

**LAPORAN AKHIR
PENELITIAN MANDIRI**



**PERANCANGAN SISTEM KEAMANAN PADA SEPEDA
MOTOR BERBASIS ANDROID**

TIM PENGUSUL :

Sukandar, S.Pd, M.T NIDN : 0618087402

UNIVERSITAS SULTAN FATAH (UNISFAT) DEMAK

2022

LEMBAR PENGESAHAN USUL PENELITIAN

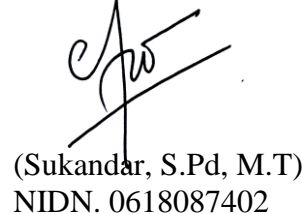
1. a. **Judul Penelitian** : Perancangan Sistem Keamanan Sepeda Motor berbasis Android
- b. Bidang Ilmu : Teknik Elektro
2. **Ketua Peneliti** :
- a. Nama Lengkap dan Gelar : Sukandar, S.Pd, M.T
- b. Jenis Kelamin : Laki-laki
- c. NIDN : 0618087402
- d. Jabatan Fungsional : Asisten Ahli
- e. Jabatan Struktural : -
- f. Fakultas/Jurusan : Teknik/Teknik Elektro
- g. Lembaga Penelitian : Universitas Sultan Fatah Demak
3. **Jumlah Anggota Peneliti** :
- a. Nama Anggota Peneliti I :
- b. Nama Anggota Peneliti II :
- c. Nama Anggota Peneliti III :
4. **Lokasi Penelitian** : Kabupaten Demak
5. **Kerjasama dengan Institusi lain** :
- a. Nama Institusi : -
- b. Alamat : -
- c. Telepon/Faks/e-mail : -
6. **Lama Penelitian** : 8 bulan
7. **Biaya yang diperlukan** :
- a. Sumber dari P3M UNISFAT : Rp. 2.500.000,-
- b. Sumber dari Dikti : Rp. -
- Jumlah : Rp. 2.500.000,-
(Tujuh Juta Lima Ratus Ribu Rupiah)

Mengetahui :
Dekan Fakultas Teknik



(Ahmad Nuruddin S., S.Kom, M.Kom)
NIDN. 06-3112-7803

Demak, 6 Februari 2022
Ketua Peneliti,



(Sukandar, S.Pd, M.T)
NIDN. 0618087402

Menyetujui,

Ketua P3M UNISFAT



(Drs. Nor Kholis, M.Pd)

NIDN. 0604096001

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas hidayah, inayah serta nikmat-Nya sehingga kami dapat menyelesaikan laporan ini.

Selama menyelesaikan laporan ini, kami banyak mendapatkan saran, petunjuk, dukungan serta bimbingan dari berbagai pihak. Untuk itu kami hanya dapat mengucapkan banyak terima kasih dari hati yang paling dalam kepada :

1. Bapak Dr. Mohhammad Kusyanto, MT. selaku Rektor Unisfat Demak yang telah memberi arahan kepada kami untuk menyelesaikan laporan ini.
2. Bapak Drs. Nor Kholis, M.Pd selaku Ketua P3M Unisfat yang telah memberi dorongan semangat kepada kami untuk menyelesaikan laporan ini.
3. Segenap civitas akademi Unisfat Demak atas dukungan dan doanya.
4. Semua pihak yang telah membantu penulis dalam menyelesaikan laporan ini.

Kami menyadari bahwa masih banyak kekurangan dalam laporan ini, untuk itu kami berharap segala partisipasi semua pihak untuk memberikan masukan, saran dan kritik yang membangun demi sempurnanya laporan ini.

Akhirnya kami mendedikasikan laporan ini untuk seluruh insan pecinta ilmu pengetahuan dan semoga laporan ini dapat bermanfaat bagi kemajuan ilmu pengetahuan. Amiin.

Demak, 7 Februari 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
LAPORAN PENELITIAN	1
I Pendahuluan.....	1
II Permasalahan	2
III Metode Penelitian	12
IV Analisis dan Pembahasan	13
V Kesimpulan	17
DAFTAR PUSTAKA	

BAB I

PENDAHULUAN.

Android merupakan platform lengkap mulai dari Sistem Operasi, Aplikasi, Tool Developing, Market Aplikasi, dukungan vendor industri motore, bahkan dukungan dari komunitas Open System. Tentu ini merupakan keunggulan yang tidak dimiliki oleh platform lain. Dengan melihat perkembangan saat ini, Android telah menjadi kekuatan yang luar biasa. Pada tahun 2009, dilaporkan oleh canalsy Estimates, market smartphone untuk Android tumbuh 1073,5% disaat platform lain tidak ada yang mencapai pertumbuhan 100 %.

Semakin mewabahnya platform Android membuat aplikasi Android pun semakin menjamur. Para developer pun berbondong-bondong membuat aplikasi Androidnya. Tiap harinya mungkin ratusan bahkan ribuan aplikasi baru disubmit ke Google Play Store. Tentunya dengan berbagai kategori dan target pengguna (Sumber: pusatteknologi.com, 2016).

Memiliki kendaraan tentu menjadi impian setiap orang baik itu motor, mobil atau lainnya. Keinginan tersebut kadang tidak diimbangi dengan dana yang dimiliki oleh setiap orang, sehingga banyak yang hanya bisa bermimpi memiliki kendaraan pribadi namun amat sulit memilikinya.

Tingginya gengsi hingga dan rendahnya tingkat ekonomi kadang menggelapkan mata seseorang sehingga memilih untuk mencuri kendaraan milik orang lain. Hal tersebut memang sudah bukan hal yang baru sehingga pastinya anda harus waspada selalu dengan ancaman kejahatan pada kendaraan kesayangan anda. Untuk itu perlu kiranya sebuah sistem keamanan tambahan pada sepeda motor yang pastinya akan sangat bermanfaat untuk menjaga keamanan motor kesayangan anda dari nakalnya tangan para pencuri.

Dalam penelitian ini, akan direalisasikan sebuah sistem keamanan tambahan pada sepeda motor dengan memanfaatkan handphone Android dengan judul **“Sistem Keamanan Pada Sepeda Motor Berbasis Android”**.

BAB II

PERMASALAHAN

Tujuan

Merancang sebuah sistem keamanan tambahan untuk motor dengan memanfaatkan handphone Android yang penggunaannya sudah bersifat umum, dimana setiap orang memilikinya, dengan catatan tanpa merubah fungsi dasar dari handphone Android tersebut.

Manfaat Penelitian

- a. Keamanan kendaraan akan lebih terjamin, karena mempunyai sistem keamanan ganda, sehingga pengemudi mempunyai rasa aman ketika meninggalkan motornya.
- b. Mengurangi kasus tindakan pencurian motor..

Landasan Teori

2.1 Mikrokontroler

2.1.1 Pengetahuan Dasar Mikrokontroler

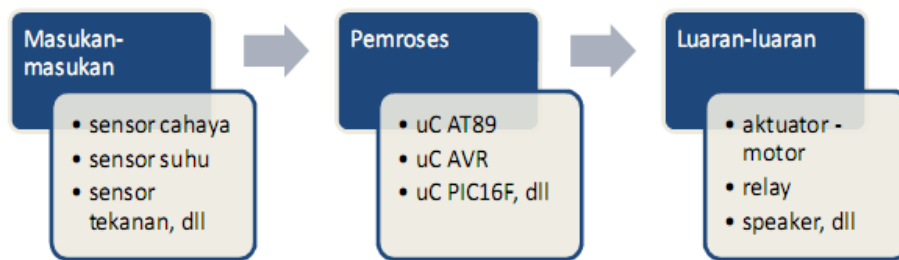
Mikrokontroler sebagai suatu terobosan teknologi *mikroprosesor* dan mikrokomputer, hadir memenuhi kebutuhan pasar dan teknologi baru. Sebagai teknologi baru, yaitu teknologi *semi konduktor* dengan kandungan *transistor* yang lebih banyak namun hanya membutuhkan ruangan yang lebih kecil serta diproduksi secara massal (dalam jumlah banyak) sehingga harganya menjadi lebih murah (dibanding *mikroprosesor*).

Jika kita bicara tentang *Mikrokontroler*, maka tidak terlepas dari pengertian atau definisi tentang komputer itu sendiri

Kesamaan-kesamaan antara *Mikrokontroler* dengan komputer. antara lain:

1. Sama-sama memiliki unit pengolah pusat atau yang lebih dikenal dengan CPU (*Central Processing Unit*).
2. CPU tersebut sama-sama menjalankan program dari suatu lokasi atau tempat, biasanya dari ROM (*Read Only Memory*) atau RAM (*Random Access Memory*).

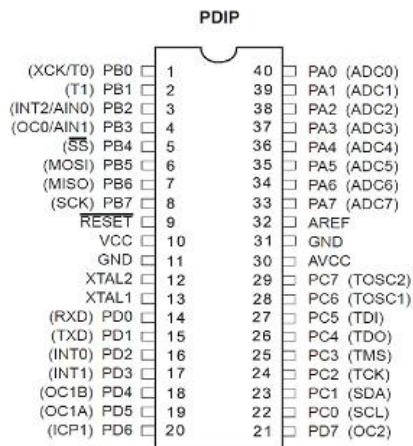
3. Sama-sama memiliki RAM yang digunakan untuk menyimpan data-data sementara atau yang lebih dikenal dengan variabel-variabel.
4. Sama-sama memiliki beberapa luaran dan masukan (I/O) yang digunakan untuk melakukan komunikasi timbal-balik dengan dunia luar, melalui sensor (masukan) dan aktuator (luaran).



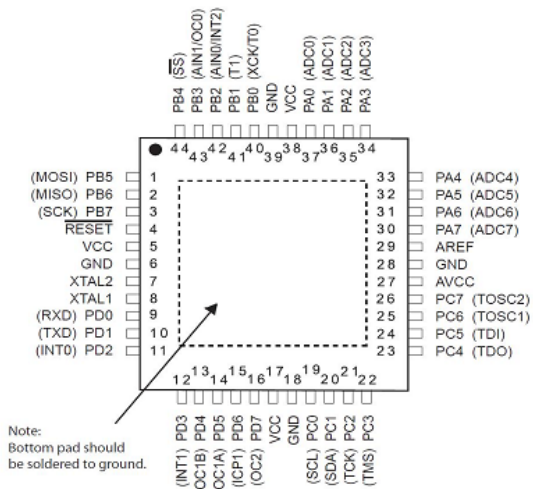
Gambar 2.1. Bagan masukan, Pemrosesan hingga Luaran

Mikrokontroler AVR ATMEGA16

Salah satu *jenis* mikrokontroler keluarga AVR yang populer adalah seri AT-MEGA 16, Pada Gambar 2.4 dan 2.5 ditunjukkan diagram pin, masing-masing, untuk Mikrokontroler AVR ATMEGA16 tipe PDIP dan TQFP/MLF atau dikenal sebagai SMD.



Gambar 2.4. Diagram Pin Mikrokontroler AVR ATMEGA16 tipe PDIP



Gambar 2.5. Diagram Pin Mikrokontroler AVR ATMEGA16 tipe SMD

Bahasa Pemrograman BASIC AVR (BASCOM AVR)

Banyak *sekali* bahasa pemrograman yang dapat digunakan dalam menuliskan program ke *mikrokontroler*, salah satunya adalah bahasa BASIC. Penggunaan bahasa ini mempunyai kemudahan dalam memprogram dan adanya fasilitas *simulator* pada *Compiler* BASCOM AVR.

