL dalam keadaan baik. Untuk kembali ke layout awal

(layout Utama) yaitu dengan mengklik tombol KEMBALI. Pada Tabel 3.14 merupakan penjelasan objek dan nilai atribut dari masing-masing objek yang terdapat pada *layout* Orange.

Tabel 3.14. Objek dan Nilai Atribut Layout Orange

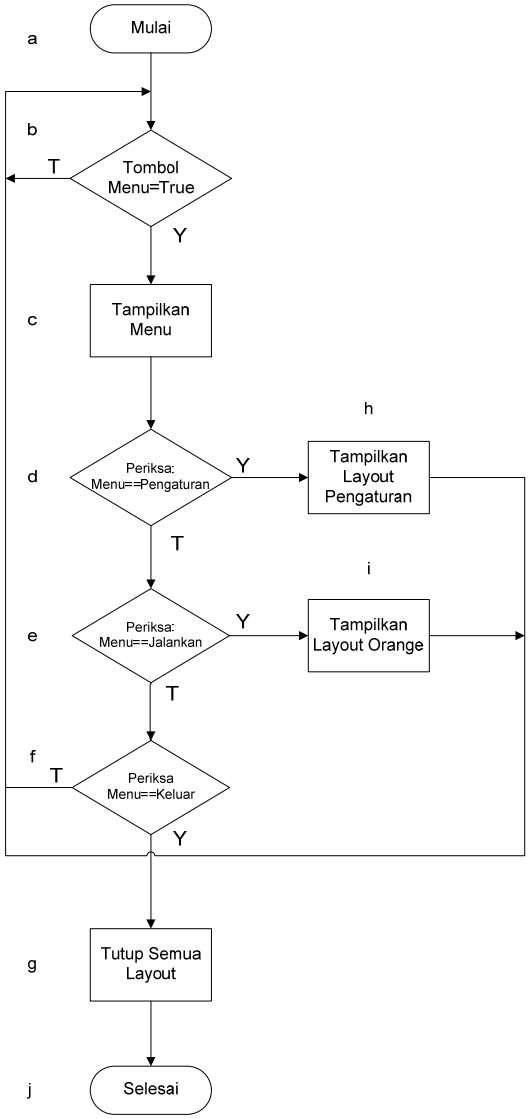
Objek	Atribut	Nilai
Linear Layout	Layout_Width	Fill_Parent
	Layout_Heigh	Fill_Parent
Label	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Text	ORANGE
TableLayout	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
TableRow	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
Button	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg1 (Button1)
	Text	TIPE 1
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg2 (Button2)
	Text	TIPE 2
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg3 (Button3)
	Text	TIPE 3
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg4 (Button4)
	Text	TIPE 4
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg5 (Button5)
	Text	TIPE 5
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg6 (Button6)
	Text	TIPE 6
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg7 (Button7)
	Text	TIPE 7
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content

Tabel 3.15. Lanjutan Objek dan Nilai Atribut Layout Orange

Button	Name	@+id/btnYorg8 (Button8)
	Text	TIPE 8
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnYorg9 (Button9)
	Text	TIPE 9
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnRstYorg (Button13)
	Text	RESET
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/btnBackAirport (Button14)
	Text	KEMBALI
Spinner	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Name	@+id/spnYorg
TextView	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Text	JENIS LAMPU
	Layout_Width	Fill_Parent
	Layout_Heigh	Wrap_Content
	Text	JENIS KESALAHAN

