

# **Monitoring Kegagalan *Sequence Flashing Lighting* Runway 28 Menggunakan *Fiber Optic* Berbasis *Microcontroller* di Bandar Udara Internasional Juanda Surabaya**

**Hartono  
Wasito Utomo**

**Politeknik Penerbangan Surabaya**  
Jl. Jemur Andayani I/73 Surabaya  
Email : hartono.subawi@yahoo.co.id

## **Abstrak**

*Rancangan ini bertujuan untuk mempermudah teknisi dalam melakukan monitoring sequence flashing lighting di bandara internasional Juanda, karena letak sequence flashing lighting yang jaraknya empat kilometer dari power house dan lokasinya yang sulit dijangkau terutama sequence flashing lighting runway 28 yang berada di tambak-tambak warga. Penelitian ini menggunakan microcontroller sebagai media kontrol dan kabel fiber optic sebagai media komunikasi, untuk mendeteksi nilai kegagalan dari lampu tersebut menggunakan sensor arus ACS712, metode untuk tampilan hasil monitoring menggunakan visual basic yang akan ditampilkan di komputer atau personal computer. Sedangkan untuk komunikasi antara personal computer dengan microcontroller menggunakan kabel UTP atau ethernet dengan dukungan ethernet media converter to serial. Hasil penelitian menunjukkan untuk monitoring nyala dan mati sequence flashing lighting dapat menggunakan personal computer sehingga dengan metode ini kinerja teknisi menjadi lebih cepat dan efisien.*

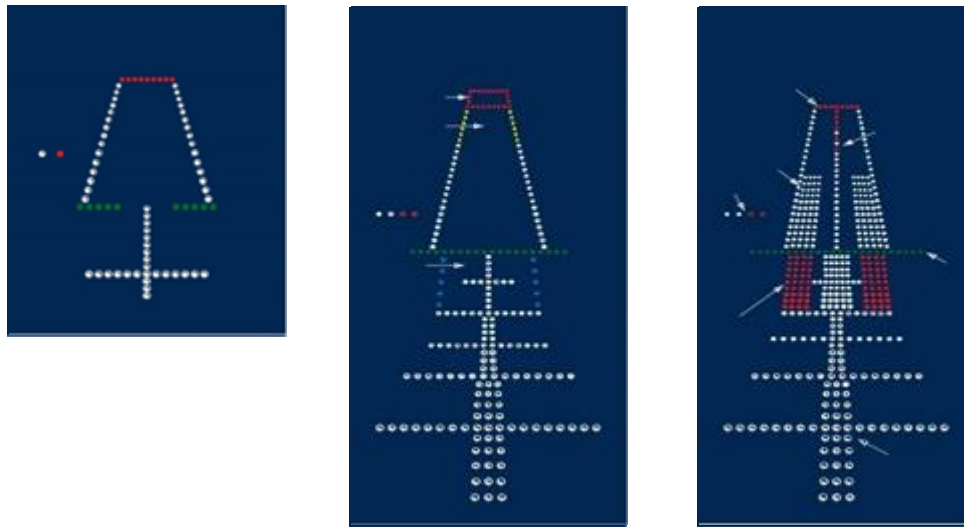
**Kata kunci :** *sequence flashing lighting, microcontroller, fiber optic, ACS712, visual basic.*

## **Latar Belakang Masalah**

Sistem lampu *sequence flashing* adalah berupa susunan lampu yang berguna untuk memandu penerbang melakukan pendekatan ke *runway* dalam proses pendaratan. Dengan melihat rambu ini, penerbang akan mengetahui apakah pesawat sudah sejalur dengan *runway* ataukah belum. Dengan demikian, bila pesawat belum sejalur dengan *runway* atau *centre runway* maka tindakan penerbang adalah menyesuaikan supaya pesawat sejalur dengan *runway*. *Sequence Flashing Lighting* terletak di perpanjangan ujung *runway*. Rambu ini memancarkan cahaya yang berwarna putih secara berurutan mulai dari awal sampai akhir lampu sesuai dengan settingan modul lampu tersebut.

Bandar Udara Juanda konfigurasi lampu *sequence* mengikuti konfigurasi lampu *approach* yaitu dengan menggunakan *PALS (Precision Approach Lighting System)* dengan kategori I. . Berikut adalah gambar konfigurasi berbagai sistem lampu *approach*, mulai dari yang simple, kategori I dan kategori II dan III. (*ICAO Annex 14, Volume I, Juli 1999*).