2.2.1. Tipe Data

Tipe data berkaitan dengan perubah atau variabel atau konstanta yang akan menunjukkan daya tampung/jangkauan dari variabel/konstanta tersebut. Tipe data dalam BASCOM ditunjukkan pada Tabel 2.4.

Tabel 2.4. Tipe Data dan Ukurannya

Tipe Data	Ukuran (Byte)	Jangkauan Data
Bit	1/8	0 atau 1
Byte	1	0 s/d 255
Integer	2	-32768 s/d 3.767
Word	2	0 s/d 65535
Long	4	-2147483648 s/d 2147483647
Single	4	$1,5 \times 10^{-45}$ s/d $3,4 \times 10^{38}$
Double	8	5×10^{-324} s/d $1,7 \times 10^{308}$
String	s/d 254	

2.2.2. Variabel

Variabel atau peubah digunakan untuk menyimpan data sementara. *Variabel* diberi nama dan dideklarasikan terlebih dahulu sebelum digunakan. Aturan pemberian nama variabel sebagai berikut:

- a. Harus dimulai dengan huruf (bukan angka).
- b. Tidak ada nama variabel yang sama dalam sebuah program.
- c. Maksimum 32 karakter
- d. Tanpa menggunakan spasi, pemisahan bisa dilakukan dengan garis bawah.
- e. Tidak menggunakan karakter-karakter khusus yang digunakan sebagai operator
BASCOM

Variabel dapat dideklarasikan dengan beberapa cara :

1. Dengan pernyataan DIM

Deklarasi ini dibuat dengan perintah DIM (singkatan dari dimension) dengan aturan sebagai berikut:

Dim <NamaVariabel> As <TipeData>

Contoh :

Dim angka As Integer

Dim bilangan As byte

Jika beberapa variabel dideklarasikan dalam satu baris, maka harus dipisah dengan tanda koma.

Contoh:

Dim angka As Integer, bilangan As byte

2. Dengan pernyataan DEFINT, DEFBIT, DEFBYTE, DEFWORD

Deklarasi dengan pernyataan tersebut secara prinsip tidak berbeda dengan "DIM", perhatikan keterangan dari masing-masing pendeklarasian tersebut:

DEFINT = untuk tipe data integer,

DEFBIT = untuk tipe data bit,

DEFBYTE = untuk tipe data byte,

DEFWORD = untuk tipe data word,

DEFLNG = untuk tipe data long,
DEFSNG = untuk tipe data singel,
DEFDBL = untuk tipe data double.

Cara pendeklarasiannya sebagai berikut:

DEFINT/DEFBIT/DEFBYTE/DEFWORD <variabel>

Contoh :

DEFINT angka

DEFBYTE bilangan

Untuk variabel dengan tipe data yang sama dapat dideklarasikan dengan dipisah titik koma, misal:

DEFINT bil_1 ; bil_2 ; bil_3

2.2.3. Konstanta

Berbeda dengan variabel, sebuah konstanta akan bernilai tetap. Sebelum digunakan, konstanta dideklarasikan terlebih dulu dengan cara (ada dua cara):

Dim nama_konstanta As const nilai_konstanta

Const nama_konstanta = nilai_konstanta

Contoh :

Dim pembagi as const 23

Const pembagi = 23

2.2.4. Penulisan Bilangan

Pada *BASCOM-AVR*, bilangan dapat ditulis dalam 3 bentuk :

1. **Desimal ditulis biasa, contoh : 16**
2. **Biner diawali dengan &B, contoh : &B10001111**
3. **Heksadesimal diawali dengan &H, contoh : &H8F**

2.2.5. Alias

Untuk mempermudah pemrograman, biasanya nama register *dalam* mikrokontroler dibuatkan nama yang identik dengan hardware yang dibuat, contoh :

LED_1 alias PORTC.O ‘ nama lain dari PORTC.O adalah LED_1

SW_1 alias PINC.1 ‘ nama lain dari PINC.1 adalah SW_1

2.2.6. Array atau Larik

Array atau larik merupakan sekumpulan variabel dengan nama dan tipe yang sama, yang berbeda indeks keanggotaannya.

Cara mendeklarasikan array sebagai berikut:

Dim nama_array(jumlah anggota) as tipe_data

Contoh:

*Dim daku(8) as byte ' variabel daku dengan tipe data byte
' dengan 8 anggota*

Untuk mengakses array dengan cara :

daku(1) = 25 ' anggota pertama variabel daku isinya 25

PORTC=daku(1) ' PORTC = nilai anggota pertama variabel daku

2.2.7. Operator

Operator digunakan dalam pengolahan data *pemrograman* dan biasanya membutuhkan dua variable atau dua parameter, sedangkan operator dituliskan di antara kedua parameter tersebut. Operator-operator BASCOM AVR ditunjukkan pada Tabel 2.5, Tabel 2.6 dan Tabel 2.7.

Tabel 2.5. Operator Aritmetik

Operator	Keterangan	Contoh
+	Operasi penjumlahan	A + B
-	Operasi pengurangan	A - B
*	Operasi perkalian	A * B
/	Operasi pembagian	A / B
%	Operasi sisa pembagian	A % B

Tabel 2.6. Operator Relasional

Operator	Keterangan	Contoh
=	Sama dengan	A = B
<>	Tidak sama dengan	A <> B
>	Lebih besar dari	A > B
<	Lebih kecil dari	A < B
>=	Lebih besar atau sama dengan	A >= B
<=	Lebih kecil atau sama dengan	A <= B

Tabel 2.7. Operator Logika

Operator	Keterangan	Contoh
AND	Operasi AND	&B110 And &B101 = &B100
OR	Operasi OR	&B11001 Or &B10111 = &B11111
NOT	Operasi NOT	NOT &HFF = &HO
XOR	Operasi XOR	&B1001 Xor &B0111 = &B1110

2.2.8. Operasi Bersyarat

A. IF – THEN

Sebuah atau serangkaian instruksi akan dikerjakan jika memenuhi syarat-syarat atau kondisi tertentu. Cara penulisannya sebagai berikut :

If <kondisi> Then <perintah> ‘ 1 baris perintah

If <kondisi> Then ‘ lebih dari 1 perintah

<perintah 1>

<perintah 2>

.....

End If

B. IF - THEN – ELSE

Versi lengkap dari sebuah atau serangkaian instruksi akan dikerjakan jika memenuhi syarat-syarat atau kondisi tertentu, jika tidak dipenuhi maka instruksi atau *serangkaian* instruksi lainnya yang akan dikerjakan. Cara penulisannya sebagai berikut:

If <kondisi> Then

<perintah 1>

...

Else

<perintah 2>

...

End If

C. IF - THEN - ELSEIF

Sama *seperti* IF-THEN-ELSE, hanya jika kondisi tidak dipenuhi masih dilakukan pengujian apakah suatu kondisi memenuhi syarat lainnya. Cara atau sintaks (*syntax*) penulisannya sebagai berikut:

```
If <kondisi 1> Then  
  <perintah 1>  
  ...  
Elseif <kondisi 2> Then  
  <perintah 2>  
  ...  
End If
```

D. SELECT - CASE

Baik digunakan untuk menangani pengujian kondisi yang jumlahnya cukup banyak. Cara penulisannya :

```
Select case <variabel>  
  Case 1: <perintah 1>  
  Case 2: <perintah 2>  
  ...  
End Select
```

2.2.9. Pengulangan Operasi

A. FOR - NEXT

Perintah ini digunakan untuk melaksanakan perintah *secara* berulang sesuai dengan jumlah yang ditentukan. Sintaks penulisannya :

```
For <var> = <nil_awal> To <nil_akhir> <step angka>  
  <perintah>  
Next [<var>]
```

B. DO - LOOP

Pernyataan ini untuk melakukan pengulangan terus menerus tanpa henti (pengulangan tak berhingga) selama mikrokontroler-nya masih mendapatkan detak dan/atau catu daya. Cara penulisannya :

Do
<pernyataan>

...

Loop

Jika pengulangan dibatasi oleh suatu kondisi maka caranya ditunjukkan berikut ini, artinya pengulangan terus dilakukan sehingga suatu kondisi terpenuhi atau melakukan pengulangan selama kondisinya salah:

Do
<pernyataan>

...

Loop Until <kondisi>

C. WHILE - WEND

Berbeda dengan DO-LOOP, instruksi ini digunakan untuk melakukan pengulangan selama kondisinya benar, cara penulisannya:

While <kondisi>
<perintah>

...

Wend

2.2.10. Lompatan Proses

A. GOSUB <nama_subrutin>

Perintah ini akan melakukan lompatan sebuah subrutin, *kemudian* kembali lagi setelah subrutin perintah tersebut selesai dikerjakan. Rutin yang dibuat harus diakhiri dengan instruksi RETURN. Contoh:

Print "We will start execution here"

Gosub Routine

Print "Back from Routine"

End

Routine:

Print "This will be executed"

Return

B. GOTO <label>

Perintah ini untuk melakukan lompatan ke label kemudian *melakukan* serangkaian instruksi tanpa harus kembali lagi, sehingga tidak perlu RETURN.

Contoh:

Dim A As Byte

Start: 'sebuah label diakhiri dengan :

A = A + 1 'naikkan variabel A

If A < 10 Then 'apakah lebih kecil 10?

Goto Start 'ya, lakukan lagi

End If 'akhir IF

Print "Ready" 'ok

C. EXIT

Untuk keluar secara langsung dari perulangan DO-LOOP, FOR-NEXT, WHILE-WEND. Cara penulisannya sebagai berikut :

EXIT FOR (keluar dari For-Next)

EXIT DO (keluar dari Do-Loop)

EXIT WHILE (keluar dari While-Wend)

EXIT SUB (keluar dari Sub-Endsub)

EXIT FUNCTION (keluar dari suatu *fungsi*)

Sumber :

“Mudah menguasai pemrograman Mikrokontroler ATMEL AVR menggunakan BASCOM AVR”. Agfianto Eko Putro. Versi 1.5 – 2010. Kelompok Riset DSP dan Embedded Intelligent System – ELINS Universitas Gadjah Mada – Yogyakarta.

2.3. Bahasa Pemrograman Basic 4 Android

Basic4android adalah development tool sederhana yang *powerful* untuk membangun aplikasi Android. Bahasa Basic4android mirip dengan bahasa Visual Basic dengan tambahan dukungan untuk objek. Aplikasi Android (APK) yang dicompile oleh Basic4Android adalah aplikasi Android *native*/asli dan tidak ada extra runtime seperti di Visual Basic yang ketergantungan file *msvbvm60.dll*, yang pasti aplikasi yang dicompile oleh Basic4Android adalah NO DEPENDENCIES (tidak ketergantungan file oleh lain). IDE Basic4Android hanya fokus pada development Android.

Basic4android termasuk designer GUI untuk aplikasi Android yang powerful dengan dukungan Built - in untuk multiple screens dan orientations, serta tidak dibutuhkan lagi penulisan XML yang rumit. Disamping itu Basic4android memiliki kekayaan dalam satuan libraries (perpustakaan) yang membuatnya menjadi lebih mudah untuk mengembangkan macam -macam aplikasi Android yang advanced.

