



## Rancangan Sistem *Monitoring Indicator LED Transmitter ILS* di Bandar Udara Tjilik Riwut Palangkaraya

### *Design of ILS LED Transmitter Indicator Monitoring System at Tjilik Riwut Palangkaraya Airport*

Jamaluddin Rahim<sup>1</sup>, Seacher Junaedi<sup>2</sup>

[jamaluddinrahim@gmail.com](mailto:jamaluddinrahim@gmail.com), [seacherjunaedi@gmail.com](mailto:seacherjunaedi@gmail.com)

Akademi Teknik dan Keselamatan Penerbangan Makassar

#### ABSTRAK

*Monitoring peralatan Instrument Landing System (ILS) memegang peranan penting dalam pelayanan informasi untuk keselamatan penerbangan, keadaan alat harus selalu di monitor untuk mengantisipasi adanya error. Tujuan perancangan untuk merancang aplikasi berbasis visual basic 6.0 yang berfungsi untuk memantau pergantian Tx main ke Tx standby Instrument Landing System (ILS) di Bandar udara Tjilik Riwut Palangkaraya. Metode perancangan menggunakan program Microsoft visual basic 6.0 dan dikoneksikan dengan menggunakan jaringan Wifi. Hasil perancangan ini menunjukkan kemajuan monitoring terhadap pergantian Tx main ke Tx stand by yang akan mampu mengatasi ketidakstabilan keselamatan. Untuk itu, disarankan menyusun suatu system monitoring yang lebih canggih lagi untuk peralatan ILS (Instrument Landing System).*

*Kata kunci: aplikasi; monitoring; Instrument Landing System*

#### ABSTRACT

*Monitoring indicator led equipment Instrument Landing System (ILS) plays an important role in information services for aviation safety, the situation should always be monitored to anticipate the error on. The purpose of planning to design applications based on visual basic 6.0, to monitor the change of main Tx to Tx standby Instrument Landing System (ILS \) at the airport Tjilik Riwut Palangkaraya. Design using Microsoft Visual Basic 6.0 and connect using Wi-Fi networks. The results of this planning monitoring progress towards the turn of the main Tx to Tx stand by to be able to overcome the instability of security. Therefore, it is advisable to arrange a more sophisticated monitoring system for equipment ILS (Instrument Landing System).*

*Keywords: design; monitoring; Instrument Landing System*

#### 1. PENDAHULUAN

Bandar udara merupakan pintu gerbang kedatangan dan keberangkatan bagi pemakai I jasa transportasi udara baik tingkat wilayah nasional maupun internasional oleh karena itu pelayanan dan kelancaran jasa transportasi udara harus di tunjang dengan berbagai

fasilitas yang cepat, tepat, aman dan nyaman yang selalu siap beroperasi.

Perkembangan dunia penerbangan saat ini menuntut setiap Bandar Udara memanfaatkan fasilitas-fasilitas Telekomunikasi dan Navigasi Udara yang termasuk berkategori teknologi canggih dan

tingkat keselamatan tinggi, Penyelenggaraan penerbangan di wilayah Negara Republik Indonesia dilaksanakan berdasarkan pada Undang-Undang Nomor 1 Tahun 2009 tentang Penerbangan, *Civil Aviation Safety Regulation (CASR)* serta peraturan-peraturan yang mencakup ada peraturan internasional yaitu *International Civil Aviation Organization (ICAO)*. Di dalam regulasi penerbangan tersebut diatur mengenai keamanan dan keselamatan penerbangan sejak penentuan lokasi Bandar Udara hasil dari studi kelayakan sehingga diharapkan Bandar Udara yang akan dibangun menjamin keamanan dan keselamatan operasional penerbangan. Fasilitas penerbangan yang sangat berperan penting untuk keselamatan dan keamanan penerbangan adalah peralatan navigasi *Instrument Landing System*.

Tingkat keamanan suatu bandar udara pun, harus mendapat perhatian yang extra karena apabila kurangnya perhatian, maka dapat menimbulkan bahaya di dalam dunia penerbangan serta jika dibiarkan begitu saja, para pengguna jasa transportasi udara akan merasa kurang aman, nyaman, serta kurang efisien.

Informasi Tx main dan Tx stand by peralatan *Instrument Landing System* berada pada *shelter*. Pada saat terjadi perpindahan Tx teknisi tidak mengetahui kondisi peralatan karena teknisi tidak selalu berada dalam *shelter*, hal ini disebabkan karena belum adanya system monitoring alarm yang berada pada ruang standby teknisi sehingga mengakibatkan kurang efisiennya pengawasan terhadap peralatan.

Seiring dengan perkembangan teknologi, berbagai peralatan navigasi telah berbasis komputerisasi, sehingga lebih memudahkan untuk memonitoring peralatan. Untuk itu, dalam meningkatkan pengawasan peralatan, sehingga hal-hal yang kurang efisien seperti lambatnya para teknisi mengetahui keadaan yang terjadi pada peralatan dapat diselesaikan, dimana kewajiban teknisi yang harus dengan cepat, tepat, dan keselamatan yang terpenting. Pada indicator LED transmitter peralatan *Instrument Landing System (ILS)* pada waktu dini atau dalam artian memantau di tempat dimana para teknisi *standby*.

## 2. METODE PERANCANGAN

Waktu pelaksana perancangan yaitu bulan September 2015 sampai dengan Agustus 2016 pada lokasi Bandar Udara Tjilik Riwut Palangkaraya, menggunakan bahasa pemrograman Visual Basic 6.0 dan Arduino Uno.