Sistem pendekatan pencahayaan sederhana (*Simple Approach Lighting System*) dimulai 500 meter sebelum landasan pacu ambang batas. Di Bandara-bandara Inggris menggunakan sistem Calvert, dimana lampu-lampu *approach* dipasang 900 m sebelum landasan pacu ambang batas.

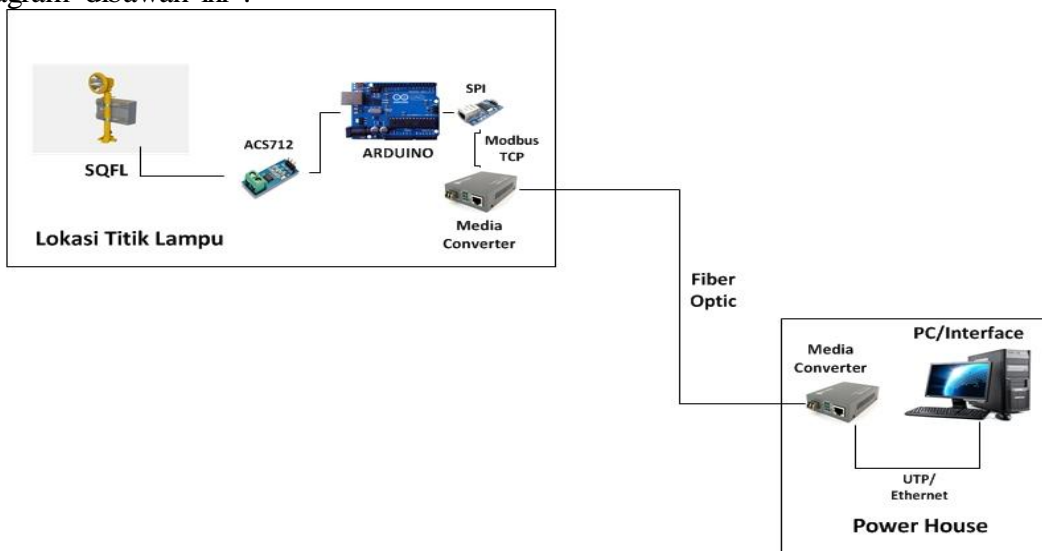


(a) (b) (c)

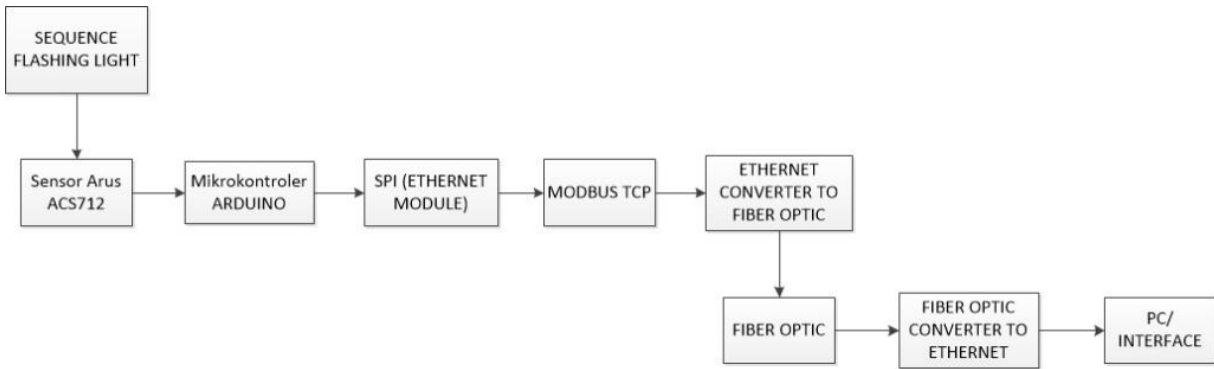
Gambar 1(a) *Simple approach and Runway lighting system* (b) *Approach and runway lighting Typical cat I system showing Take-off starter extension and Stopway lighting* dan (c) *Approach and runway lighting Typical cat II or cat III system*

## METODOLOGI PENELITIAN

Proses *monitoring* ini sepenuhnya dapat dilakukan di tempat yang dekat dengan *power house*. Saat lampu *sequence flashing* menyala, artinya ada arus yang menuju ke beban, arus ini yang dibaca oleh sensor arus. Kemudian, *arduino* mengirimkan perintah dan selanjutnya hasil perintah tersebut ditampilkan dalam sebuah layar komputer. Proses *monitoring* ini diharapkan dapat mempermudah teknisi dalam mengontrol lampu *sequence flashing* yang berada di bandara Juanda. Kondisi yang diinginkan oleh penulis dapat dilihat pada gambar konsep rancangan dan blok diagram dibawah ini :