2.3.1. Tipe Data

Seperti bahasa pemrograman lainnya Basic 4 Android mengenal beberapa tipe data yang dapat dimanfaatkan oleh pengembang aplikasi mobile, untuk lebih detailnya dapat dilihat pada tabel berikut:

Tabel Tipe Data Basic 4 Android

B4A	Type	min value	max value
Boolean	boolean	False	True
Byte	integer 8 bits	-2^7 -128	$2^7 - 1$ 127
Short	integer 16 bits	-2^{15} - 32768	$2^{15} - 1$ 32767
Int	integer 32 bits	-2^{31} -2147483648	$2^{31} - 1$ 2147483647
Long	long integer 64 bits	-2^{63} -9223372036854775808	$2^{63} - 1$ 9223372036854775807
Float	floating point number 32 bits	-2^{-149} 1.4E-45	$(2 - 2^{-23}) * 2^{127}$ 3.4028235 E 38
Double	double precision number 64 bits	-2^{-1074} 2.2250738585072014 E - 308	$(2 - 2^{-52}) * 2^{1023}$ 1.7976931348623157 E 308
Char	character		
String	array of characters		

2.3.2. Variabel

Variabel atau peubah digunakan untuk menyimpan data sementara. *Variabel* diberi nama dan dideklarasikan terlebih dahulu sebelum digunakan. Aturan pemberian nama variabel sebagai berikut:

- a. Harus dimulai dengan huruf (bukan angka).
- b. Tidak ada nama variabel yang sama dalam sebuah program.
- c. Tanpa menggunakan spasi, pemisahan bisa dilakukan dengan garis bawah.
- d. Tidak menggunakan karakter-karakter khusus yang digunakan oleh basic 4 android sebagai *reserved words*.

Dalam basic 4 android dikenal beberapa cara untuk mendeklarasikan Variabel, antara lain:

2.3.2.1. Simple Variables

Variabel yang di deklarasikan dengan menggunakan kata *dim* yang kemudian diikuti dengan tipe variabel tersebut, contoh:

Dim Rate As Double

Dim i As Int

Dim i, j, k As Int

Dim i= 0, j= 2, k= 5 As Int

2.3.2.2. Array Variables

Array merupakan kumpulan dari data atau objek yang dapat dipilih melalui urutan indeks data atau objek tersebut. Penulisan deklarasi diawali dengan kata “DIM” yang kemudian diikuti dengan nama variable disertai dengan tanda kurung buka dan kurung tutup yang berisi jumlah total array data atau objek, kemudian diakhiri dengan tipe data atau objek tersebut, Contoh :

```
Dim LastName(50) As String
Dim Matrix(3, 3) As Double
Dim Data(3, 5, 10) As Int
```

2.3.2.3. Array Of Objects

Views atau objects juga dapat disusun secara Array, contoh :

```
Dim b1, b2, b3, b4, b5, b6, b7 As Button
Buttons= Array As Button(b1, b2, b3, b4, b5, b6, b7)
Dim Buttons() As Button
```

2.3.3. Scope

Scope menandakan sejauh mana tipe variabel yang dideklarasikan berlaku, dalam basic 4 android ada beberapa scope variabel, antara lain:

2.3.3.1. Process Variables

Merupakan variabel yang dideklarasikan dalam sub “Process_Globals”, variabel ini berlaku sepanjang proses aplikasi hidup, contoh :

```
Sub Process_Globals
    Dim MyVar As String
End Sub
```

2.3.3.2. Activity Variables

Merupakan variabel yang dideklarasikan dalam sub “Globals”, variabel ini berlaku sebatas modul saja, contoh :

```
Sub Process_Globals
    Dim MyVar As String
End Sub
```

2.3.3.3. Local Variables

Merupakan variabel yang dideklarisikan dalam subrutin khusus, variabel ini berlaku sebatas subrutin tersebut saja, dan akan dihancurkan pada saat eksekusi rutin tersebut selesai, contoh :

```
Sub Hitung
    Dim a As int
    Dim b As int
    Dim c As int
    C = a + b
End Sub
```

2.3.4. Module

Aplikasi yang dibuat dengan basic 4 android akan terbentuk berdasarkan beberapa modul, diantaranya:

2.3.4.1. Activity Modules

Setiap activity sebuah aplikasi pasti mempunyai minimal satu buah modul atau lembar kode khusus untuk sebuah interface tunggal, module tersebut berisi subrutin dasar sebuah interface antara lain :

```
Sub Process_Globals

End Sub

Sub Globals

End Sub

Sub Activity_Create(FirstTime As Boolean)
    'Activity.LoadLayout("Layout1")
End Sub
```

```
Sub Activity_Resume
```

```
End Sub
```

```
Sub Activity_Pause (UserClosed As Boolean)
```

```
End Sub
```

2.3.4.2. Class Modules

Merupakan sebuah modul yang digunakan untuk membentuk serta memberikan karakteristik sebuah objek. Contoh:

```
Sub Class_Globals
```

```
    Private FirstName, LastName As String
```

```
    Private BirthDate As Long
```

```
End Sub
```

```
Sub Initialize (aFirstName As String, aLastName As String,  
aBirthDate As Long)
```

```
    FirstName= aFirstName
```

```
    LastName= aLastName
```

```
    BirthDate= aBirthDate
```

```
End Sub
```

```
Public Sub GetName As String
```

```
    Return FirstName& " "& LastName
```

```
End Sub
```

```
Public Sub GetCurrentAge As Int
```

```
    ReturnGetAgeAt(DateTime.Now)
```

```
End Sub
```

```
Public Sub GetAgeAt(Date As Long) As Int
```

```
    Dimdiff As Long
```

```
    diff = Date - BirthDate
```

Return Floor(diff / DateTime.TicksPerDay / 365)

End Sub

2.3.4.3. Code Modules

Merupakan modul yang khusus berfungsi menyimpan kode saja, tidak ada penulisan aktifitas yang diijinkan dalam modul tersebut. Contoh:

- **Modul DBUtils**
- **Modul HttpUtils**
- **Modul StateManager**

2.3.4.4. Service Modules

Service Modules memainkan peranan penting dalam sebuah aplikasi dan life cycle proses sebuah aplikasi. Dalam sebuah service paling tidak mempunyai subrutin sebagai berikut :

Sub Process_Globals

End Sub

Sub Service_Create

End Sub

Sub Service_Start (StartingIntent As Intent)

End Sub

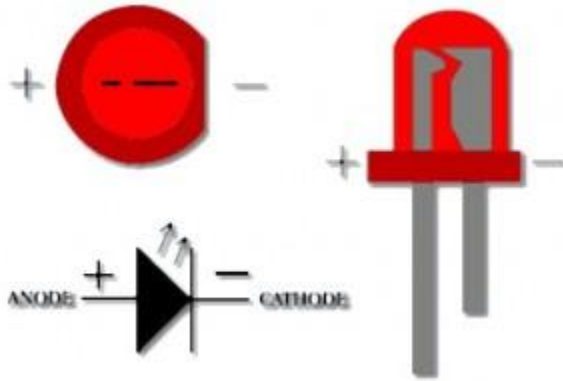
Sub Service_Destroy

End Sub

2.4. LED

LED (*Light Emitting Dioda*) adalah dioda yang dapat memancarkan cahaya pada saat mendapat arus bias maju (*forward bias*). LED dapat memancarkan cahaya karena menggunakan *dopping* galium, arsenic dan phosporus. Jenis *doping* yang berbeda diata dapat menghasilkan cahaya dengan warna yang berbeda. LED merupakan salah satu jenis dioda, sehingga hanya akan mengalirkan arus listrik satu arah saja. LED akan

memancarkan cahaya apabila diberikan tegangan listrik dengan konfigurasi *forward bias*. Berbeda dengan dioda pada umumnya, kemampuan mengalirkan arus pada LED cukup rendah yaitu maksimal 20 mA. Apabila LED dialiri arus lebih besar dari 20 mA maka LED akan rusak, sehingga pada rangkaian LED dipasang sebuah resistor sebagai pembatas arus.



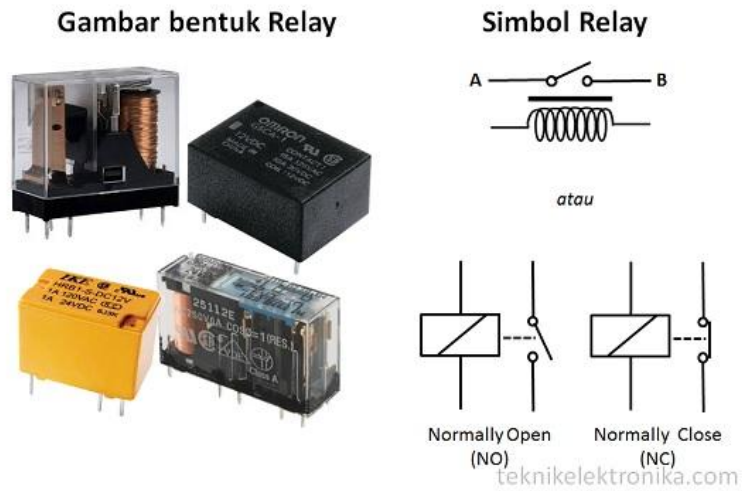
Gambar: Simbol dan bentuk fisik LED

Dari gambar diatas dapat kita ketahui bahwa LED memiliki kaki 2 buah seperti dengan dioda yaitu kaki anoda dan kaki katoda. Pada gambar diatas kaki anoda memiliki ciri fisik lebih panjang dari kaki katoda pada saat masih baru, kemudian kaki katoda pada LED ditandai dengan bagian *body* LED yang di papas rata. Kaki anoda dan kaki katoda pada LED disimbolkan seperti pada gambar diatas. Pemasangan LED agar dapat menyala adalah dengan memberikan tegangan bias maju yaitu dengan memberikan tegangan positif ke kaki anoda dan tegangan negatif ke kaki katoda (sumber: elektronika-dasar.web.id, 2012).

2.5. Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni *Elektromagnet (Coil)* dan Mekanikal (seperangkat Kontak Saklar/*Switch*). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan Relay yang menggunakan

Elektromagnet 5V dan 50 mA mampu menggerakkan Armature Relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

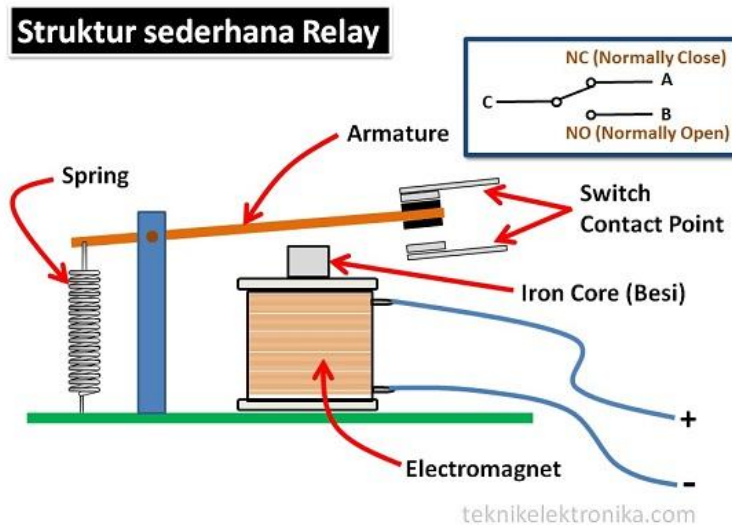


Gambar: Bentuk Fisik Dan Simbol Relay

2.5.1. Prinsip Kerja Relay

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

- **Electromagnet (Coil)**
- **Armature**
- **Switch Contact Point (Saklar)**
- **Spring**



