

PROTOTYPE SYSTEM PERINGATAN PENUMPANG KERETA API BERBASIS MIKROKONTROLER ARDUINO UNO ATMEGA328

Dimas Adi Prawira¹, E-mail: dimas@api.ac.id
Bahrul Ilmi Darmawan², E-mail: bahrul.tmp1504@alumni.api.ac.id
Fadli Irnanda Saputra³, E-mail: fadli.tmp1507@alumni.api.ac.id

^{1,2,3}Teknik Mekanika Perkeretaapian, Akademi Perkeretaapian Indonesia Madiun

ABSTRAK

Pelayanan dan kenyamanan merupakan faktor penting dalam menjalankan moda transportasi terutama perkeretaapian. Seiring berkembangnya perkeretaapian di Indonesia dibutuhkan peningkatan dalam hal pelayanan dan kenyamanan. Sering terjadi penumpang yang terlewat stasiun tujuan masih menjadi masalah yang harus diselesaikan dalam pelayanan dan kenyamanan fasilitas kereta api. Sistem peringatan dini yang masih *universal* menggunakan *speaker* masih kurang memberikan pelayanan yang maksimal. Dari hasil survei yang dilakukan masih banyak calon penumpang yang khawatir terlewat stasiun tujuannya. Pada penelitian ini dibuat sebuah alat yaitu prototipe sistem peringatan dini dengan penggetar yang dipasang pada kursi kereta api saat penumpang akan tiba di stasiun tujuan. Alat ini mampu memberikan peringatan dini secara individu, sehingga orang yang akan tiba di stasiun tujuan tidak terlewatkan. Prototipe ini bekerja menggunakan mikrokontroler Arduino Uno Atmega328 sebagai sistem kontrolnya. Mesin penggetar dengan besar getaran 53 Hz. Setelah dilakukan uji coba kepada 43 orang responden, rata-rata orang terbangun setelah diberikan getaran sebesar 53 Hz adalah 4 detik. *Keypad* sebagai *input* data kode stasiun tujuan penumpang. Lampu jika penumpang tidak terbangun menyala selama 5 menit. Hasil penelitian ini mampu membantu sistem kerja peringatan dini melalui *speaker* lebih maksimal.

Kata kunci : Pembuatan prototipe, sistem peringatan dini, penggetar

ABSTRACT

Service and convenience are important factors in carrying out transportation modes, especially railways. As the development of railways in Indonesia requires improvement in terms of service and comfort. It often happens that passengers who miss the destination station are still a problem that must be solved in the service and comfort of railroad facilities. Early warning systems that are still universal using speakers are still lacking in providing maximum service. From the results of the survey conducted there were still many prospective passengers who were worried about being missed the destination station. In this study, a tool was made, namely a prototype of an early warning system with a vibrator installed on the train seat when the passenger will arrive at the destination station. This tool is able to provide early warning individually so that people who will arrive at the destination station are not overlooked. This prototype works using the Arduino Uno Atmega328 microcontroller as its control system. Vibrating machine with a large vibration of 53 Hz. After testing the 43 respondents, the average person awakened after being given a vibration of 53 Hz is 4 seconds. The keypad as input data for the passenger destination station code. Lights if the passenger is not awakened lights up for 5 minutes. The results of this study are able to help the working system of early warning through the speaker more leverage.

Keywords: Making prototypes, early warning systems, vibrators

1. Pendahuluan

Transportasi kereta api merupakan transportasi masal yang saat ini sedang dalam masa perkembangan. Perkembangan perkeretaapian di Indonesia sangatlah pesat. Perkembangan itu sendiri harus diiringi dengan peningkatan fasilitas pelayanan agar minat masyarakat untuk menggunakan jasa transportasi bertambah.

Fasilitas pelayanan yang ada di kereta saat ini adalah toilet yang bersih, semua penumpang mendapatkan tempat duduk, AC, TV, lampu penerangan, stop kontak pada setiap kursi, kereta makan, selimut, bantal, majalah kereta, sistem peringatan dini berupa atensi melalui *speaker*.

Pada sistem peringatan dini yang ada di kereta masih bersifat umum. Peringatan dini melalui *speaker* ini masih memiliki kekurangan yakni masih terjadi penumpang yang terlewatkan stasiun tujuannya karena tertidur pulas. Hasil survei yang sudah dilakukan juga menunjukkan bahwa masih banyak penumpang yang khawatir terlewat stasiun tujuannya dengan sistem peringatan dini melalui *speaker*.