3.6.3 Diagram Alur Program AFL Remote

1. Diagram alur layout Utama.



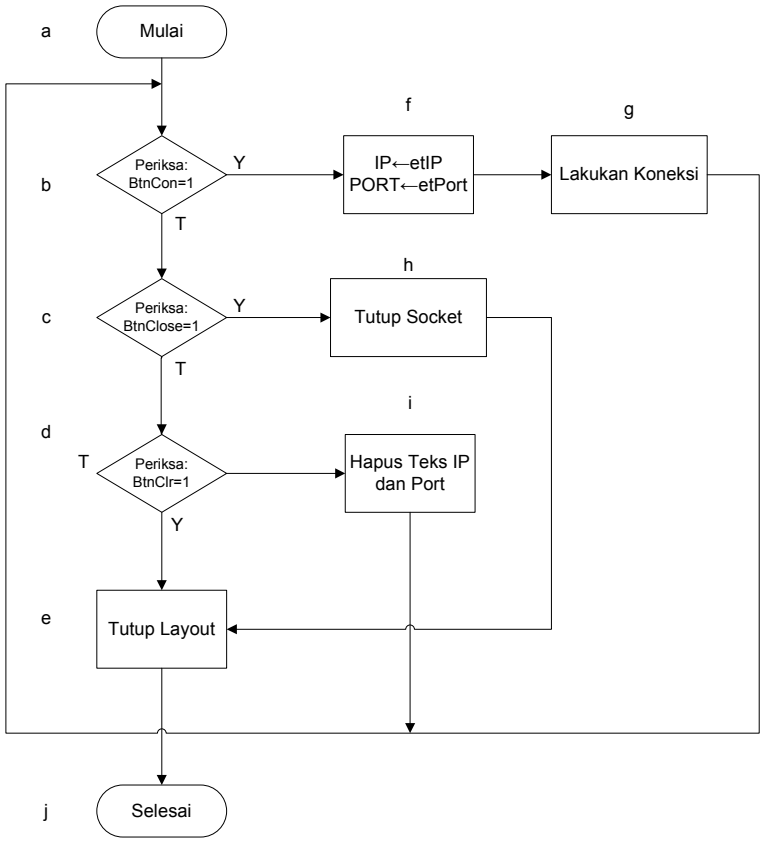
Gambar 3.33. Diagram Alur Layout Utama

Alur program *layout* Utama seperti terlihat pada Gambar 3.33 dijelaskan sebagai berikut berikut penjelesannya:

a. Awal program dijalankan.
b. Program menunggu pengguna menekan tombol menu.
c. Program menampilkan menu dan menunggu pengguna untuk memilih salah satu menu tersebut.

d. Program memeriksa menu yang diplih oleh pengguna, jika menu Pengaturan terpilih maka program akan melanjutkan ke langkah h, jika tidak maka program akan memeriksa menu pada langkah e.
e. Program memeriksa menu Jalankan, jika menu ini terpilih maka program akan melanjutkan ke langkah I, jika tidak maka program memeriksa menu pada langah f.
f. Program memeriksa menu Keluar, jika menu ini terpilih maka program akan ke langkah g, jika tidak maka program akan kembali memeriksa menu dari awal.
g. Program mentutup semua layout dan melanjutkan ke langkah i.
h. Program menampilkan *layout* Pengaturan.
i. Program menampilkan *layout* Orange.
j. Akhir program.