*Visual BASIC* (Beginners All-Purpose Symbolic Instruction Code) merupakan sebuah bahasa pemrograman yang dapat digunakan untuk membuat suatu aplikasi dalam Microsoft Windows. *Visual BASIC* menggunakan metode *Graphical User Interface (GUI)* dalam pembuatan program aplikasi (*project*). Istilah visual mengacu pada metode pembuatan tampilan program (*Interface*) atau objek pemrograman yang biasa dilakukan secara langsung terlihat oleh programmer. Dalam *Visual BASIC*, pembuatan program aplikasi harus dikerjakan dalam sebuah *project*. Sebuah *Project* dapat terdiri dari File Project (.vbp), File Form (.frm), File data binary (.frx), Modul Class (.cls), Modul Standar (.bas), dan file resource tunggal (.res). Bahasa yang digunakan adalah bahasa BASIC yang sangat populer pada era sistem operasi DOS.

Dalam rancangan ini pula penulis menggunakan arduino sebagai pengendali mikro *single-board* yang bersifat *open source*, diturunkan dari *Wiring Platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino. Arduino UNO juga adalah sebuah board mikrokontroler yang didasarkan pada ATmega328.

Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang kerja mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya. Arduino Uno berbeda dari semua board Arduino sebelumnya, Arduino

UNO tidak menggunakan chip driver FTDI USB-to-serial. Sebaliknya, fitur-fitur Atmega16U2 (Atmega8U2 sampai ke versi R2) diprogram sebagai sebuah pengubah USB ke serial. Revisi 2 dari board Arduino Uno mempunyai sebuah resistor yang menarik garis 8U2 HWB ke ground, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU mode.

Dalam perancangan ini, penulis merancang monitoring transmitter menggunakan LED. *Instrument Landing System* adalah alat bantu pendaratan secara instrument (non visual) yang digunakan pilot dalam melakukan prosedur pendekatan dan pendaratan pesawat di suatu bandara walaupun dalam kondisi visibility yang minim. ILS dioperasikan beserta alat pendukung navigasi lain seperti DME, VOR, NDB dan Compass Locator.

ILS terdiri dari: VHF Localizer, UHF Glide Path dan VHF Marker Beacon. Pada interlock system, switching antara satu ILS dengan lainnya harus kurang dari 20 detik. Adapun rancangan system yang akan dibuat meliputi perancangan perangkat keras dan perancangan perangkat lunak.

Dari segi perancangan perangkat keras dilakukan dengan cara terlebih dahulu mengukur tegangan yang terdapat pada indicator ILS ketika LED dalam kondisi aktif/ON (pemancar/Transmitter dalam kondisi normal mancar) begitu pula pada kondisi padam/OFF (pemancar/Transmitter dalam kondisi standby). Selanjutnya, setelah nilai tegangan output di Indicator LED di temukan selanjutnya, dihubungkan dengan port input microcontroller agar statusnya dapat ditransmisikan ke ruang stanby teknisi.

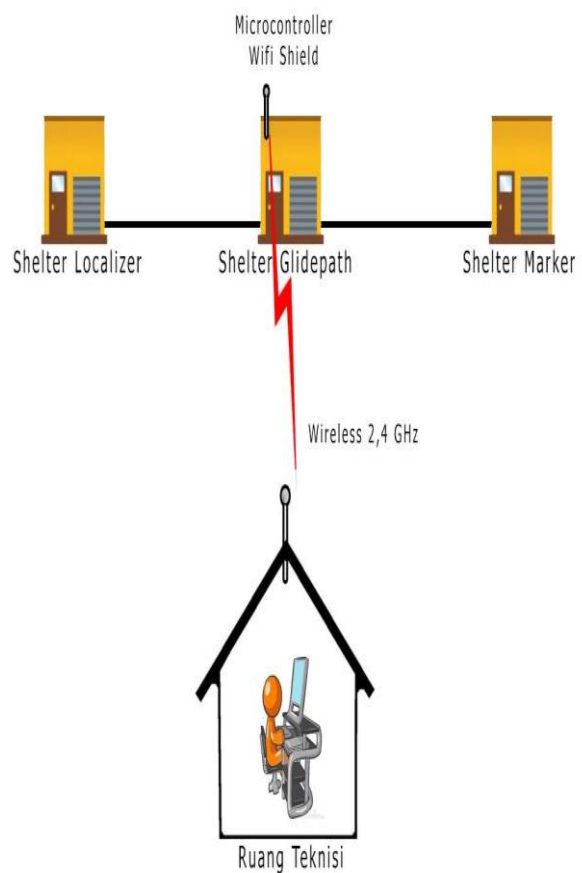
Dalam proses pengiriman data dari masing-masing *Shelter* idealnya masing-masing dilengkapi dengan microcontroller, namun mengingat pada penelitian peniliti berfokus pada fungsional system yang dirancang maka pembacaan status indicator dari masing-masing ILS pada *Shleter* cukup diwakili dengan sebuah *microcontroller*, dengan pertimbangan bahwa I/O pada *microcontroller* yang digunakan masih memenuhi kebutuhan I/O untuk memonitoring status indicator transmisi ILS.