Untuk mengatasi masalah yang terjadi dilakukan penelitian yakni pembuatan prototipe sistem peringatan dini dengan penggetar yang dipasang pada kursi kereta api saat penumpang akan tiba di stasiun tujuan. Prototipe ini bekerja secara otomatis dengan cara memasukkan kode stasiun tujuan pada keypad yang tersedia disisi kiri kursi penumpang. Setelah kode dimasukkan, data akan tersimpan di mikrokontroler Arduino Uno Atmega328. Pada saat stasiun yang dituju sudah terdeteksi oleh RFID reader, selanjutnya Arduino akan mengirim sinyal untuk menggetarkan kursi selama 4 detik. Setelah 4 detik bergetar, penggetar akan mati kemudian lampu indikator pada bagian atas sandaran punggung akan menyala selama 5 menit. Lampu ini dimaksudkan bagi penumpang yang tidak merespon getaran yang diberikan agar segera dibangunkan

secara manual oleh petugas yang sedang bertugas.

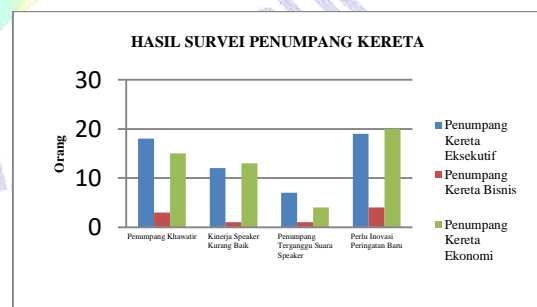
2. Metode Penelitian

Dilakukan eksperimen dengan membuat prototipe sistem peringatan dini dengan penggetar yang dipasang pada kursi kereta api saat penumpang akan tiba di stasiun tujuan. Dilakukan dengan cara mendesain ulang atau modifikasi kursi yang sudah ada, kursi yang digunakan adalah kursi kereta Bangladesh *single seat*. Membuat pemrograman sistem peringatan berbasis Arduino Uno Atmega328 sebagai mikrokontroler.

Sebelum alat dibuat dilakukan survei untuk mengetahui apakah penumpang sudah cukup dengan peringatan dini melalui *speaker*. Hasil survei menunjukkan bahwa penumpang masih merasa khawatir terlewatkan stasiun tujuan jika hanya menggunakan peringatan melalui *speaker*.

2.1. Hasil Survei Penumpang

Survei dilakukan dengan mengajukan 9 pertanyaan dengan 4 pertanyaan utama. Survei dilakukan langsung kepada penumpang kereta yang sedang melakukan perjalanan dan yang berada di stasiun agar survei yang dilakukan tepat sasaran. Untuk sampel yang diambil adalah 50 responden. Jumlah ini diambil mengacu pada jumlah maksimal penumpang yang dapat ditampung dalam satu kereta eksekutif. Berikut merupakan hasil survei yang sudah dilakukan.



Gambar 1. Grafik Hasil Survei Penumpang

2.2. Desain Awal Kursi Prototipe

Desain awal dari prototipe ini merupakan desain dari kursi kereta Bangladesh produksi PT. Industri Kereta Api. Kursi yang digunakan merupakan kursi tipe *single seat* dan *reclining*, sehingga dapat diatur untuk sudut sandaran duduknya. Spesifikasi kursi ini terdapat sandaran tangan di kedua sisi kanan dan kiri. *Leg rest* untuk penumpang yang berada di belakangnya. Berikut merupakan gambar kursi yang diproduksi oleh PT. Industri Kereta Api (Persero).



Gambar 2. Desain Awal Kursi

2.3. Pemrograman Prototipe

a. Sketch atau Program RFID dengan LED

Program RFID dengan LED dibuat sebagai pembelajaran sebelum pembuatan pemrograman untuk prototipe. Dalam program ini nantinya akan menggunakan RFID dan LED. Sistem kerja dari program ini akan membuat LED nyala jika RFID reader membaca RFID tag yang sesuai dengan yang ditentukan. Program ini dibuat dengan mengambil program dari library RFID kemudian di modifikasi

untuk menghasilkan sistem kerja yang diinginkan.

```

Simulate <SDI >
Simulate <MFR622>

#define RST_PIN 8 // Configurable, see typical pin layout above
#define SS_PIN 10 // Configurable, see typical pin layout above
#define LED_R 8
#define LED_G 7
MFR622 mfr622(SS_PIN, RST_PIN); // Create MFR622 instance

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  SPI.begin(); // Init SPI bus
  mfr622.PICC_Init(); // Init MFR622
  pinMode(LED_R, OUTPUT);
  pinMode(LED_G, OUTPUT);
  Serial.println("Put your card to the reader. ...");
  Serial.println();
}

void loop()
{
  // Wait for see card
  if ( ! mfr622.PICC_ReadCardSerial() )
  {
    return;
  }
  // Select one of the cards
  if ( ! mfr622.PICC_ReadSerial() )
  {
    return;
  }