2. Diagram alur layout TCP/IP.

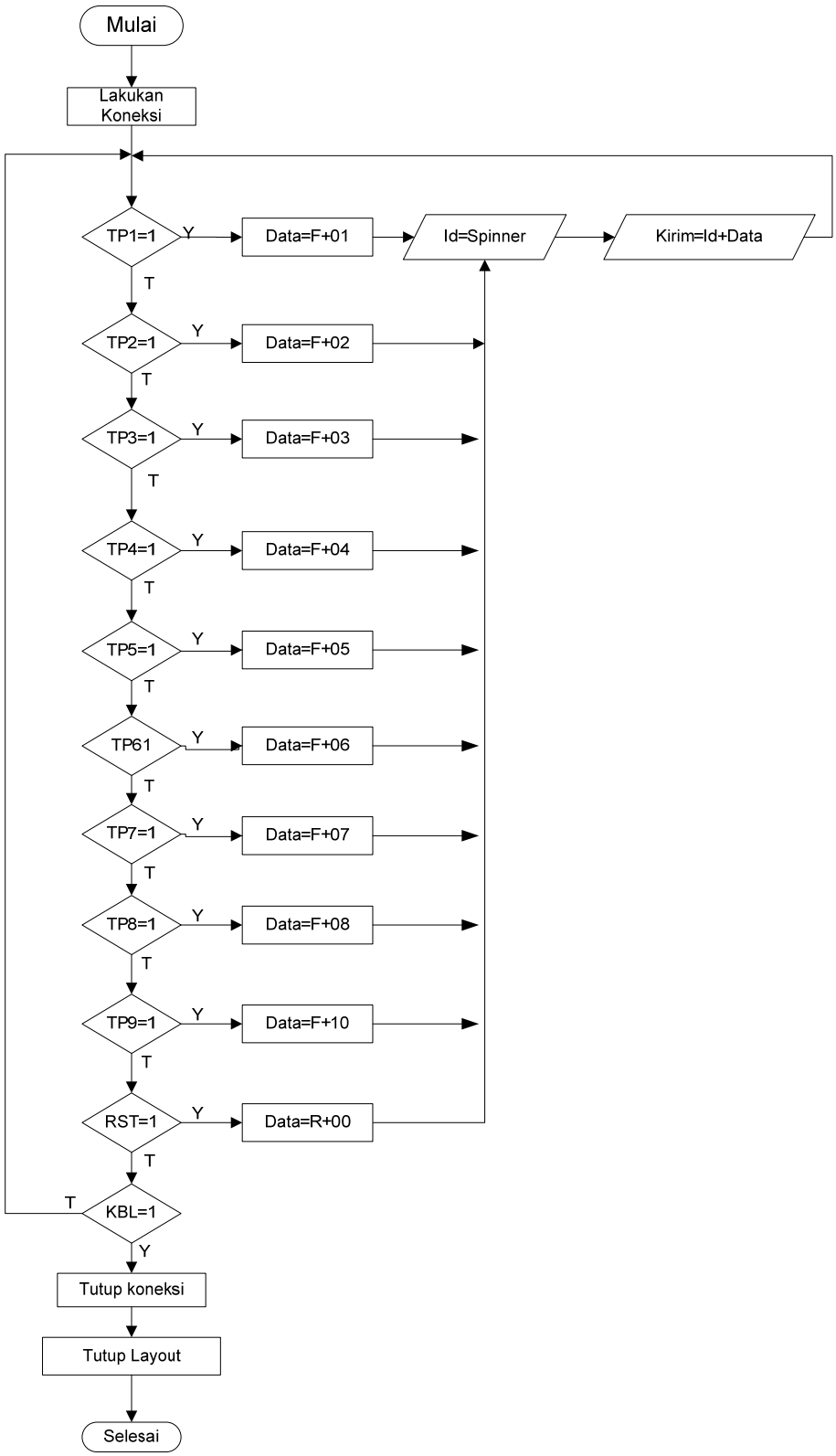


Gambar 3.34. Diagram Alur Layout TCP/IP

Layout Pengaturan yang akan dirancang memiliki alur program seperti terlihat pada Gambar 3.35. Berikut penjelesan alur program *layout* Pengaturan.