Gambar : Struktur Relay

Kontak Poin (*Contact Point*) Relay terdiri dari 2 jenis yaitu :

- Normally Close (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
- Normally Open (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Berdasarkan gambar diatas, sebuah Besi (*Iron Core*) yang dililit oleh sebuah kumparan *Coil* yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan *Coil* diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik *Armature* untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana *Armature* tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, *Armature* akan kembali lagi ke posisi Awal (NC). *Coil* yang digunakan oleh Relay untuk menarik *Contact Poin* ke Posisi *Close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

2.5.2. *Pole* dan *Throw* pada *Relay*

Karena *Relay* merupakan salah satu jenis dari Saklar, maka istilah *Pole* dan *Throw* yang dipakai dalam Saklar juga berlaku pada *Relay*. Berikut ini adalah penjelasan singkat mengenai Istilah Pole and Throw :

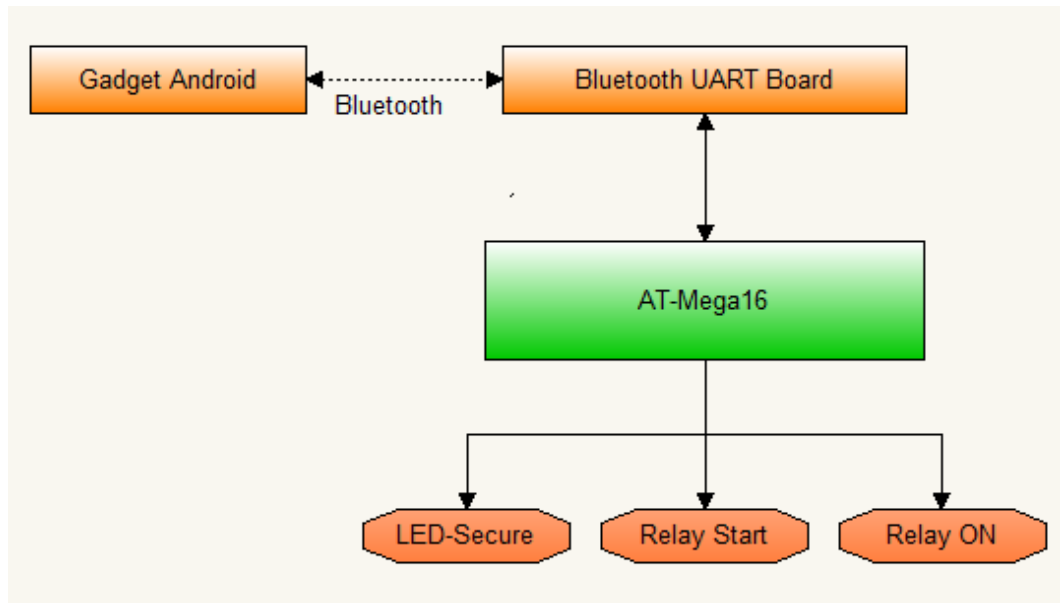
BAB III

ANALISA PERANCANGAN

Bab ini akan menguraikan perencanaan perangkat keras (*hardware*) dan perangkat lunak (*software*). Perencanaan *hardware* bertujuan merealisasikan ide *aplikatif* menjadi rangkaian elektronika yang didukung sistem mikrokontroler ATMEGA16 yang akan dilakukan dalam beberapa tahap sistematis. Perencanaan *software* akan mendukung kerja perangkat keras dan bisa memfungsikan perangkat keras sesuai dengan yang direncanakan.

3.1. Diagram fungsional

Gambaran secara umum tentang penelitian ini dapat dilihat dari Gambar Diagram Fungsional dibawah ini:



Gambar Diagram Fungsional Sistem

3.2. Sistem Kerja Alat

Prinsip kerja dari alat ini berawal dari karakter “r” yang dikirim oleh gadget android ke bluetooth UART Board melalui gelombang radio bluetooth sebagai request password sistem, kemudian sistem akan mengirimkan sebuah password khusus ke gadget android.

Pada saat password yang dimasukan betul maka user akan dibawa ke menu utama yang terdiri dari 3 button:

- Button ON
- Button Start
- Button OFF

Pada saat **Button ON** ditekan maka gadget akan mengirim karakter “ON” hal ini diartika oleh mikrokontroler sebagai perintah untuk menyalakan Relay ON yang kemudian diiringi pengiriman string “Motor ON !!!”, string tersebut ditampilkan ke gadget android sebagai pesan.

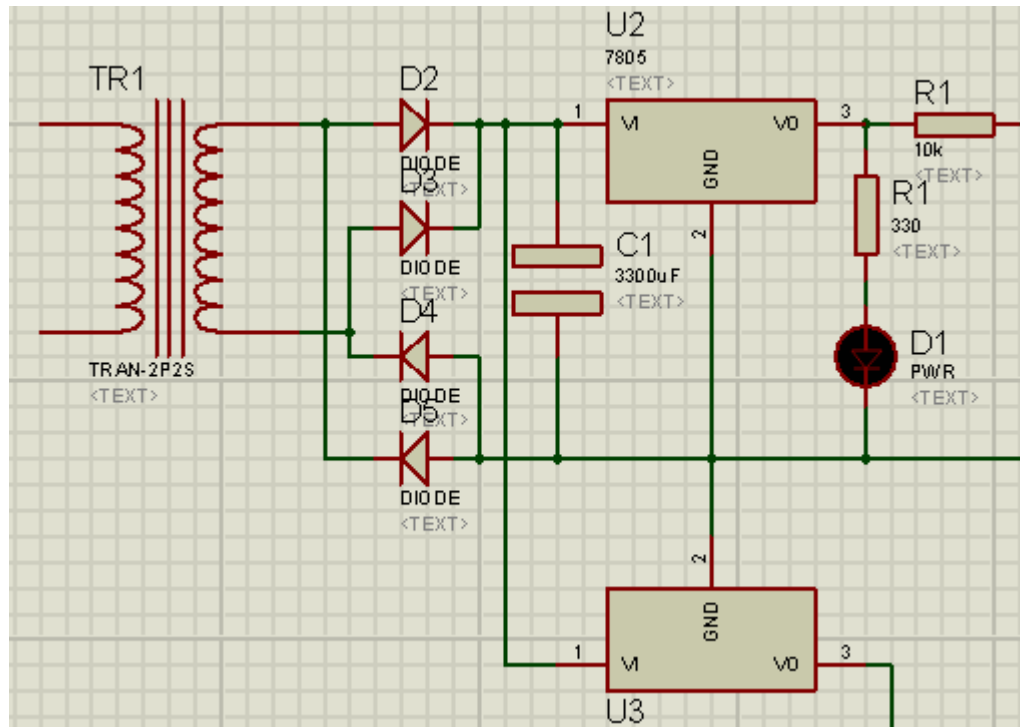
Pada saat **Button Start** ditekan maka gadget akan mengirim karakter “START” hal ini diartika oleh mikrokontroler sebagai perintah untuk menyalakan Relay ON yang kemudian diiringi pengiriman string “Motor START !!!”, string tersebut ditampilkan ke gadget android sebagai pesan.

Pada saat **Button OFF** ditekan maka gadget akan mengirim karakter “OFF” hal ini diartika oleh mikrokontroler sebagai perintah untuk menyalakan Relay ON yang kemudian diiringi pengiriman string “Motor Lock !!!”, string tersebut ditampilkan ke gadget android sebagai pesan.

3.3. Perancangan Perangkat Keras (*hardware*)

3.3.1. *Power Supply*

Power Supply / catu daya digunakan untuk memasok kebutuhan daya yang digunakan oleh semua perangkat yang terhubung dalam sistem.



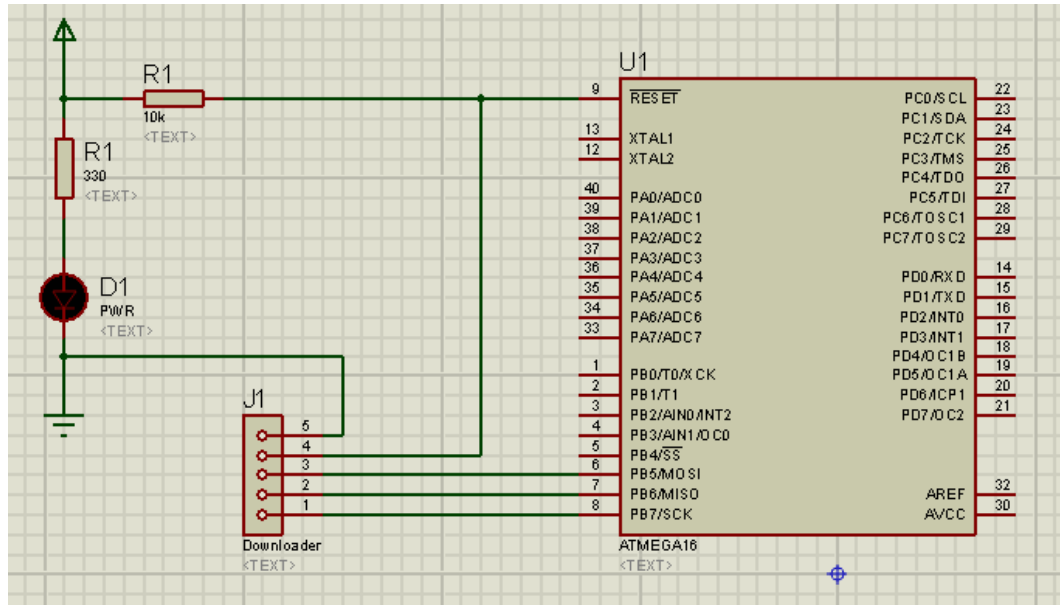
Gambar: Rangkaian *power supply*

Pada rangkaian tersebut, input tegangan dari PLN sebesar 220 VAC diturunkan menjadi 12 VAC melalui transformator penurun tegangan (transformator stepdown). Selanjutnya tegangan output dari transformator masing-masing disearahkan oleh rangkaian dioda *bridge rectifier* (penyearah jembatan). Hasil dari penyearahan tersebut masih memiliki tegangan ripple. Kapasitor 3300uf/16V (untuk keluaran 12v) digunakan sebagai filter untuk mengurangi tegangan ripple dari keluaran penyearah jembatan. Keluaran dari dioda ini masuk ke IC regulator 7812 yang berfungsi untuk menstabilkan tegangan searah 12V DC dan ke IC regulator 7805 yang berfungsi untuk menstabilkan tegangan searah 5V DC.