```

Gambar 3. Sketch atau Pemrograman RFID

```

Serial.print("UID tag:");
String content= "";
byte letter;
for (byte i = 0; i < mfr622.uid.size; i++)
{
  Serial.print(mfr622.uid.uidByte[i] < 0x10 ? "0" : "");
  Serial.print(mfr622.uid.uidByte[i], HEX);
  content.concat(String(mfr622.uid.uidByte[i] < 0x10 ? " " : ""));
  content.concat(String(mfr622.uid.uidByte[i], HEX));
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
if (content.substring(1) == "30 A1 C8 8A")
{
  Serial.println("Authorized access");
  Serial.println();
  digitalWrite(LED_R, HIGH);
  delay(3000);
  digitalWrite(LED_R, LOW);
}
else
{
  Serial.println(" Access denied");
  digitalWrite(LED_R, HIGH);
  delay(1000);
  digitalWrite(LED_R, LOW);
}

```

Gambar 4. Sketch atau Pemrograman LED

b. Sketch atau Program Keypad, LED, dan Buzzer

Pemrograman ini dibuat untuk pelatihan pemrograman keypad sebelum dibuat pemrograman untuk prototipe. Pemrograman ini menggunakan komponen keypad, LED dan buzzer. Program nantinya akan menyalakan LED hijau dan buzzer satu kali jika inputan dari keypad sesuai dengan inputan yang sudah ditentukan dan jika inputan keypad salah maka akan mengaktif LED merah dan

buzzer dua kali. Program dibuat dengan menggabungkan program dari library *keypad* dan *password*. Setelah mendapatkan dua library utama, selanjutnya menggabungkan kedua library tersebut sehingga menghasilkan program sesuai dengan sistem kerja yang diinginkan.

```
#include <Keypad.h>
#include <Password.h>

byte currentLength = 0;

Password password = Password("1234");//Password bisa di ubah ulang

const byte ROWS= 4; //baris pada keypad
const byte COLS= 4; //kolom pada keypad

const int red = 13; //pin yang digunakan
const int green = 12;
const int buzzer = 11;

char keys[ROWS][COLS] =
{
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPin[ROWS] = {0, 7, 6, 5};
byte colPin[COLS] = {4, 3, 2, 1};

Keypad keypad = Keypad(makeKeymap(keys), rowPin, colPin, ROWS, COLS);

void setup()
{
  Serial.begin(9600);
  Serial.println("AT-MD PRODUCTION");
  Serial.println("Masukan Password");
}

void loop()
{
  Serial.println("Masukan Password");
  Serial.println("Pass: ");
  keypad.getKey();

  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(buzzer, OUTPUT);
  keypad.addEventListener(keypadEvent);
}

void keypadEvent(KeypadEvent eKey)
{
  switch (keypad.getState()) {
    case PRESSED:
      Serial.println(eKey);
      switch (eKey) {
        case '*': checkPassword(); currentLength=0; break;
        case '#': password.reset(); currentLength=0; break;
        default: // password.append(eKey);
          password << eKey;
          currentLength++;

          Serial.println("Pass: ");
          for (byte i=0; i<currentLength; i++){
            Serial.print('*');
          }
      }
  }
}
```

Gambar 5. *Sketch* atau Pemrograman Keypad

```
Serial.println("Pass: ");
keypad.getKey();

pinMode(red, OUTPUT);
pinMode(green, OUTPUT);
pinMode(buzzer, OUTPUT);
keypad.addEventListener(keypadEvent);
}

void loop()
{
  Serial.println("Masukan Password");
  Serial.println("Pass: ");
  keypad.getKey();
}

void checkPassword(){
  if(password.evaluate()){
    digitalWrite(green, HIGH);
    digitalWrite(buzzer, HIGH);
    Serial.println("Berhasil");
  }
  else {
    digitalWrite(red, HIGH);
    digitalWrite(buzzer, HIGH);
    Serial.println("Salah coba lagi");
    digitalWrite(red, LOW);
    digitalWrite(buzzer, LOW);
  }
}
```