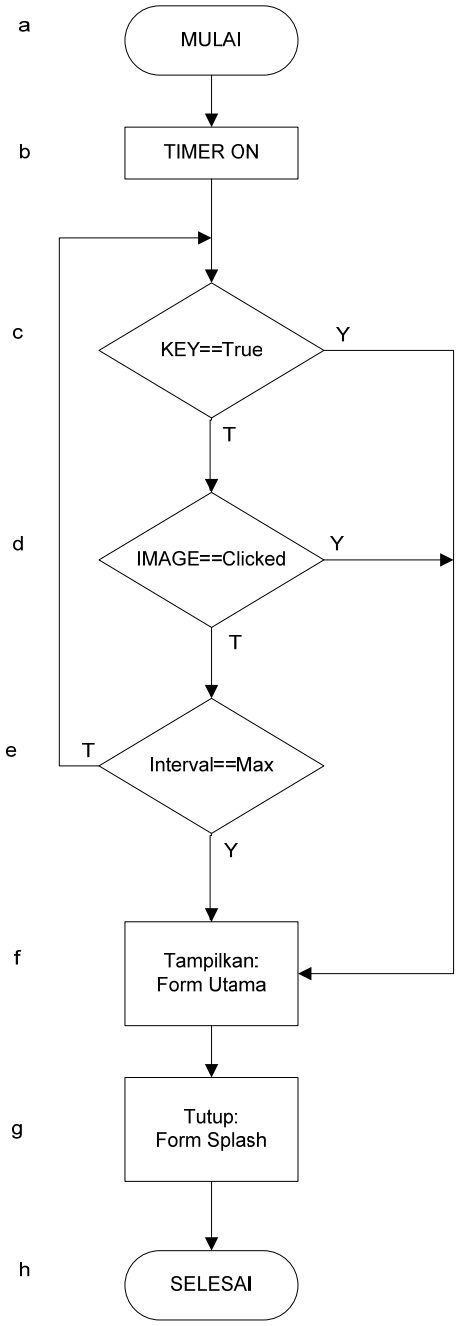
a. Awal program dijalankan.
b. Jika tombol BtnCon dipilih maka program akan melanjukan ke langkah f jika tidak maka program akan memeriksa tombol berikutnya yang dilakukan pada langkah c.
c. Jika tombol BtnClose dipilih maka program akan melanjutkan ke langkah h, jika tidak maka program akan memeriksa tombol BtnClr pada langkah d.
d. Jika tombol BtnClr dipilih maka program akan melanjutkan ke langkah i, jika tidak maka pemeriksaan tombol diulang kembali pada langkah b.
e. Program menutup layout TCP dan melanjutkan ke langkah j.
f. Program membaca IP dan nomor port pada objek Text, kemudian melanjutkan ke langkah g.
g. Program melakukan koneksi TCP/IP dan kembali ke langkah b.
h. Program menutup atau memutuskan koneksi TCP/IP.
i. Program membersikan teks.
j. Akhir program.

3. Diagram alur layout Orange.



Gambar 3.35. Diagram Alur Layout Orange

Layout Orange memiliki diagram alur seperti terlihat pada Gambar 3.35. ketika *Layout* Orange ditampilkan maka objek Socket akan terkoneksi melakukan koneksi dengan AFL Simulator, *layout* Orange menunggu salah satu objek Button ditekan oleh pengguna. Objek Button pada *layout* Orange menyediakan 9 jenis simulasi kerusakan lampu AFL, masing-masing objek Button berisi satu data kerusakan lampu AFL yang akan dikirimkan ke AFL simulator. Ketika salah satu objek Button ditekan, maka akan dikirimkan data keruskan yang disertakan dengan jenis lampu yang dituju untuk mensimulasikan kerusakan lampu AFL. Jenis lampu tujuan ditentukan dengan pemilihan lampu yang disediakan oleh objek Spiner. Tombol reset merupakan tombol untuk me-reset simulasi kerusakan lampu AFL yang telah dikirimkan sebelumnya, dengan demikian lampu yang ditampilkan oleh AFL Simulator dan besar arus pada CCR Simulator dapat dikendalikan kembali oleh peserta latihan AFL. Jika pengguna mengklik tombol kembali maka layou AFL Remote akan memutuskan koneksi dengan AFL Simulator dan menutup layout Orange.


a
MULAI
b
TIMER ON
c
KEY==True
Y
T
d
IMAGE==Clicked
Y
T
e
T
Interval==Max
Y
f
Tampilkan:
Form Utama
g
Tutup:
Form Splash
h
SELESAI