Untuk indicator di *Shelter Localizer* dan *Marker* disatukan pada *Shelter glide path indicator* yang ada pada *localizer* dan *marker*

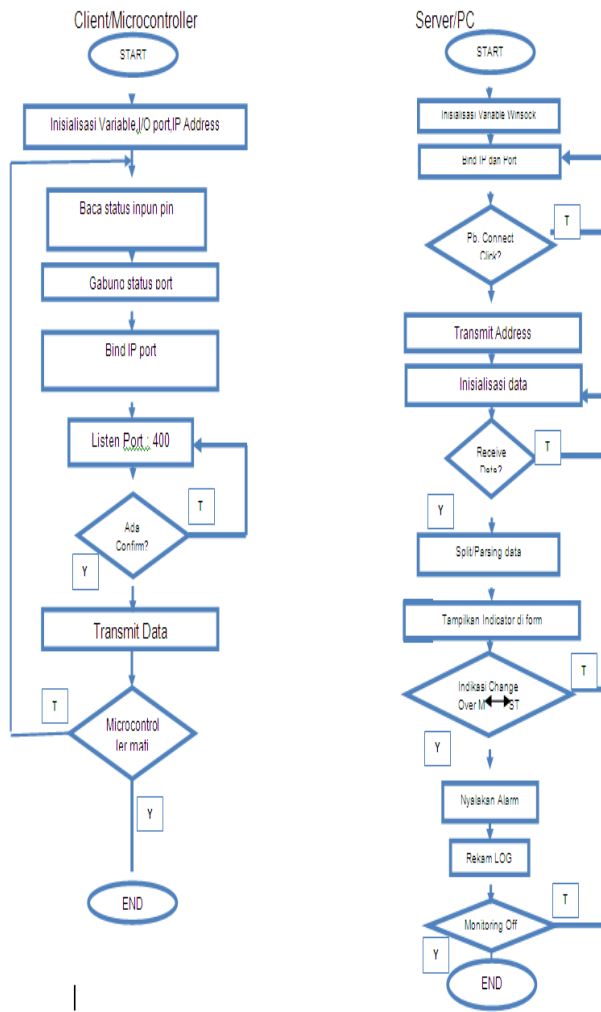
di hubungkan ke input microcontroller yang terdapat pada *Shelter Glide Path* yang selanjutnya akan di transmisikan ke ruang *standby* teknisi memakai media transmisi *wireless* pada *frequency* 2,4 Ghz.

Selanjutnya agar data yang ditransmisikan dapat ditampilkan pada PC atau Laptop di ruang standby teknisi sekaligus memberikan peringatan serta laporan terkait jika terjadi kondisi kegagalan transmisi pada masing-masing *shelter*, maka diperlukan sebuah aplikasi yang rencananya akan di implementasikan dengan menggunakan teknik pemrograman socket pada *Visual Basic* 6.0. Pada aplikasi desktop yang dirancang terdapat beberapa fitur diantaranya: fungsi koneksi ke microcontroller, fungsi parsing data, fungsi alarm, laporan.

Secara keseluruhan *system* yang dirancang pada penelitian ini digambarkan pada gambar di bawah ini.



Gambar 1. Sistem operasional peralatan navigasi yang ideal.

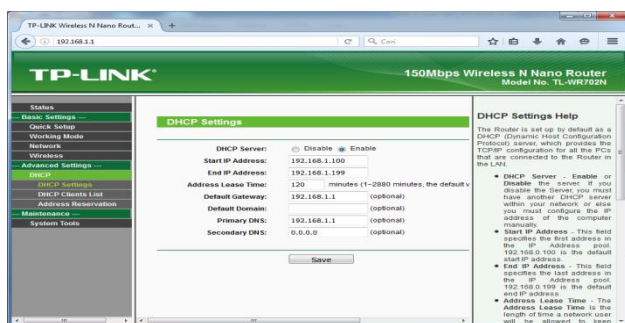


Gambar 2. Flow Chart System

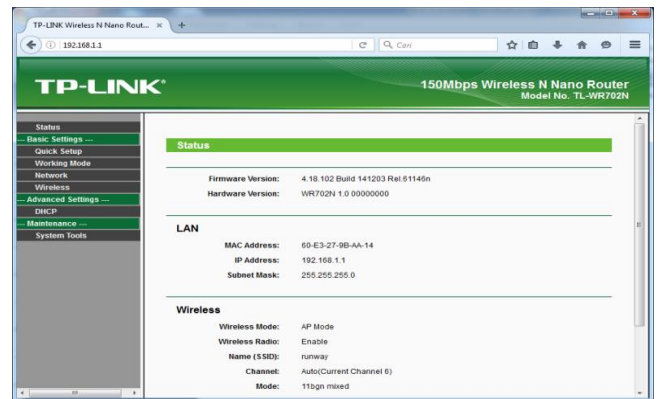
### 3. HASIL DAN PEMBAHASAN Operasi System

#### a. Implementasi Hardware

- 1) Konfigurasi Access Point  
 Pertama-tama kita harus mengkonfigurasi *Access Point* yang akan kita pakai.

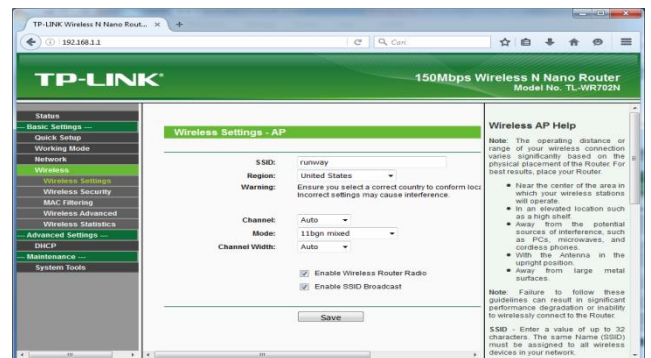


Gambar 3. Konfigurasi LAN IP Address dan DHCP Address  
 (Sumber: Hasil Rancangan)



Gambar 4. Pengkonfigurasi Nama Access Point  
 (Sumber: Hasil Rancangan)

Jadi, adapun langkah-langkah yang harus di perhatikan dalam pengkonfigurasi yaitu sebagai berikut: Pertama buka browser yang anda pakai, setelah itu tulis lah di kolom alamat “192.168.1.1” lalu tekan Enter. Setelah itu akan muncul gambar seperti ini.