3.3.2. Sistem *Mikrokontroler*

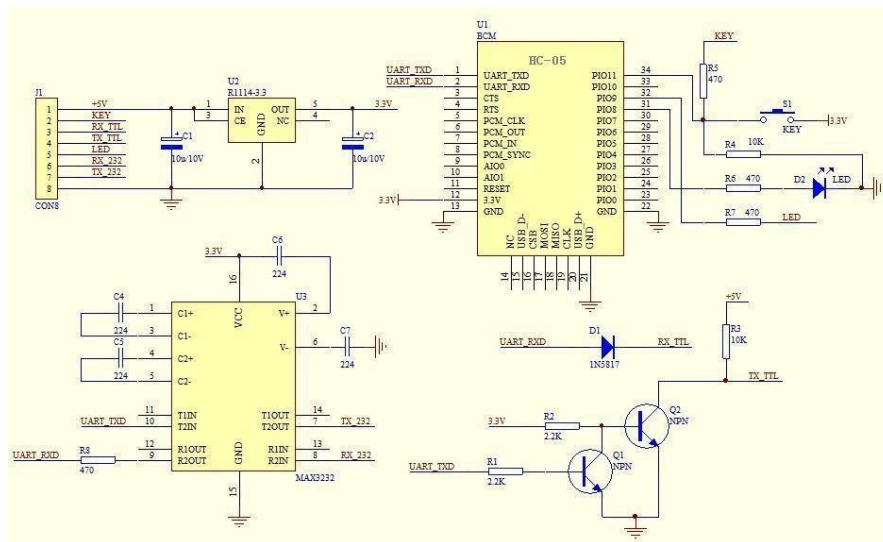
Rangkaian sistem minimum mikrokontroler merupakan rangkaian minimal agar mikrokontroler dapat bekerja. Rangkaian yang digunakan dalam penelitian ini merupakan rangkaian standar yang ada pada mikrokontroler AT-MEGA16, yaitu terdiri dari rangkaian mikrokontroler dengan clock internal, rangkaian reset (resistor),

rangkaian *downloader* line dan indikatos *power*. Rangkaian sistem minimum mikrokontroler ditunjukkan pada gambar berikut:



Gambar Sekematik Rangkaian Minimum Sistem

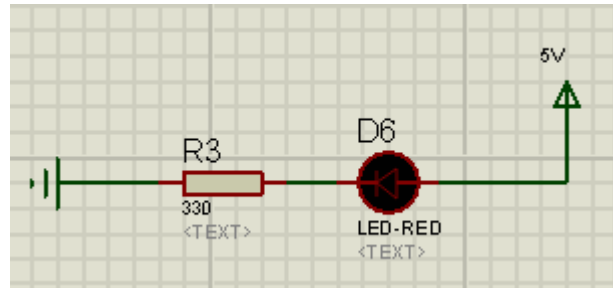
3.3.3. Bluetooth UART Board



Gambar Skema Rangkaian Bluetooth UART Board

Bluetooth UART Board merupakan sebuah solusi komunikasi pengiriman data secara *wireless* melalui gelombang radio bluetooth, dengan alat ini memungkinkan terjadinya komunikasi dua arah (*Transmit* dan *Receive*) antara mikrokontroler AT-Mega16 dengan *Gadged* android dengan rentang baudrate mulai 1200bps sampai 1382400bps.

3.3.4. Rangkaian LED

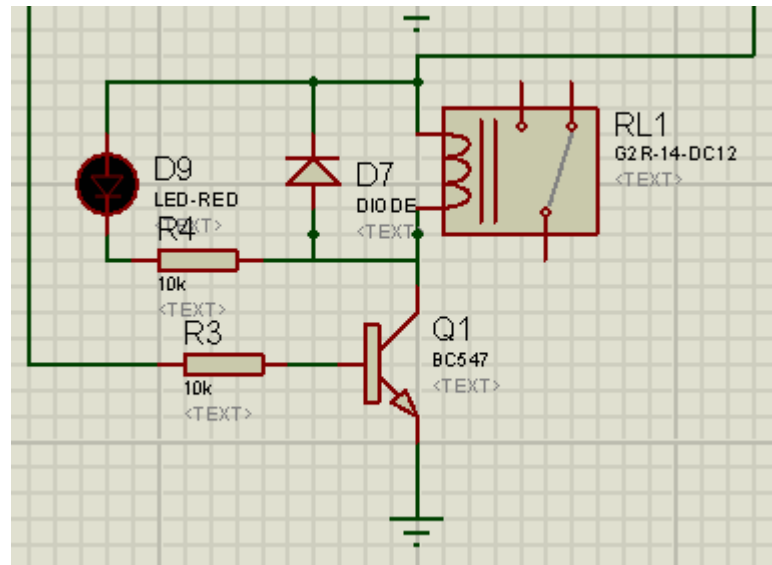


Gambar Rangkaian LED

Lampu LED dihubungkan dengan sebuah resistor bernilai 330 Ohm hal ini bertujuan agar arus yang mengalir pada rangkaian maksimal

3.3.5. Rangkaian Relay

Sebagai driver dari rangkaian *relay* menggunakan transistor BC547 (*general purpose* NPN), pada dasarnya *relay* di hubungkan ke tegangan Vcc 5 Volt, karena adanya transistor, maka *relay* mendapatkan arus atau tidaknya tergantung dari kondisi transistor saat itu, jika transistor ON (karena adanya arus low pada basis, dengan pemberian logika '1'), maka *relay* mendapat tegangan Vcc, namun sebaliknya jika transistor OFF (karena adanya arus high pada basis, dengan pemberian logika '0'), maka *relay* juga OFF. Skema rangkaian *relay* di pellihatkan pada Gambar berikut:



Gambar : Skema Rangkaian *Relay*

Tahanan 10 KOhm pada kaki basis *transistor* berfungsi sebagai pembatas arus yang masuk melalui basis.

Dioda dipasang dengan tujuan untuk menghalangi tegangan *feed back* dari *Relay*.

3.4. Perancangan Perangkat Lunak (*Software*)

Perencanaan perangkat lunak atau perencanaan *software* merupakan perancangan spesifikasi sistem yang berhubungan dengan program kinerja atau kerja mikrokontroler dalam hal pembacaan data, proses pengolahan data dan proses pengiriman data.

3.4.1. Perancangan Program LED dan Relay

Pengujian ditujukan untuk memastikan bahwa rangkaian LED dan rangkaian Relay dapat berfungsi dengan normal. Pengujian dilakukan dengan menyalakan masing-masing LED dan Relay secara bergantian.

Berikut adalah source yang digunakan selama pengujian :

```
$regfile = "m16def.dat"
$crystal = 11059200
$baud = 9600
```

Relay_on Alias Porta.0

Relay_start Alias Porta.1

Lampu_lock Alias Porta.2

Config Relay_on = Output

Config Relay_start = Output

Config Lampu_lock = Output

Lampu_lock = 1

Relay_start = 0

Relay_on = 0

Do

Lampu_lock = 0

Wait 1

Relay_start = 1

Wait 1

Relay_on = 1

Wait 1

Lampu_lock = 1

Relay_start = 0

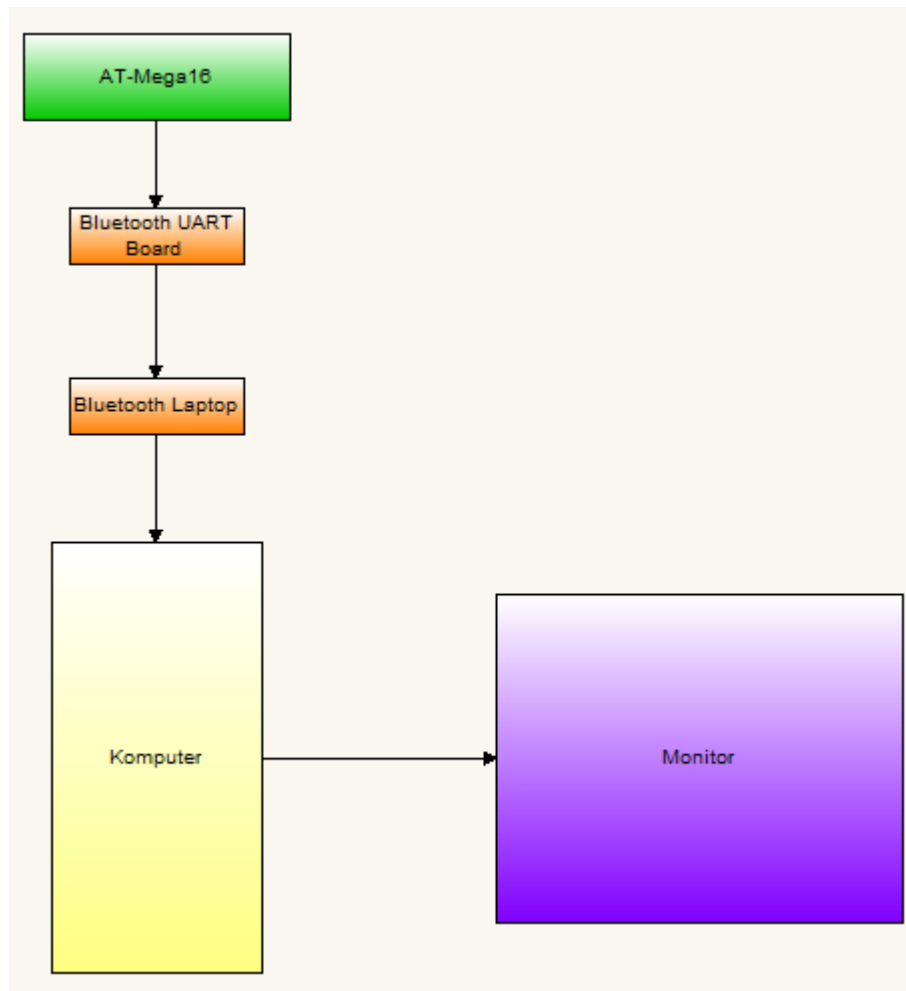
Relay_on = 0

Loop

End

3.4.3. Perancangan Program *Bluetooth UART Board*

Pengujian ditujukan untuk memastikan bahwa *Bluetooth UART Board* dapat berkomunikasi melalui UART / Serial Port. Dalam pengujian kali ini digunakan pengujian melalui komputer agar lebih mudah dalam melihat hasil dari komunikasi tersebut.



Gambar sekema rangkaian pengujian *Bluetooth UART Board*

Berikut adalah source yang digunakan selama pengujian :

```
$regfile = "m16adef.dat"
```

```
$crystal = 11059200
```

```
$baud = 9600
```

```
Dim X As String * 50
```

```
Do
```

```
    Print "Masukan Nama Anda "
```

```
    Input X Noecho
```