Gambar 2. Konsep Rancangan

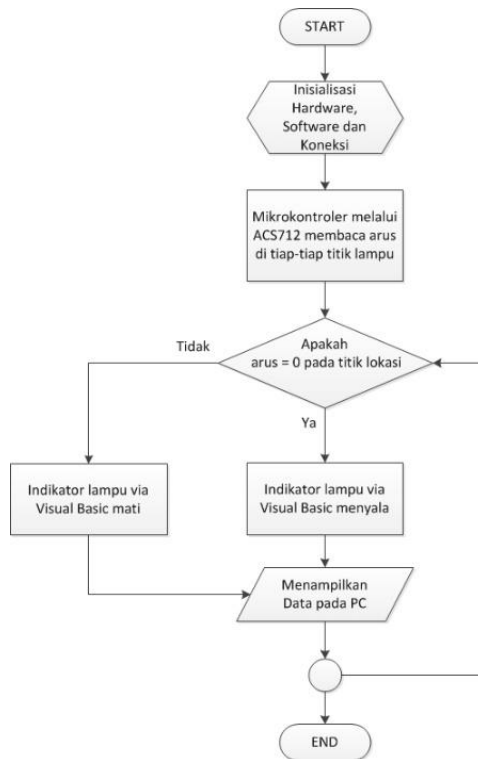


Gambar 3. Blok Diagram Sistem

Gambar blok diagram diatas menjelaskan bagaimana *monitoring* lampu *sequence flashing* dengan menggunakan sensor arus ACS 712. Dengan proses diagram seperti ini, memiliki keuntungan lebih yaitu apabila salah satu lampu *sequence flashing* tersebut tidak berfungsi dengan baik sesuai nilai arus yang seharusnya menuju beban, maka dapat diketahui lampu manakah yang tidak berfungsi dan *monitoring* dapat dilakukan tanpa menuju ke tempat lampu tersebut berada. Dengan demikian akan mempermudah dalam pengontrolan maupun dalam *monitoringnya*.

Secara keseluruhan sistem *monitoring* yang diinginkan yaitu *monitoring* dapat dilakukan secara komputerisasi. Komputerisasi disini yaitu dengan kata lain menggunakan tampilan *interface*, berupa tampilan nilai arus pada beban dan kondisi lampu menggunakan sensor arus ACS 712. *Ouput* dari sensor arus tadi kemudian dibaca oleh *microcontroller* ATmega 328 menggunakan pembacaan *Analog to Digital Converter (ADC)*. Setelah pembacaan *ADC*, hasil dari pembacaan *ADC* selanjutnya dikirimkan menuju *microcontroller* ATmega 328 dan diolah sehingga dapat ditampilkan pada *personal computer* secara digital. Di Bandara Juanda, lampu *sequence flashing* yang digunakan memiliki daya sebesar 120 W dengan tegangan kerja sebesar 220 VAC. Itu artinya, untuk satu lampu *sequence flashing*, nilai arusnya sekitar 0,68 Ampere.

Apabila hasil pembacaan nilai arus pada setiap lampu *sequence flashing* tidak sesuai dengan nilai tadi, maka teknisi dapat secara langsung memastikan lampu mana yang tidak beroperasi dengan baik dan dapat menggantinya dengan lampu *sequence flashing* yang baru. Sensor arus ACS 712 yang digunakan memiliki kapasitas arus maksimal sebesar 5 A, hal ini telah memenuhi kriteria dari arus minimal yang dilewati yaitu sebesar 0,68 A untuk setiap lampu *sequence flashing*. Pada kenyataannya, jarak antara lampu *sequence flashing* dengan pusat kontrol sejauh 4 km.