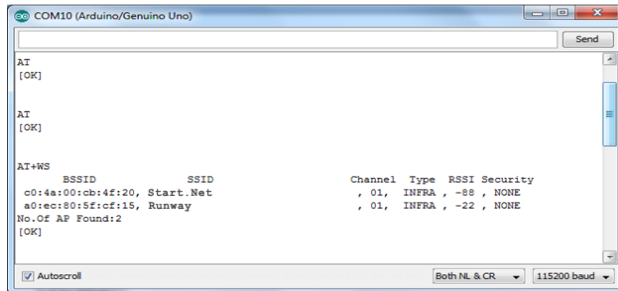
Gambar 5. Tampilan awal dari web 192.168.1.1  
 (Sumber: Hasil Rancangan)

Lalu pilih DHCP Settings di menu ini anda akan mengatur LAN IP Address dan DHCP Address yang akan anda pakai. Ketika selesai pengkonfigurasi LAN dan DHCP lalu pilih menu Wireless >> Wireless Settings Di menu ini anda akan menamai SSID yang akan anda pakai.

- 2) Konfigurasi Wifi Shield

Adapun langkah untuk mengkonfigurasi *Wifi Shield* menggunakan perintah *AT Command* dengan cara menghubungkan *Wifi Shield* ke Arduino selanjutnya dikoneksikan melalui kabel USB lalu mengetik beberapa perintah *AT command* dasar mulai dari melakukan proses scan *Access Point* yang akan dikoneksikan,

mononaktifkan fitur DHCP pada *Wifi Shield* menentukan *IP Address* dan *Port* untuk komunikasi data ke aplikasi desktop.



**Gambar 6. Pengkonfigurasi Wifi Shield**  
 (Sumber: Hasil Rancangan)

AT  
 [OK]

Scan Access Point

AT+WS  
 BSSID SSID Channel  
 Type RSSI Security  
 c0:4a:00:cb:4f:20, Start.net , 01,  
 INFRA , -88 , NONE  
 a0:ec:80:5f:cf:15, Runway ,  
 01, INFRA , -22 , NONE  
 No.Of AP Found:2  
 [OK]

Nonaktifkan fitur DHCP

AT+NDHCP=0  
 [OK]

Set IP Address

AT+NSET=192.168.1.212,255.255.255.0,192.  
 168.1.1  
 [OK]

Set Port : 400

AT+NAUTO=1,1,,400  
 [OK]

Simpan Konfigurasi

AT&W0  
 [OK]

Disable Auto konek ketika direstart

ATC0  
 [OK]

Set Auto konek ketika direstart

ATC1  
 [OK]

ATA  
 IP SubNet Gateway  
 192.168.1.212: 255.255.255.0: 192.168.1.1  
 [OK]

3) Konfigurasi Arduino Uno

```
int tbl1 = 0;
int tbl2 = 0;
int tbl3 = 0;
int btnSimulasi = 13;

//Localizer
int TxMLocalizer =2;
int TxSLocalizer =3;

//Glide
int TxMGlide = 4;
int TxSGlide = 5;
//Marker
int TxMMarker = 6;
int TxSMarker = 7;

int btnLocalizer = 8;
int btnGlide= 9;
int btnMarker = 10;

int ValbtnLocalizer = 0;
int ValbtnGlide= 0;
int ValbtnMarker = 0;

int ValTxMLocalizer =0;
int ValTxSLocalizer =0;
int ValTxMGlide = 0;
int ValTxSGlide = 0;
int ValTxMMarker = 0;
int ValTxSMarker = 0;

void setup() {
Serial.begin(115200);
KirimStatus();

pinMode(btnLocalizer, INPUT);
pinMode(btnGlide, INPUT);
pinMode(btnMarker, INPUT);

pinMode(TxMLocalizer, OUTPUT);
pinMode(TxMGlide, OUTPUT);
pinMode(TxMMarker, OUTPUT);
pinMode(TxSLocalizer, OUTPUT);
pinMode(TxSGlide, OUTPUT);
pinMode(TxSMarker, OUTPUT);
```

```
digitalWrite(TxMLocalizer, HIGH);
digitalWrite(TxMGlide, HIGH);
digitalWrite(TxMMarker, HIGH);

digitalWrite(TxSLocalizer, LOW);
digitalWrite(TxSGlide, LOW);
digitalWrite(TxSMarker, LOW);
delay(1000);
KirimStatus();
}

void loop() {

int ValbtnLocalizer =
digitalRead(btnLocalizer);
int ValbtnGlide = digitalRead(btnGlide);
int ValbtnMarker = digitalRead(btnMarker);

if(ValbtnLocalizer == 1){
digitalWrite(TxMLocalizer, LOW);
digitalWrite(TxSLocalizer, HIGH);
}else{
digitalWrite(TxMLocalizer, HIGH);
digitalWrite(TxSLocalizer, LOW);
}

if(ValbtnGlide == 1){
digitalWrite(TxMGlide, LOW);
digitalWrite(TxSGlide, HIGH);
}else{
digitalWrite(TxMGlide, HIGH);
digitalWrite(TxSGlide, LOW);
}

if(ValbtnMarker == 1){
digitalWrite(TxMMarker, LOW);
digitalWrite(TxSMarker, HIGH);
}else{
digitalWrite(TxMMarker, HIGH);
digitalWrite(TxSMarker, LOW);
}
KirimStatus();
}

void KirimStatus(){
int ValTxMMarker =
digitalRead(TxMMarker);
int ValTxSMarker = digitalRead(TxSMarker);
int ValTxMGlide = digitalRead(TxMGlide);
int ValTxSGlide = digitalRead(TxSGlide);
int ValTxMLocalizer =
digitalRead(TxMLocalizer);
```

```
int ValTxSLocalizer =
digitalRead(TxSLocalizer);

Serial.print (ValTxMLocalizer);
Serial.print (",");
Serial.print (ValTxSLocalizer);
Serial.print (",");
Serial.print (ValTxMGlide);
Serial.print (",");
Serial.print (ValTxSGlide);
Serial.print (",");
Serial.print (ValTxMMarker);
Serial.print (",");
Serial.println (ValTxSMarker);
delay(500);
}
```