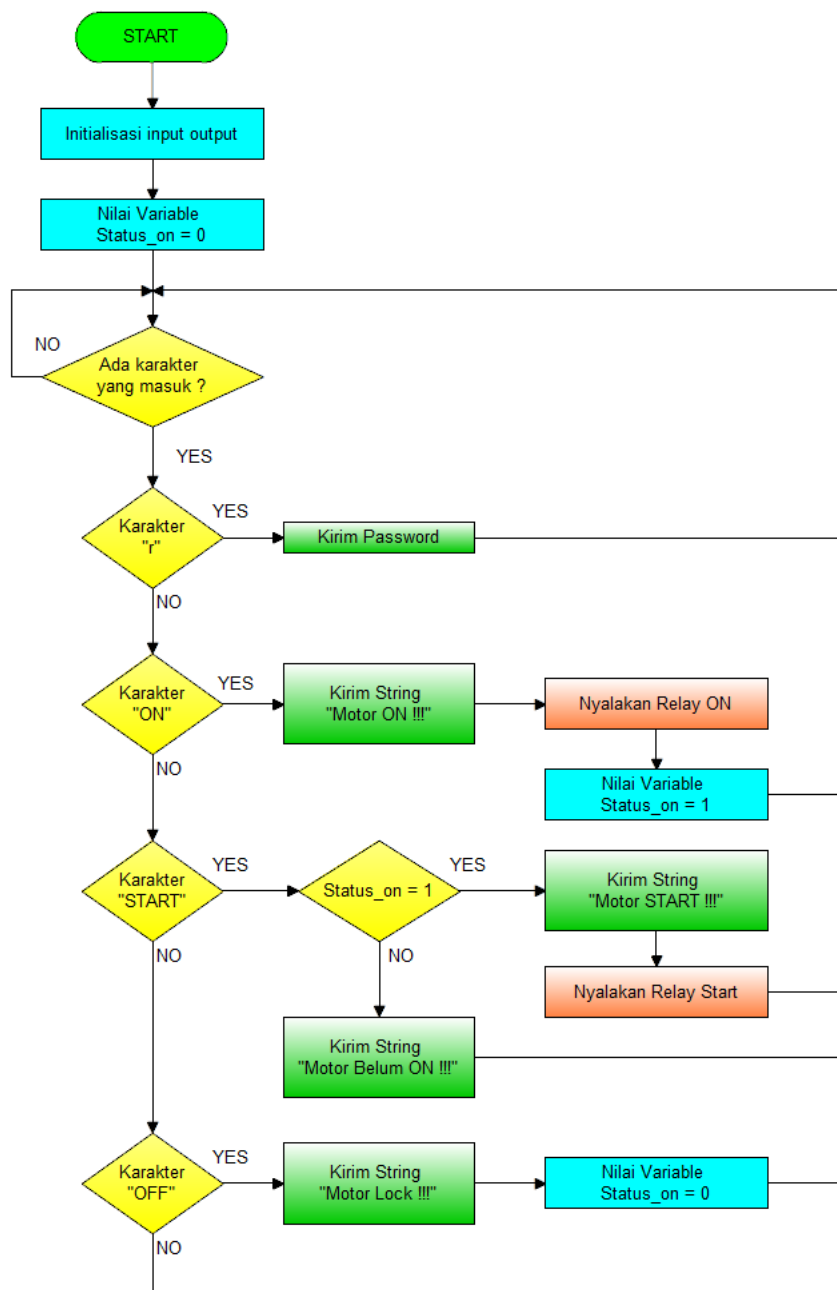
```
    Print "halo" ; X
```

```
Loop
```

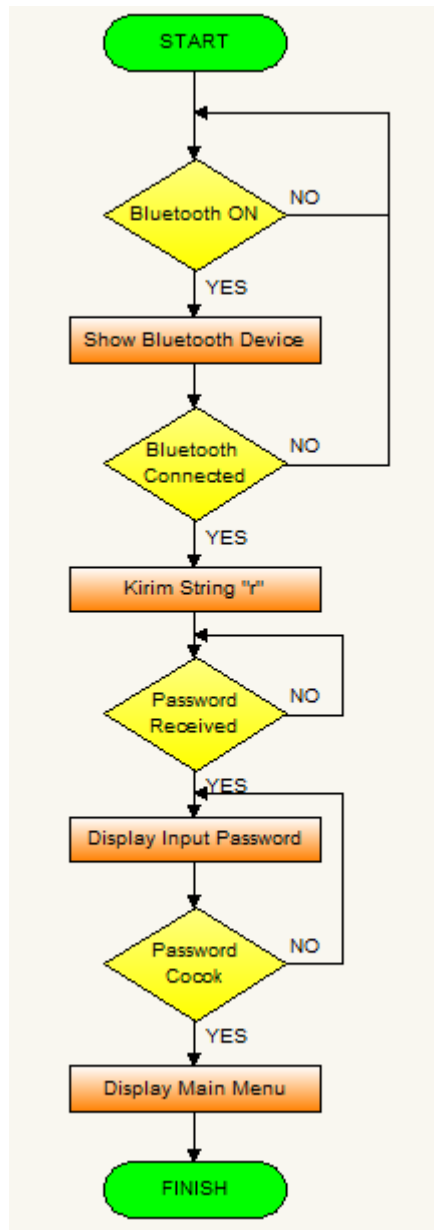
```
End
```

3.4.5. Flow Chart *Software* AT-Mega16 dan Android

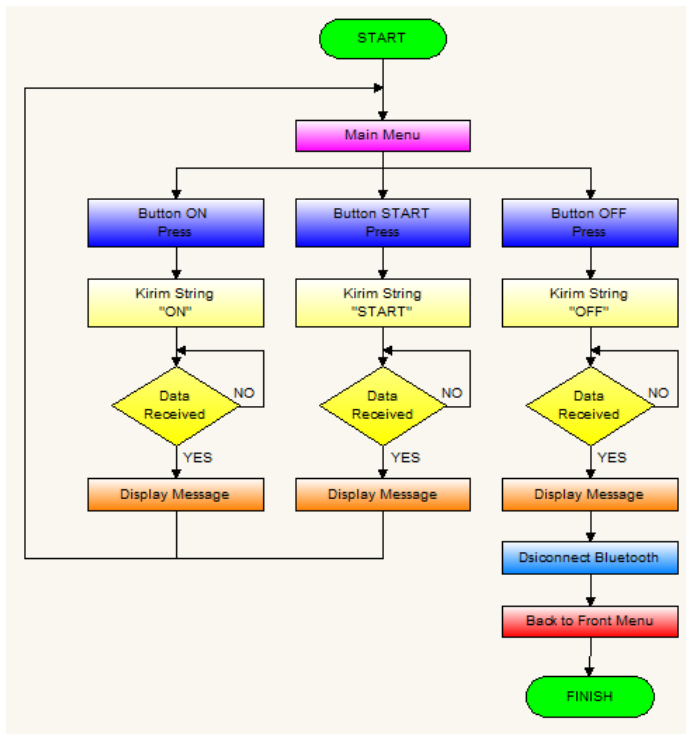
Sebelum *software* dibuat menjadi program secara utuh maka terlebih dahulu dibuat sebuah *flowchart* yang merupakan alur pikir dari program inti yang akan dibuat. Selanjutnya hasil dari *flowchart* ini akan dibuat *software* secara utuh. Dalam sistem yang dibuat, *software* dibagi menjadi dua yaitu *software* untuk AT-Mega16 dan *Software* untuk Gadget Android



Gambar *Flow Chart* Program AT-Mega16



Gambar *Flow Chart* Program Android Front Menu



Gambar *Flow Chart* Program Android Main Menu

BAB IV

ANALISIS DAN PEMBAHASAN

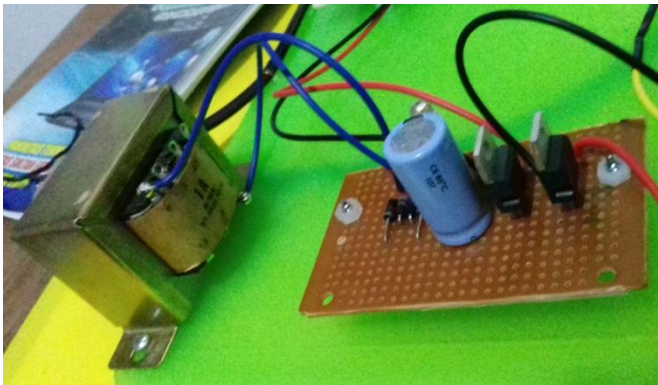
Bab ini akan membahas mengenai implementasi secara keseluruhan terhadap sistem serta pengujian alat, hal ini perlu dilakukan baik terhadap perangkat keras maupun perangkat lunak. Pengujian ini dilakukan untuk mengetahui hasil dari program yang telah dibuat apakah sudah sesuai dengan yang diharapkan. Dari pengujian ini diharapkan dapat diketahui kesalahankesalahan atau kekurangan-kekurangan yang ada, sehingga dapat dilakukan perbaikan atau penyempurnaan pada sistem.

FOTO ALAT KOMPLIT

4.1. Pengujian Perangkat Keras (*Hardware*)

Pengujian *hardware* dilakukan dengan tujuan untuk mengetahui karakteristik (besaran Voltase dan Ampere) masing-masing *hardware* yang digunakan dalam sistem.

4.1.1. Pengujian Rangkaian Power Supply



Gambar Rangkaian *Power Supply*

Pengujian rangkaian Catu Daya ini dilakukan dengan cara mengukur tegangan keluaran dari Catu Daya tersebut dengan menggunakan AVO Meter. Dari rangkaian pengujian diperoleh keluaran atau output dari Catu Daya sebagai berikut:

Tabel Hasil Pengujian Rangkaian Catu Daya

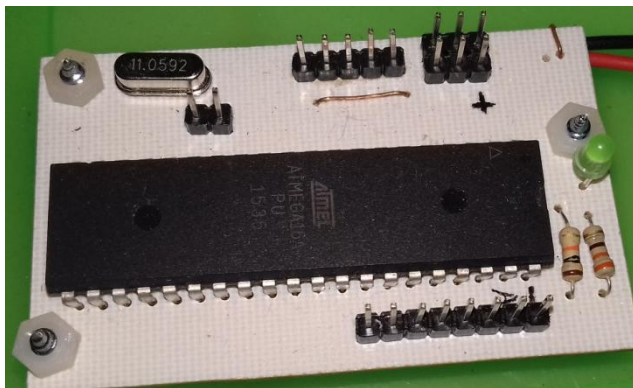
Tegangan Output yang diinginkan (Vdc)	Tegangan Output Terukur (Vdc)
+5	+4,9
+12	+12



Gambar pengukuran tegangan power supply 5V dan 12V

Dikarenakan sistem yang dibangun menggunakan basic tegangan 12V dan tegangan antara 4.5-5.5 V (terutama tegangan pada mikrokontroler) maka tegangan tersebut masih mencukupi.

4.1.2. Pengujian Rangkaian Sistem Minimum Mikrokontroler



Gambar Fisik Rangkaian Minimum Sistem

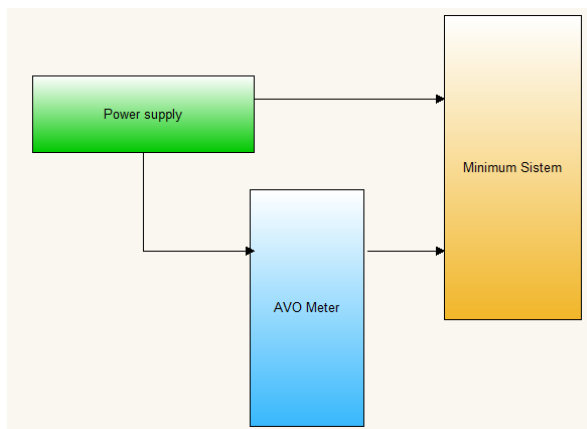
Pengujian rangkaian Sistem Minimum Mikrokontroler ini dilakukan dengan cara mengukur besaran arus yang mengalir dalam rangkaian tersebut dengan menggunakan AVO Meter. Dari rangkaian pengujian diperoleh besaran arus yang mengalir dalam minimum sistem adalah 26.0 mA atau lebih besar 6mA dari *datasheet* AT-Mega16, besaran tersebut sangat wajar dikarenakan dalam minimum sistem terdapat rangkaian LED sebagai indikator power.