Gambar 4. Flow Chart Sistem

## HASIL DAN PEMBAHASAN

### Perangkat Lunak dan Aplikasi Pemrograman Program Perangkat Lunak Arduino Uno

Berikut adalah bahasa pemrograman *arduino* uno keseluruhan yang digunakan penulis dalam pembuatan alat tersebut :

```

arduino - sensor.ino | Arduino 1.8.1
File Edit Sketch Tools Help

arduino karakter sensor

double read_ arus(int pin) {
  int rVal = 0;
  int sampleDuration = 100;
  int sampleCount = 0;
  unsigned long rSquaredSum = 0;
  int rZero = 511;
  uint32_t startTime = millis();
  while((millis()-startTime) < sampleDuration)
  {
    rVal = analogRead(pin) - rZero;
    rSquaredSum += rVal * rVal;
    sampleCount++;
  }

  double voltRMS = 5.0 * sqrt(rSquaredSum / sampleCount) / 1024.0;
  double ampsRMS = voltRMS * 18.5;
  ampsRMS=ampsRMS-0.45;
  if (ampsRMS<0){ampsRMS=0;}
  return ampsRMS;
}
  
```

Gambar 5. Bahasa Pemrograman ACS 712

```

arduino - karakter.ino | Arduino 1.8.1
File Edit Sketch Tools Help

arduino karakter sensor

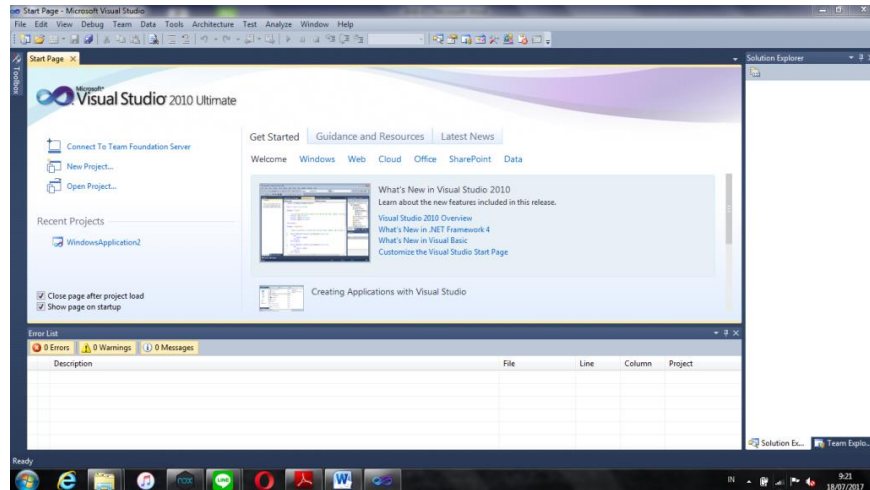
String karakter (double datax) {
  int a= String(datax).length();
  String sendd = String(datax);
  if(a<5){
    sendd="0"+sendd;
  }
  return sendd;
}

String karaktera (double datax) {
  int a= String(datax).length();
  String sendd = String(datax);
  sendd="0000"+sendd;
  return sendd;
}
  
```

Gambar 6. Bahasa Pemrograman *Flashing Lamp*

## Interface Perangkat Lunak Visual Basic

Pada pengujian ini dapat dilihat bahwa tampilan *software* dapat berjalan baik. Jika tanda koneksi pada pojok kiri atas sudah memperlihatkan tanda *connected* dan arus yang muncul dalam detail properti sudah membaca nilai arus di tiap-tiap beban berarti alat berjalan dengan baik.



Gambar 7. Tampilan Utama Software Visual Basic



Gambar 8. Pengujian Tampilan Visual Basic

Berikut adalah beberapa bahasa pemrograman dari *softwarevisual basic* yang digunakan penulis.

```
App.config x Class1.vb main form.vb [Design] login.vb [Design] graph2.vb [Design]
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />
  </startup>
</configuration>
```