Gambar 6. *Sketch* atau Pemrograman Buzzer

```
void keypadEvent(KeypadEvent eKey)
{
  switch (keypad.getState()) {
    case PRESSED:
      Serial.println(eKey);
      switch (eKey) {
        case '*': checkPassword(); currentLength=0; break;
        case '#': password.reset(); currentLength=0; break;
        default: // password.append(eKey);
          password << eKey;
          currentLength++;

          Serial.println("Pass: ");
          for (byte i=0; i<currentLength; i++){
            Serial.print('*');
          }
      }
  }
}
```

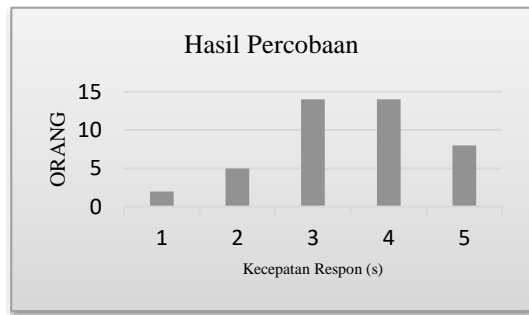
Gambar 7. *Sketch* atau Pemrograman LED

2.4. Hasil Percobaan Alat

Setelah desain dan pembuatan program dilakukan. Selanjutnya adalah melakukan percobaan. Percobaan dilakukan terhadap beberapa responden. Percobaan ini dimaksudkan dengan tujuan untuk mengetahui seberapa cepat manusia merespon getaran sebesar 53 Hz. Dari hasil percobaan yang dilakukan ditemukan jumlah rata-rata yaitu 4 detik. Berikut merupakan tabel hasil percobaan kecepatan respon manusia terhadap getaran sebesar 53 Hz.

Orang ke -	Respon saat alat bergetar ketika sedang tertidur (s)
1	1
2	3
3	3
4	4
5	5
6	4
7	1
8	3
9	3
10	3

Tabel 1. Data Percobaan Kecepatan Respon Manusia



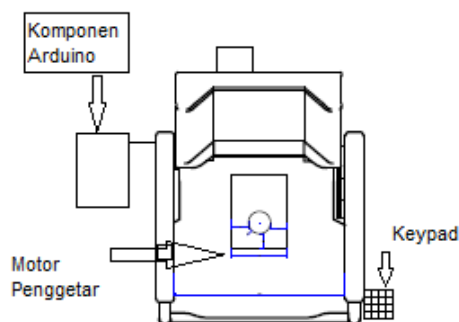
Gambar 8. Grafik Hasil Percobaan Alat

3. Hasil dan Pembahasan

Penelitian ini menghasilkan sebuah prototipe sistem peringatan dini dengan penggetar yang dipasang pada kursi kereta api saat penumpang akan tiba di stasiun tujuan.

3.1. Modifikasi Desain Kursi

Modifikasi dari Prototipe ini adalah pada bagian dudukan kursi yang ditanamkan komponen utama yaitu penggetar, pada bagian sandaran tangan sebelah kiri adalah *keypad*, pada sandaran punggung bagian atas ada penambahan lampu indikator. Berikut merupakan gambar *layout* prototipe.



Gambar 9. Layout Kursi Setelah Modifikasi

Prototipe ini menggunakan getaran berfrekuensi 53 Hz. Getaran ini dipilih

karena mampu memberikan getaran kejut kepada manusia yang sedang tertidur. Dan berdasarkan hasil percobaan alat ini diatur dengan lama getaran 4 detik.

3.2. Standar Operasi Prosedur (SOP)

Standar Operasi Prosedur meliputi alur pembuatan alat, penggunaan alat, dan perawatan alat. Berikut merupakan langkah penggunaan alat.

A. Langkah Persiapan

- 1) Sambungkan kabel penghubung bertuliskan "*keypad*" pada kotak akrilik dengan kabel bertuliskan sama yang terpasang di kursi.
- 2) Sambungkan kabel penghujung bertuliskan "*lampu*" pada kotak akrilik dengan kabel bertuliskan sama yang terpasang di kursi.
- 3) Sambungkan kabel *power supply* untuk dinamo penggetar yang berada dibagian belakang kursi pada kotak akrilik.
- 4) Sambungkan dua kabel *power supply* untuk arduino dan dinamo penggetar yang berada pada kotak akrilik.