Tabel Karakteristik Pemakaian Daya Mikrokontroler AT-Mega16

DC Characteristics (Continued)

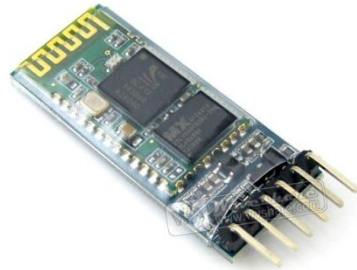
$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units	
I_{CC}	Power Supply Current	Active 4 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)			5	mA	
		Active 8 MHz, $V_{CC} = 5\text{V}$ (ATmega16)			20	mA	
		Idle 4 MHz, $V_{CC} = 3\text{V}$ (ATmega16L)			2	mA	
		Idle 8 MHz, $V_{CC} = 5\text{V}$ (ATmega16)			12	mA	
	Power-down Mode ⁽⁵⁾	WDT enabled, $V_{CC} = 3\text{V}$			< 25	40	μA
		WDT disabled, $V_{CC} = 3\text{V}$			< 10	15	μA



Gambar 4.2 Rangkaian Pengujian Sistem Minimum Mikrokontroler

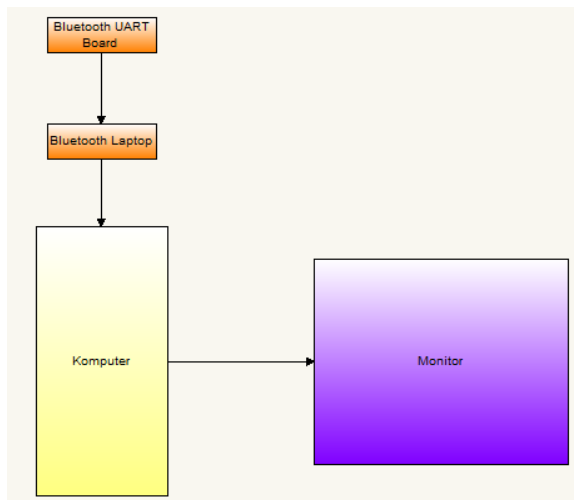
4.1.3. Pengujian Bluetooth UART Board



Gambar Fisik *Bluetooth UART Board*

Pada pengujian rangkaian *Bluetooth UART Board* ini dilakukan dengan melihat apakah modul dapat mengirim dan menerima data dari komputer.

Rangkaian pengujiannya adalah sebagai berikut:

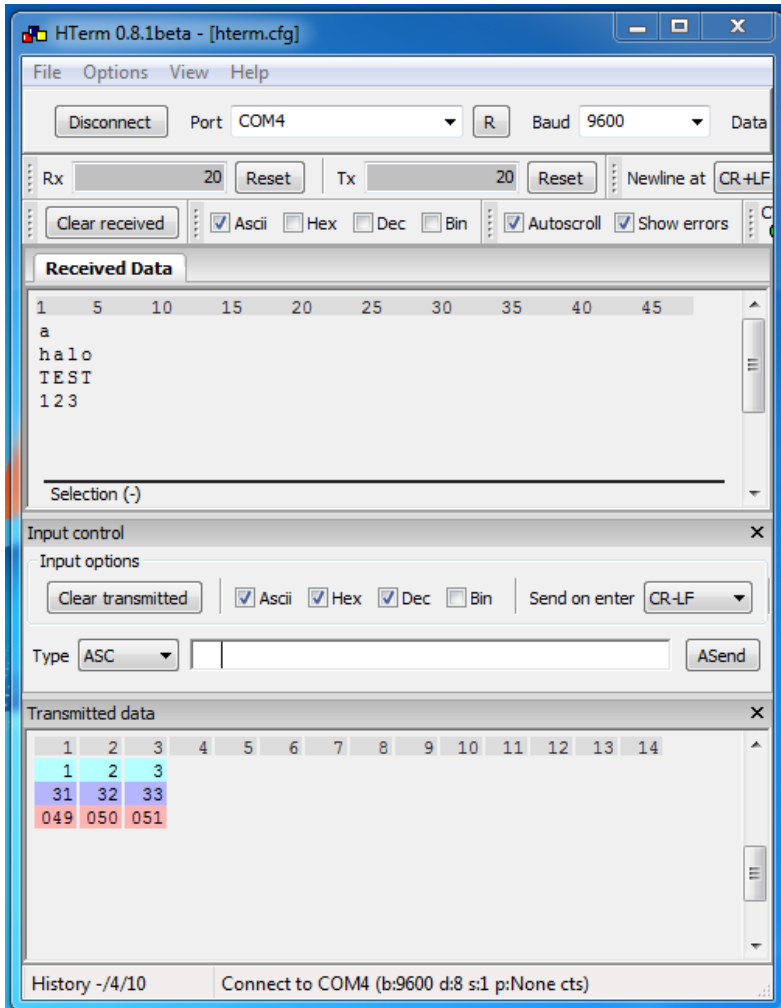


Gambar 4.2 Skema Pengujian *Bluetooth UART Board*

Pengujian dilakukan dengan cara mengirimkan data secara acak melalui bluetooth komputer dengan baudrate serial sebesar 9600bps.

Tabel Hasil Pengujian Bluetooth UART Board

Data Kirim (String)	Data Terima (String)
a	a
halo	halo
TEST	TEST
123	123



Gambar Hasil Pengujian Bluetooth UART Board

4.1.4. Pengujian Rangkaian LED



Gambar Rangkaian LED

Dalam sistem yang dibuat lampu LED terhubung dengan resistor yang kemudian diteruskan ke kaki mikrokontroler. Secara umum karakteristik sebuah LED mempunyai arus maju maksimal sebesar 20mA. Mengingat tegangan yang berasal dari power supply sebesar 5V dan jika dihungkan langsung dengan tegangan utama sistem akan mengalami kerusakan, maka dibutuhkan sebuah resistor untuk menurunkan arus yang mengalir.

Dalam rangkain sistem digunakan sebuah resistor bernilai 330 Ohm, maka sesuai dengan hukum Ohm diperoleh arus sebesar:

$$I = V / R$$

$$I = 5 / 330$$

$$I = 0,015A$$

$I=15\text{mA}$

4.1.5. Pengujian Rangkaian Relay

Untuk rangkaian *Relay* tidak terhubung secara langsung dengan mikrokontroler melainkan terhubung dengan sebuah transistor BC547 sebagai penguat arus, berdasarkan data dari *datasheet* BC547, transistor tersebut mempunyai karakteristik sebagaiberikut:

Tabel karakteristik BC547

Electrical Characteristics $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
I_{CBO}	Collector Cut-off Current	$V_{CB}=30\text{V}, I_E=0$			15	nA
h_{FE}	DC Current Gain	$V_{CE}=5\text{V}, I_C=2\text{mA}$	110		800	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C=10\text{mA}, I_B=0.5\text{mA}$ $I_C=100\text{mA}, I_B=5\text{mA}$		90 200	250 600	mV mV
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	$I_C=10\text{mA}, I_B=0.5\text{mA}$ $I_C=100\text{mA}, I_B=5\text{mA}$		700 900		mV mV
$V_{BE(on)}$	Base-Emitter On Voltage	$V_{CE}=5\text{V}, I_C=2\text{mA}$ $V_{CE}=5\text{V}, I_C=10\text{mA}$	580	660	700 720	mV mV
f_T	Current Gain Bandwidth Product	$V_{CE}=5\text{V}, I_C=10\text{mA}, f=100\text{MHz}$		300		MHz
C_{ob}	Output Capacitance	$V_{CB}=10\text{V}, I_E=0, f=1\text{MHz}$		3.5	6	pF
C_{ib}	Input Capacitance	$V_{EB}=0.5\text{V}, I_C=0, f=1\text{MHz}$		9		pF
NF	Noise Figure	$V_{CE}=5\text{V}, I_C=200\mu\text{A}$ $f=1\text{KHz}, R_G=2\text{K}\Omega$ $V_{CE}=5\text{V}, I_C=200\mu\text{A}$ $R_G=2\text{K}\Omega, f=30\sim 15000\text{MHz}$		2 1.2 1.4 1.4	10 4 4 3	dB dB dB dB

Didalam driver *Relay* terdapat transistor sebagai saklar untuk mengaktifkan *Relay*. Tegangan supply yang digunakan input transistor adalah 12V DC yang difungsikan untuk mensaklar sebuah *Relay* 12V DC 10 mA. Transistor dipilih bervariasi dengan variasi h_{fe} dari 110 – 800 yaitu BC547.

Karena transistor mempunyai h_{fe} antara 110 - 800 maka di pilih terlebih dahulu h_{fe} minimum nya (100). Arus kolektor adalah 50mA, maka Arus Basis yang dibutuhkan adalah:

$$h_{fe} = I_c / I_b$$

$$I_b = I_c / h_{fe}(\text{min})$$

$$I_b = 0,05/100$$

$$I_b = 0,0005\text{A}$$

$$i_b = 0,5\text{mA}$$

Nilai V_{in} adalah 12V DC, nilai V_{be} adalah 0,7V DC (konstanta) berarti tegangan yang melewati R_b adalah:

$$V_{in} - V_{be} = 12 - 0,7$$

$$V_{in} - V_{be} = 11,3 \text{ V DC}$$

Sehingga Nilai R_b dapat di hitung:

$$R_b = 11.3 / 0.0005$$

$$R_b = 22.600$$

$$R_b = 22,6 \text{ K}\Omega$$

4.2. Pembahasan Perangkat Lunak (*Software*) AT-Mega16

4.2.1. Inisialisasi Hardware

Inisialisasi Hardware difungsikan untuk menentukan detail hardware yang digunakan beserta sifatnya dan juga hardware external yang terhubung dengan mikrokontroler serta menentukan nilai default dalam keadaan normal. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

```
$regfile = "m16adef.dat"
```

```
$crystal = 11059200
```

```
$baud = 9600
```

```
Relay_on Alias Porta.0
```

```
Relay_start Alias Porta.1
```

```
Lampu_lock Alias Porta.2
```

```
Config Relay_on = Output
```

```
Config Relay_start = Output
```

```
Config Lampu_lock = Output
```

```
Dim X As String * 50
Dim Passw As String * 10
Dim Status_on As Bit
```

```
Passw = "unisfat"
Status_on = 0
```

```
Lampu_lock = 0
Relay_start = 0
Relay_on = 0
```

4.2.2. Baca String Input

Baca String Input difungsikan untuk menerima data string yang dikirim melalui UART. Berikut potongan *software* yang digunakan untuk melakukan tugas tersebut:

```
Input X Noecho
```

4.2.3. Proses Data String

Proses Data Timbang difungsikan untuk mengolah data *String Input*, data tersebut diolah untuk penentuan keputusan mati dan nyalanya lampu LED dan Relay. Berikut potongan *software* yang digunakan untuk melakukan tugas tersebut:

```
Select Case X
```

```
Case "r"
    Waitms 500
    Print "p>>" ; Passw
```

```
Case "ON"
    Waitms 500
    Print "Motor ON !!!"
    Relay_on = 1
```

Lampu_lock = 1

Status_on = 1

Case "START"

Waitms 500

If Status_on = 1 Then

Print "Motor START !!!"

Relay_start = 1

Wait 2

Relay_start = 0

Else

Print "Motor Belum ON !!!"

End If

Case "OFF"

Waitms 500

Print "Motor Lock !!!"