Gambar 9. Bahasa Pemrograman untuk Visual Basic

```

App.config Class1.vb X main form.vb [Design] login.vb [Design] graph2.vb [Design] current graphic.vb [Design] WindowsApplication2*
(General) (Declarations)
Public Class Class1
    Dim string_error As String
    Dim string_blank As String
    Dim string_false As String

    Public Sub instruksi(data_string As String)
        string_blank = "Insert Each data before you can go to next step"
        string_false = "you have done inserting data but your data is wrong, correct all your data before next step"
        string_error = "file error please correct your instruction before next step"
    End Sub
End Class

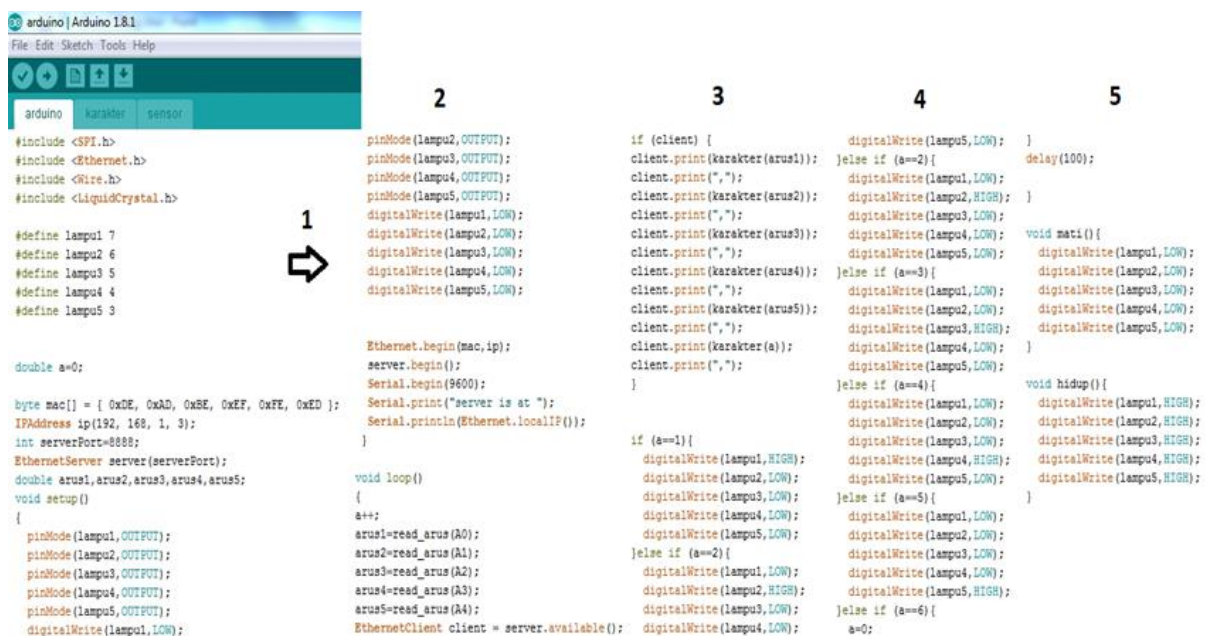
```

Gambar 10. Bahasa Pemrograman atau coding Visual Basic untuk Indikator Lampu

### Sistem Alat Keseluruhan

Dari pengujian di tiap-tiap rangkaian komponen dan pengujian *software* tersebut diatas terbentuklah suatu rancangan alat monitoring kegagalan *sequence flashing light* dengan hasil pengujian sebagai berikut :

Memasukkan bahasa pemrograman atau coding seluruh sistem pada aplikasi atau *software arduino*.



```

arduino | Arduino 1.8.1
File Edit Sketch Tools Help
arduino karakter sensor
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include <LiquidCrystal.h>

#define lampu1 7
#define lampu2 6
#define lampu3 5
#define lampu4 4
#define lampu5 3

double a=0;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 3);
int serverPort=8080;
EthernetServer server(serverPort);
double arus1, arus2, arus3, arus4, arus5;
void setup()
{
    pinMode(lampu1, OUTPUT);
    pinMode(lampu2, OUTPUT);
    pinMode(lampu3, OUTPUT);
    pinMode(lampu4, OUTPUT);
    pinMode(lampu5, OUTPUT);
    digitalWrite(lampu1, LOW);

    pinMode(lampu2, OUTPUT);
    pinMode(lampu3, OUTPUT);
    pinMode(lampu4, OUTPUT);
    pinMode(lampu5, OUTPUT);
    digitalWrite(lampu2, LOW);
    digitalWrite(lampu3, LOW);
    digitalWrite(lampu4, LOW);
    digitalWrite(lampu5, LOW);

    Ethernet.begin(mac, ip);
    server.begin();
    Serial.begin(9600);
    Serial.print("server is at ");
    Serial.println(Ethernet.localIP());
}

void loop()
{
    ++;
    arus1=read_arus(A0);
    arus2=read_arus(A1);
    arus3=read_arus(A2);
    arus4=read_arus(A3);
    arus5=read_arus(A4);
    EthernetClient client = server.available();

    if (client) {
        client.print(karakter(arus1));
        client.print(", ");
        client.print(karakter(arus2));
        client.print(", ");
        client.print(karakter(arus3));
        client.print(", ");
        client.print(karakter(arus4));
        client.print(", ");
        client.print(karakter(arus5));
        client.print(", ");
        client.print(karakter(a));
        client.print(", ");
    }

    if (a==1){
        digitalWrite(lampu1, HIGH);
        digitalWrite(lampu2, LOW);
        digitalWrite(lampu3, LOW);
        digitalWrite(lampu4, LOW);
        digitalWrite(lampu5, LOW);
    }
    else if (a==2){
        digitalWrite(lampu1, LOW);
        digitalWrite(lampu2, HIGH);
        digitalWrite(lampu3, LOW);
        digitalWrite(lampu4, LOW);
        digitalWrite(lampu5, LOW);
    }
    else if (a==3){
        digitalWrite(lampu1, LOW);
        digitalWrite(lampu2, LOW);
        digitalWrite(lampu3, HIGH);
        digitalWrite(lampu4, LOW);
        digitalWrite(lampu5, LOW);
    }
    else if (a==4){
        digitalWrite(lampu1, LOW);
        digitalWrite(lampu2, LOW);
        digitalWrite(lampu3, LOW);
        digitalWrite(lampu4, HIGH);
        digitalWrite(lampu5, LOW);
    }
    else if (a==5){
        digitalWrite(lampu1, LOW);
        digitalWrite(lampu2, LOW);
        digitalWrite(lampu3, LOW);
        digitalWrite(lampu4, LOW);
        digitalWrite(lampu5, HIGH);
    }
    else if (a==6){
        a=0;
    }
}

void mati() {
    digitalWrite(lampu1, LOW);
    digitalWrite(lampu2, LOW);
    digitalWrite(lampu3, LOW);
    digitalWrite(lampu4, LOW);
    digitalWrite(lampu5, LOW);
}

void hidup() {
    digitalWrite(lampu1, HIGH);
    digitalWrite(lampu2, HIGH);
    digitalWrite(lampu3, HIGH);
    digitalWrite(lampu4, HIGH);
    digitalWrite(lampu5, HIGH);
}
}