Output data yang yang disajikan pada Microcontroller akan berbentuk data *array* yang dipisahkan dengan tanda "koma".

```
int ValTxMGlide = digitalRead(TxMGlide);
int ValTxSGlide = digitalRead(TxSGlide);
int ValTxMLocalizer = digitalRead(TxMLocalizer);
int ValTxSLocalizer = digitalRead(TxSLocalizer);

Serial.print (ValTxMLocalizer);
Serial.print (",");
Serial.print (ValTxSLocalizer);
Serial.print (",");
Serial.print (ValTxMGlide);
Serial.print (",");
Serial.print (ValTxSGlide);
Serial.print (",");
Serial.print (ValTxMMarker);
Serial.print (",");
Serial.println (ValTxSMarker);
delay(500);
}
|
```

**Gambar 7. Pengupload-an data**  
 (Sumber: Hasil Rancangan)

```
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
1,0,1,0,1,0
```

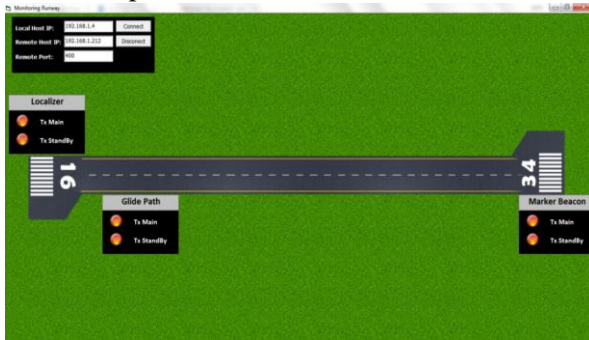
**Gambar 8. Output status Indikator LED pada Microcontroller**  
 (Sumber: Hasil Rancangan)

Selanjutnya agar dapat di visualisasikan dalam aplikasi yang mudah dipahami maka diperlukan teknik parsing data sehingga nantinya diperoleh kondisi dari masing-masing peralatan.

b. Implementasi Software

Agar pengiriman data dari Microcontroller dapat diterima pada aplikasi desktop maka diperlukan komponen untuk bisa berkomunikasi melalui jaringan TCP/IP, dalam hal ini peneliti menggunakan komponen Winsock (*Windows Socket*).

1) Tampilan *Visual Basic*



**Gambar 9. Tampilan *Visual Basic***  
 (Sumber: Hasil Rancangan)

2) *Form Connect*

Di *form connect* ini kita dapat mengoneksikan aplikasi VB ini dengan Arduino Uno, tetapi pertama-tama kita harus memperhatikan IP kita, IP *Remote*

```
.RemoteHostIP = txtRemoteHost.Text '
IP Address Microcontroller
.RemotePort = txtRemotePort.Text ' Port
yang digunakan Microcontroller

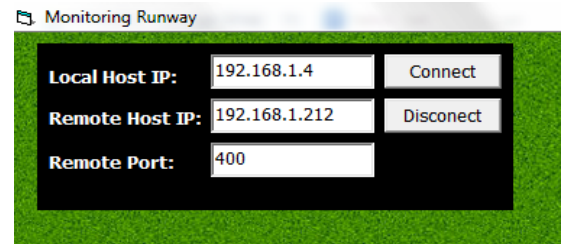
If Not .OpenConnection Then ' Jika
koneksi gagal dilakukan
Timer1.Enabled = False ' Maka non
aktifkan timer untuk parsing data
TampilError "Koneksi Ke
Microcontroller Gagal, Periksa Konfigurasi IP
Address anda" Else
```

Fungsi dari bahasa program di atas yaitu, untuk mengkoneksikan aplikasi Visual Basic dengan microcontroller Arduino Uno. Jadi ketika tombol connect ditekan maka akan memulai perintah untuk mengkoneksikan aplikasi Visual Basic ini dengan Microcontroller Arduino Uno.

b. Implementasi program untuk memutuskan koneksi ke microcontroller

```
Private Sub cmdCloseConnection_Click()
'Putuskan Koneksi ke Microcontroller
On Error Resume Next
tcp.ShutdownConnection
```

wifi dan *Port* yang akan kita pakai Perhatikan pada gambar 10.



**Gambar 10. Form Connect**  
 (Sumber: Hasil Rancangan)

a. Implementasi koneksi ke Microcontroller menggunakan Winsock

```
Private Sub cmdConnect_Click ()
' Fungsi untuk konek ke Microcontroller
Set tcp = New VBWinsock.TCPIP
With tcp
.LocalHostIP = txtLocalHost.Text ' IP
Address komputer yang menggunakan
Aplikasi
```

```
'Sebaliknya jika koneksi berhasil
aktifkan timer untuk mem parsing data dari
Micro
Timer1.Enabled = True
End If
End With
End Sub
```

```
Timer1.Enabled = False
Set tcp = Nothing
End Sub
```

Fungsi dari bahasa program di atas yaitu, untuk memutuskan koneksi antara aplikasi *Visual Basic* dengan *Microcontroller* Arduino Uno. Jadi ketika tombol *Disconnect* ditekan maka akan memulai perintah untuk memutuskan koneksi antara aplikasi *Visual Basic* dengan *Microcontroller* Arduino Uno.

c. Implementasi Laporan

```
Private Sub CmdLaporan_Click()
'Fungsi untuk menampilkan Folder
Laporan
ShellExecute Me.hwnd, "Open", App.Path
& "\Laporan\", vbNullString,
vbNullString, SW_SHOWNORMAL
End Sub
```

Bahasa program di atas untuk menampilkan folder tempat dimana log laporan itu berada, Jadi ketika tombol laporan di tekan maka langsung akan membuka folder tempat log laporan itu berada.