Lampu_lock = 0

Status_on = 0

Relay_on = 0

End Select

4.3. Pembahasan Perangkat Lunak (*Software*) Android



Gambar *Interface* Utama *Software* Android

4.3.1. List Bluetooth Device

List Bluetooth Device difungsikan untuk menampilkan semua device yang terhubung melalui jalur bluetooth. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

```
SerialData.Initialize("SerialData")
```

```
If SerialData.IsEnabled = False Then
```

```
    MsgBox("Aktifkan Bluetooth Dahulu", "")
```

```
    Return
```

```
Else
```

```
    SerialData.Listen
```

```
End If
```

```
Dim PairedDevices As Map
```

```
PairedDevices = SerialData.GetPairedDevices
```



```

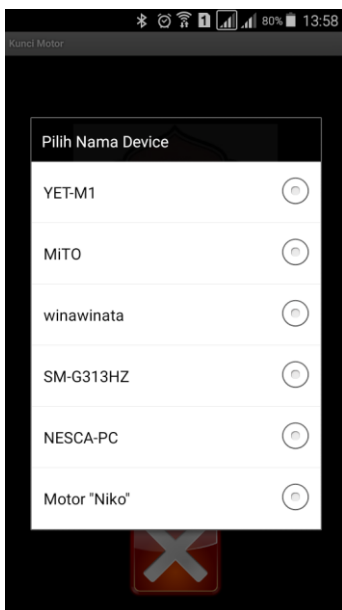
Dim Device As List
Device.Initialize
For i = 0 To PairedDevices.Size - 1
    Device.Add(PairedDevices.GetKeyAt(i))
Next

Dim res As Int
res = InputList(Device, "Pilih Nama Device", -1) 'show
list with paired devices
If res <> DialogResponse.CANCEL Then

    SerialData.Connect(PairedDevices.Get(Device.Get(res)))
'convert the name to mac address
End If

```

Source diatas akan menghasilkan interface seperti berikut:



Gambar *Interface Pemilihan Device Bluetooth*

4.3.2. Kirim string UART

Kirim string UART difungsikan untuk mengirim karakter berupa string melalui bluetooth yang nantinya akan diterima oleh AT-Mega16. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

```
sendData.Initialize(SerialData.OutputStream)
recData.initialize(SerialData.InputStream )
Timer1.Enabled=True
Connected = True

sendData.Write("r" & Chr(13) )
sendData.Flush
```

4.3.3. Terima string UART

Terima string UART difungsikan untuk menerima karakter berupa string melalui bluetooth yang dikirim oleh AT-Mega16. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

```
Sub Time_Tick
If Connected Then
    If recData.Ready Then 'check if there is any data waiting To be read
        datamasuk=recData.ReadLine ' pengiriman data harus di ahiri
        dengan CR

    If datamasuk.Contains("p>>") Then
        Dim id As InputDialog
        Timer1.Enabled = False

        Do While ("p>>" & id.Input <> datamasuk)
            id.Input = ""
```

```

        If id.Show ("", "Masukan Password !!!", "OK",
"Cancel", "", Null ) = DialogResponse.POSITIVE Then
            If id.Input <> "" Then

                If "p>>" & id.Input = datamasuk

Then
                    StartActivity(menu_utama)
                Else
                    MsgBox( "Masukan
Password Lagi" ,"Password Tidak Sesuai !!!")
                End If
            End If
        Else
            SerialData.Disconnect
            Return
        End If
    Loop
End If

End If
End If
End Sub

```

4.3.4. Request Password

Request Password difungsikan untuk mengirim karakter “r” berupa string melalui bluetooth yang nantinya akan diterima oleh AT-Mega16 dan diterjemahkan sebagai *request* password. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

```
sendData.Write("r" & Chr(13) )  
sendData.Flush
```

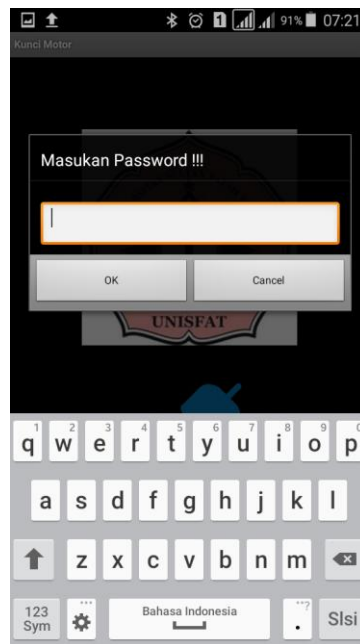
Source tersebut mengirimkan karakter “r” yang nantinya direspon oleh source berikut:

```
Case "r"
```

```
Waitms 500
```

```
Print "p>>" ; Passw
```

Gabungan Source diatas akan menghasilkan interface seperti berikut:



Gambar Interface Request Password

4.3.3. Send String “ON”

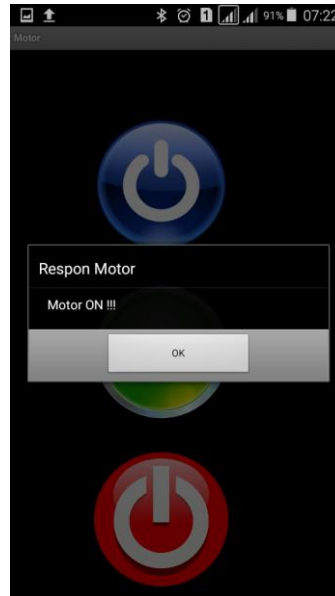
Send String “ON” difungsikan untuk mengirim karakter “ON” berupa string melalui bluetooth yang nantinya akan diterima oleh AT-Mega16 dan diterjemahkan sebagai perintah untuk menyalakan *Relay* “ON”. Berikut potongan *software* yang digunakan untuk melakukan tugas tersebut:

```
Timer1.Enabled=True
Main.datamasuk=""
Main.sendData.Write("ON" & Chr(13) )
Main.sendData.Flush
```

Source tersebut mengirimkan karakter “ON” yang nantinya direspon oleh source berikut:

```
Case "ON"
Waitms 500
Print "Motor ON !!!"
Relay_on = 1
Lampu_lock = 1
Status_on = 1
```

Gabungan Source diatas akan menghasilkan interface seperti berikut:



Gambar Interface Motor ON

4.3.3. Send String “START”

Send String “START” difungsikan untuk mengirim karakter “START” berupa string melalui bluetooth yang nantinya akan diterima oleh AT-Mega16 dan diterjemahkan sebagai perintah untuk menyalakan *Relay* “START” namun dengan syarat kondisi

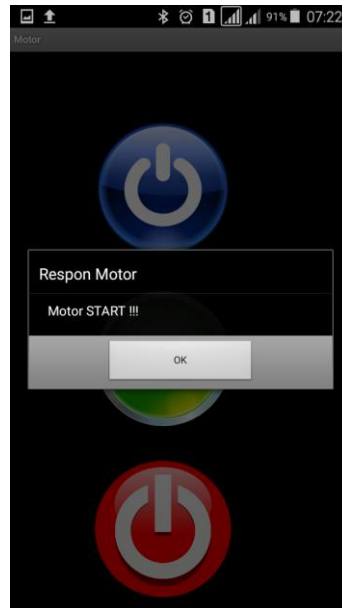
Relay “ON” sudah menyala. Berikut potongan *software* yang digunakan untuk melakukan tugas tersebut:

```
Timer1.Enabled=True  
Main.datamasuk=""  
Main.sendData.Write("START" & Chr(13) )  
Main.sendData.Flush
```

Source tersebut mengirimkan karakter “START” yang nantinya direspon oleh source berikut:

```
Case "START"  
Waitms 500  
If Status_on = 1 Then  
    Print "Motor START !!!"  
    Relay_start = 1  
    Wait 2  
    Relay_start = 0  
Else  
    Print "Motor Belum ON !!!"  
End If
```

Gabungan Source diatas akan menghasilkan interface seperti berikut:



Gambar Interface Motor START

4.3.3. Send String “OFF”

Send String “OFF” difungsikan untuk mengirim karakter “OFF” berupa string melalui bluetooth yang nantinya akan diterima oleh AT-Mega16 dan diterjemahkan sebagai perintah untuk mematikan *relay* “ON”, mematikan *relay* “START” dan menyalakan Lampu LED. Berikut potongan *software* yang digunakan ntuk melakukan tugas tersebut:

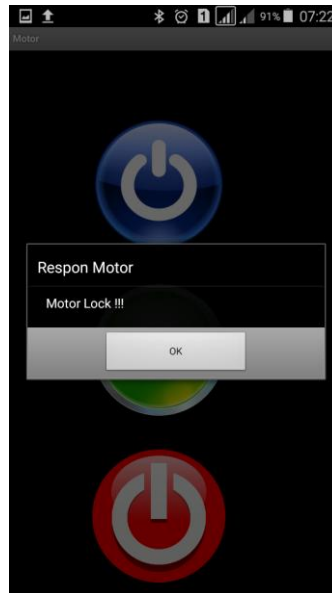
```
Timer1.Enabled=True  
Main.datamasuk=""  
Main.sendData.Write("OFF" & Chr(13) )  
Main.sendData.Flush
```

Sorce tersebut mengirimkan karakter “OFF” yang nantinya direspon oleh source berikut:

```
Case "OFF"  
    Waitms 500  
    Print "Motor Lock !!!"  
    Lampu_lock = 0  
    Status_on = 0  
51
```

Relay_on = 0

Gabungan Source diatas akan menghasilkan interface seperti berikut:



Gambar Interface Motor OFF

BAB V

KESIMPULAN DAN SARAN

Kesimpulan

Setelah melakukan pengujian baik perangkat lunak maupun perangkat keras, maka didapatkan kesimpulan dan saran-saran yang diharapkan berguna bagi pengembangan alat yang telah dibuat.

Dari perancangan penelitian yang telah dikerjakan ini dapat diambil kesimpulan:

1. Jarak jangkauan bluetooth sangat pendek, hanya sekitar 10m tanpa penghalang.
2. Dari segi keamanan sepeda motor menjadi lebih aman dengan adanya proteksi password.
3. Karena sistem keamanan berbasis android sehingga dengan device android yang terinstal aplikasi tersebut dan dengan password tertentu maka semua orang dapat menggunakan motor tersebut.
4. Sistem yang dikembangkan hanya sebatas *prototype* saja, namun tidak menutup kemungkinan untuk di implementasikan secara riil, dengan catatan mengganti semua aktuator yang ada dengan skala yang lebih besar tentunya.

Saran

Dari hasil akhir dalam pembuatan alat ini ada beberapa hal yang dapat disarankan untuk penyempurnaan, yaitu :

1. Jika ingin diaplikasikan kedalam sistem sungguhan maka ada beberapa hal yang perlu diperhatikan, kaitanya dengan jarak jangkauan bluetooth yang terbatas.
2. Sistem yang dibuat terbatas untuk menyalakan motor starter pada sepeda motor dan tidak dilengkapi dengan sensor untuk mendeteksi apakah motor sudah benar-benar menyala atau belum.
3. Untuk fleksibilitas keamanan, diperlukan penambahan fitur perubahan password.

DAFTAR PUSTAKA

- Admin. 2010. "*Bluetooth*". <https://id.wikipedia.org/wiki/Bluetooth>. Diakses tanggal 21 januari 2016.
- Admin. 2016. "*Android Adalah? Pengertian dan Kelebihan HP Android*". <http://pusatteknologi.com/android-adalah.html>. Diakses tanggal 1 Maret 2016.
- Budiharto, Widodo. 2010. "*Robotika Teori dan Implementasi*". Andi. Yogyakarta
- Kho, Dickson. 2015. "*Pengertian Relay dan Fungsinya*". <http://teknikelektronika.com/pengertian-relay-fungsi-relay/>. Diakses tanggal 17 Maret 2016.
- Lingga, W. 2006. "*Belajar sendiri Pemrograman AVR ATmega8535*". Andi Offset. Yogyakarta.
- Putra, Agfianto Eko. 2010. "*Mudah Menguasai Pemrograman Mikrokontroler Atmel AVR menggunakan BASCOM-AVR*". ELINS Universitas Gadjah Mada. Yogyakarta.