```

Gambar 11. Bahasa Pemrograman SoftwareArduino untuk Sistem Keseluruhan

### Kesimpulan

Dengan melakukan perancangan monitoring lampu *sequence flashing* di bandara Juanda, penulis memiliki beberapa kesimpulan yaitu :

1. Kita dapat memonitoring lampu *sequence flashing* di bandara Juanda secara *real time* dan mengetahui secara pasti lampu manakah yang tidak berfungsi dengan baik tanpa harus melakukan kontrol secara langsung ke tempat lampu *sequence flashing* berada.
2. Monitoring lampu *sequence flashing* juga dapat dilakukan dengan menggunakan visualisasi dari *software visual basic* yang langsung dapat menampilkan nilai arus pada masing-masing lampu.
3. Informasi data mengenai berfungsi atau tidaknya lampu *sequence flashing* dapat dikirim dengan sangat cepat melalui komunikasi *fiber optic* meskipun dengan jarak 4 kilometer.
4. Dapat dengan mudah mengetahui kerusakan pada lampu *sequence flashing* tanpa harus turun langsung kelapangan untuk mengecek di tiap-tiap lampu.

## DAFTAR PUSTAKA

Dalam *Visual Aids Handbook, CAP 637*.

Dipetik 2 14, 2017, dari Ethernet Module System: <http://support@innovativeelectronics.com>

Dipetik 2 13, 2017, dari [www.arduino.cc](http://www.arduino.cc): <http://www.arduino.cc>

*Datasheet.Atmega328/P*. (2016).

*Datasheet.ACS712*.

Aerodrome. *Manual of Standard CASR 139 Volume 1* .

Aerodrome. *ICAO ANNEX 14 Volume 1*.

Bejo, A. (2008). *Rahasia Kemudahan Bahasa*. Yogyakarta: Graha Ilmu.

Blocher, R. (2004). *Dasar Elektronika*. Yogyakarta: Andi.

Hantoro, G. D. (2010). *Fiber Optic*. Yogyakarta: Graha Ilmu.

Honeywell. Dalam *Manual Of Book ASL 40 Approach Sequence Flash Light with SFU 40 Pulse Generator*.

Ibrahim, K. (1996). *Prinsip Dasar Elektronika*. Jakarta: PT. Elex Media Komputindo.

Kadir, A. (2012). *Panduan Praktis Mempelajari Aplikasi Microcontroller*. Yogyakarta: Andi.

Kuncara, Purba. (2013). *Fiber Optics Technicians*. Bandung: Informatika.

Marta Dinata, Yuwono. (2016). *Arduino Itu Pintar*. Surabaya: Gramedia.

Oliviero, Andrew and Woodward, Bill. (2009). *Cabling*. Sybex.

Winoto, Ardi. (2008). *Mikrokontroler AVR ATmega dan Pemrogramannya*. Bandung: Informatika.