Fungsi dari bahasa program diatas adalah untuk proses change transmitter yang sedang aktif, sehingga memudahkan monitoring.

e. Implementasi parsing data

```
Private Sub Timer1_Timer()
' Timer untuk melakukan parsing data
(Split data yang diterima dari
Microcontroller)
On Error Resume Next
Dim strData As String
Dim l As Long

If tcp.IsDataAvailable Then
```

4) Implementasi visualisasi indikator status Runway

```
Private Sub TXTStatus_Change(Index As
Integer)
' Event/Kejadian (Kondisi) Jika terjadi
perubahan data dari Microcontroller
If TXTStatus(Index).Text = "1" Then
```

End If

```
If Index = 1 Or Index = 3 Or Index = 5 Then
If TXTStatus(1).Text = "1" Then
BunyiAlarm
SimpanLaporan Now & vbTab & " TX
Main Localizer Berpindah ke Tx Stanby "
```

```
End If

If TXTStatus (3). Text = "1" Then
BunyiAlarm
```

```
End If
Then
TampilError "Tidak dapat
menerima data"
Else
txtReceive.Text = strData
TxtData.Text =
Mid(txtReceive.Text,
Len(txtReceive.Text) - 12, 12)
' Parsing Data
TXTStatus(0).Text = Left(TxtData.Text,
1)
TXTStatus(1).Text = Mid(TxtData.Text,
3, 1)
TXTStatus(2).Text = Mid(TxtData.Text,
5, 1)
TXTStatus(3).Text = Mid(TxtData.Text,
7, 1)
TXTStatus(4).Text = Mid(TxtData.Text,
9, 1)
TXTStatus(5).Text = Mid(TxtData.Text,
11, 1)

End If
End If
End Sub
```

Fungsi dari bahasa program di atas yaitu, untuk melakukan parsing data/split data dimana parsing data itu untuk menerima data dari *Microcontroller* Arduino Uno.

'Jika hasil parsing data yang diterima 1 maka

```
'Tampilkan indicator ON
ImgStatus(Index).Picture = ImgOn.Picture
Else
'Sebaliknya jika bukan 1 maka tampilkan
indicator OFF
ImgStatus(Index).Picture = ImgOff.Picture
```

SimpanLaporan Now & vbTab & " TX Main Glide Berpindah ke Tx Stanby"

End If

```
If TXTStatus(5).Text = "1" Then
BunyiAlarm
SimpanLaporan Now & vbTab & " TX
Main Marker Berpindah ke Tx Stanby"
```

End If



End Sub

Bahasa program di atas berfungsi sebagai, indicator penanda dalam kondisi jika terjadi perubahan data dari Microcontroller Arduino Uno, jadi ketika adanya perubahan data maka akan berubah pula indicator di aplikasi Visual Basic tersebut.

f. Implementasi fungsi untuk mengaktifkan alarm

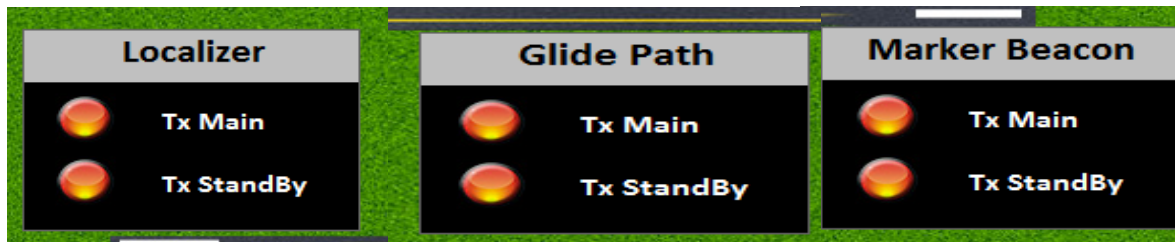
```
Option Explicit
'Api Untuk memutar suara
Private Declare Function sndPlaySound Lib
"winmm.dll" Alias "sndPlaySoundA" (ByVal
IpszSoundName As String, ByVal uFlags As
Long) As Long
Private Const SND_ASYNC = &H1
Public Const pcsSYNC = 0
Private Declare Function Beep Lib "kernel32"
_
```

```
(ByVal dwFreq As Long, _
ByVal dwDuration As Long) As Long
```

```
Public Sub BunyiAlarm() ' Fungsi untuk
menyalakan alarm
DoEvents
Call sndPlaySound(App.Path &
"\Alarm.wav", SND_ASYNC)
End Sub
```

Fungsi dari bahasa program di atas yaitu, untuk memutar suara/alarm yang sudah ditentukan, program di atas akan berfungsi ketika terjadi perubahan pada indicator, Jadi pada saat adanya perubahan data dari Microcontroller Arduino Uno indicator pada aplikasi Visual Basic akan berubah dan alarm akan langsung berbunyi

3. Indicator Localizer, Glide Path, dan Marker Beacon



Gambar 11. Indicator Localizer, Glide Path, dan Marker Beacon  
 (Sumber: Hasil Rancangan)

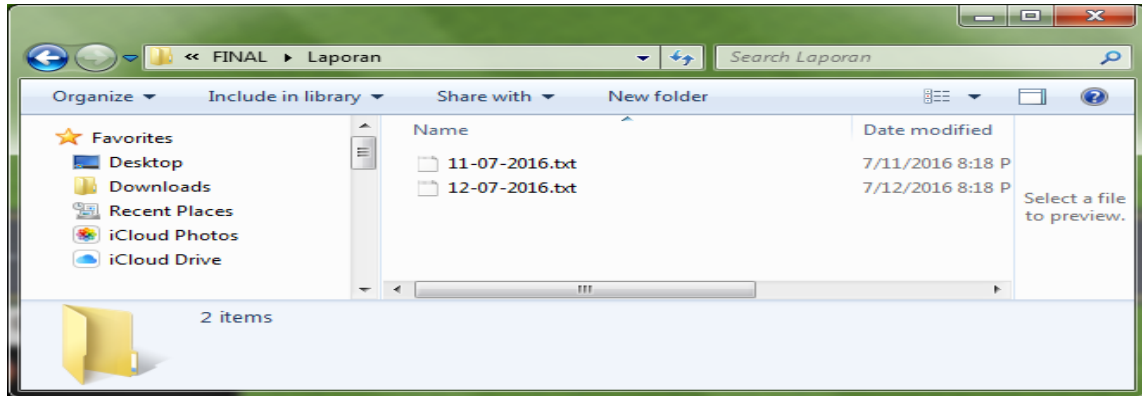
Indikator tersebut akan member tanda apakah pada saat sekarang Tx Main localizer yang lagi on ataukah Tx Standby yang lagi on

begitu pula dengan Glide Path dan Marker Beacon, Jadi teknisi dapat memonitoring langsung Tx yang yang sedang bekerja.

4. Format Log Laporan

Setiap adanya perubahan pergantian akan di catat danmasuk di log laporan, berikut gambar Format Log Laporan yang diterima ketika adanya perubahan pergantian. Lihatlah pada gambar 12.

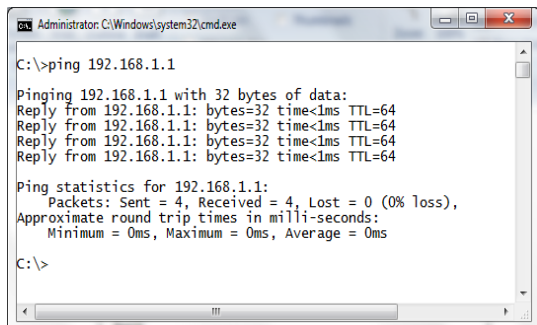
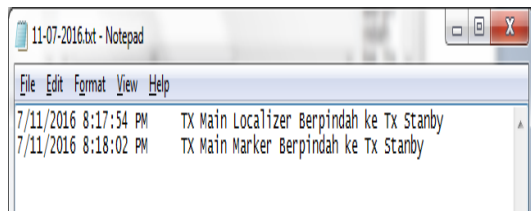
Jadi, ketika terjadi perpindahan/pergantian Tx akan langsung terkirim dan dimuat dalam log laporan tersebut, contoh format laporannya disertai dengan tanggal/bulan/tahun dan jam pada saat terjadinya perpindahan/pergantian Tx



Gambar 12. Format Log Laporan  
(Sumber: Hasil Rancangan)

## A. Uji Hasil

### 1. Uji koneksi ke Access Point



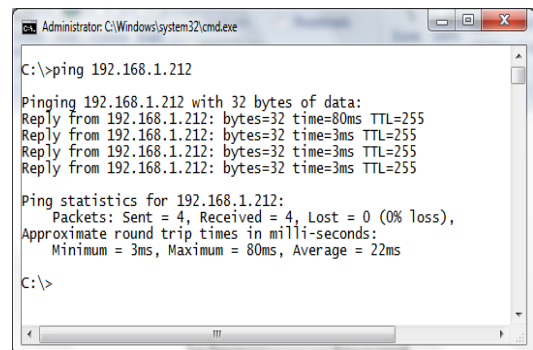
Gambar 13. Uji koneksi ke Access Point  
(Sumber: Hasil Rancangan)

Gambar di atas menjelaskan bahwa Koneksi ke Access Point telah berhasil, Jadi cara untuk mengecek koneksi ke Access Point yaitu Pertama-tama membuka cmd.exe, dengan menekan windows + R secara ni. Lalu setelah Run.exe muncul tulislah “cmd “ dan tekan tombol “ OK “ Setelah itu tulislah “ ping 192.168.1.(address yang kamu buat pada saat konfigurasi Access Point) “.

Mengecek koneksi ke *Access Point* yaitu Pertama-tama membuka cmd.exe, dengan menekan windows + R secara ni. Lalu setelah Run.exe muncul tulislah “cmd“ dan tekan tombol “ OK “ Setelah itu tulislah “ ping

192.168.1.(angka address yang kamu buat pada saat konfigurasi Access Point) “.

### 2. Uji Koneksi ke microkontroller



Gambar 14. Uji koneksi ke Microcontroller  
(Sumber: Hasil Rancangan)

Seperti yang telah teruji pada hasil di atas, peneliti dapat membuat tabel perbandingan dan tabel pembuktian, dimana tabel perbandingan dan pembuktian ini untuk melihat hasil dari rancangan yang telah di buat yaitu waktu teknisi untuk mengetahui status peralatan dan sinkronasi pemakaian indicator LED ke microcontroller, perhatikan tabel-tabel di bawah ini. Lalu setelah Run.exe muncul tulislah “cmd“ dan tekan tombol “OK“, Setelah itu tulislah “ ping 192.168.1.

**Tabel 1. Hasil Rancangan**

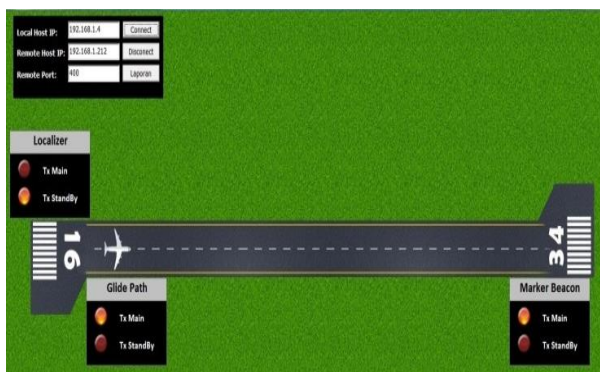
Peralatan	Waktu yang di tempuh tanpa rancangan	Waktu yang di tempuh dengan rancangan
Localizer	± 5 menit	1 > detik
Glide Path	± 7 menit	1 > detik
Marker Beacon	-	-

(Sumber: Hasil Rancangan)

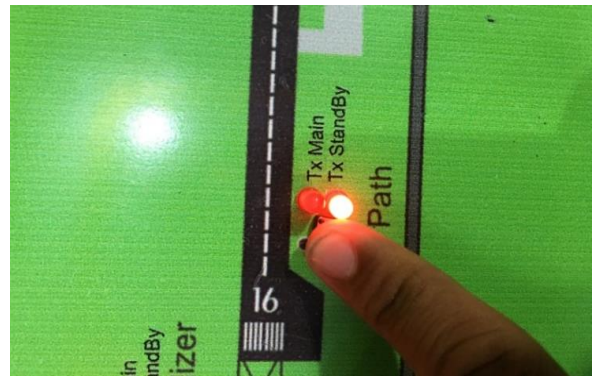
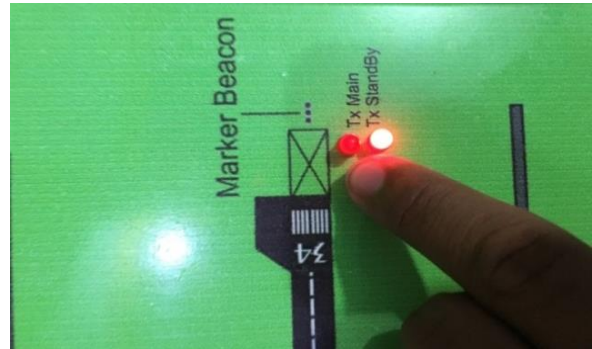
**Tabel 3. Perbandingan waktu tempuh**

Peralatan	Kondisi LED		Kondisi Microcontroller
	Hidup	Mati	
Tx Main Localizer	2 V		HIGH
Tx Standby Localizer		0,2 V	LOW
Tx Main Glide Path	2 V		HIGH
Tx Standby Glide Path		0,2 V	LOW
Tx Main Marker Beacon	2 V		HIGH
Tx Standby Marker Beacon		0,2 V	LOW

(Sumber: Hasil Rancangan)



**Gambar 15. Lay out Runway**  
 (Sumber: Hasil Rancangan)



**Gambar 17. Pengiriman data berhasil Localizer**  
 (Sumber: Hasil Rancangan)

#### 4. KESIMPULAN

Kesimpulan yang dapat ditarik adalah sebagai berikut:

- Dengan rancangan ini dapat membantu teknisi mengetahui kondisi peralatan ILS, dengan adanya monitoring status peralatan diruang teknisi. Untuk mencapai koneksi jaringan yang kuat dapat mengganti TP-Link yang mempunyai power yang lebih kuat.
- Teknisi lebih efisien memantau kondisi peralatan *Localiser*, *Glide Path*, dan *Marker Beacon* karena tidak perlu lagi ke lokasi *equipment*, tetapi hanya memonitor lewat *indicator*.

## DAFTAR PUSTAKA

- Akbar, Ali. (2005). *Pemrograman/Visual Basic Programming (Online)*. Surabaya: PT Sinar Indah Komputindo. Jakarta. 12 Februari 2014.
- Baker, Robert W. (2002). Research with The Student Adaptation to College Questionnaire (SACQ). diunduh dari mtholyoke.edu
- Hamadillah, Ajie. *Pengantar Dasar Pemrograman (6 jam)*. Jakarta: Universitas Negeri Jakarta, Fakultas Teknik.
- International Civil Aviation Organization. (2005). *Aerodrome Design Manual*.
- International Civil Aviation Organization. (2007). *Annex 14 Aerodromes*
- International Civil Aviation Organization, Air Traffic Services, Annex 11 Thirteenth Edition, July 2001.
- International Civil Aviation Organization, Aerodrome Volume Aerodrome Design and Operation, Annex 14 Third Edition, July 1999
- Jugiyanto HM dan Syifaun Nafisahdi Hakim Simanjuntak. (2013), *Pengertian Perancangan*.
- Peraturan Direktur Jenderal Perhubungan Udara Nomor: KP 6 Tahun 2014 tentang Tata Cara Pengaturan *Slot Time*.
- Peraturan Pemerintah Republik Indonesia Nomor 3 Tentang Keamanan dan Keselamatan Penerbangan, 2001.
- Rahim, Jamaluddin. (2013). *Metode Penelitian Ilmiah*, Makassar, Akademi Teknik dan Keselamatan Penerbangan, Jurusan Keselamatan Penerbangan.
- Rahim, Jamaluddin. (2015). *Materi Metode Penulisan*. ATKP Makassar.
- PP No. 3. (2001). Tentang Keamanan Dan Keselamatan Penerbangan BAB I ayat 1.
- Sugiyono. (2016). *Metode Penelitian Bandung*: CV. ALFABETA
- Thamrin. (2014). *Kumpulan Materi Kuliah Tentang ILS (Instrument Landing System)*. ATKP Makassar.
- Undang–Undang Republik Indonesia Nomor 1 Tahun 2009 Tentang Penerbangan.