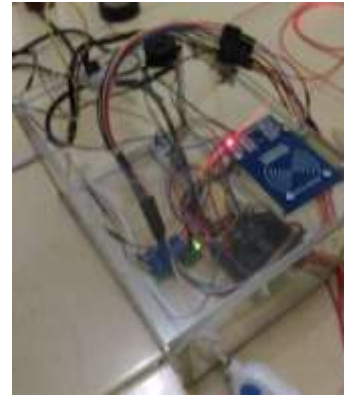
B. Langkah Penggunaan

- 1) Masukkan salah satu kode yang tertera pada kartu yang tersedia.
- 2) Setelah kode dimasukkan tekan tanda bintang (*) untuk mengkonfirmasi kode, jika kode yang dimasukkan salah maka tekan tanda pagar (#) untuk me-reset inputan kode.
- 3) Setelah itu tempelkan kartu ke RFID yang berada pada kotak akrilik.
- 4) Jika kartu yang ditempelkan pada RFID sesuai dengan kode yang di-input maka *motor vibrator* akan bergetar selama 5 detik dan dilanjutkan lampu LED akan menyala selama 5 menit.

- 5) Pengulangan proses dapat langsung dilakukan dengan menekan tombol *reset*.

C. Langkah Penyimpanan

- 1) Lepaskan dua kabel *power supply* pada stop kontak yang terhubung dengan kotak akrilik.
- 2) Lepaskan kabel *power supply* yang terhubung pada kursi yang terhubung dengan kotak akrilik.
- 3) Lepaskan kabel penghubung bertuliskan "*keypad*" pada kotak akrilik dengan kabel bertuliskan sama yang terpasang di kursi dengan hati-hati.
- 4) Sambungkan kabel penghujung bertuliskan "*lampu*" pada kotak akrilik dengan kabel bertuliskan sama yang terpasang di kursi dengan hati-hati. Simpan prototipe di tempat yang aman.



(a)



(b)

Langkah perawatan adalah untuk menjaga prototipe supaya dalam keadaan prima ketika digunakan. Berikut merupakan langkah langkah perawatan.

- 1) Membersihkan kabel, kursi dan akrilik dengan menggunakan kanebo (basah pada pembersihan kursi).
- 2) Cek sambungan kabel apakah ada kabel yang terlepas atau tidak.
- 3) Cek masing-masing kabel terputus atau tidak menggunakan voltmeter.
- 4) Uji coba kerja alat prototipe dengan mencoba semua kode pada setiap kartu.



(c)

3.3. Proses Pembuatan Prototipe

Proses pembuatan prototipe meliputi pembuatan *casing* untuk komponen mikrokontroller, *casing* untuk *motor vibrator*, merangkai komponen prototipe, dan penyelesaian (*finishing*).



(d)

Gambar 10. (a) Casing untuk Komponen Mikrokontroller, (b) Casing untuk Motor Vibrator, (c) Perangkaian Prototipe, dan (d) Pemasangan Komponen pada Kursi

3.4. Pemrograman Prototipe

Pembuatan program sistem kerja prototipe ini disesuaikan dengan logika untuk menyeleksi *inputan* data dari *keypad* dan RFID. Langkah awal dalam pembuatan program prototipe ini yaitu menentukan komponen apa saja yang digunakan. Setelah menentukan komponen yang akan digunakan, selanjutnya yaitu menentukan pin yang digunakan untuk setiap komponen pada arduino. Mengetahui komponen yang akan digunakan merupakan hal yang sangat penting karena sebelum membuat program pada aplikasi Arduino, perpustakaan (*library*) setiap komponen harus tersimpan pada Aplikasi Arduino. Jika hal itu tidak dilakukan maka akan terjadi masalah (*error*) pada proses *upload*. Untuk komponen yang kami gunakan meliputi RFID, *keypad*, buzzer, lampu LED, motor vibrator, dan relay. Pin yang digunakan masing-masing komponen meliputi:

1. RFID menggunakan 2 pin yaitu pin 10 digunakan untuk SDAPIN dan pin 9 untuk RESETPIN

2. Lampu LED menggunakan pin A5
3. Buzzer menggunakan pin A4
4. Relay dan motor vibrator menggunakan pin A3
5. *Keypad* dengan kolom dan baris adalah 4x4 menggunakan pin A1, 2, 3, 4, 5, 6, 7, 8.

```
#include <Password.h>
#include <Keypad.h>
#include <MFRC522.h>
#include <SPI.h>

#define SDAPIN 10
#define RESETPIN 9
#define LED A5
#define Buzzer A4
#define Vibrator A3

const byte ROWS = 4; // Four rows
const byte COLS = 4; // columns
// Define the Keypad
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = { 8,7,6,5 };//
byte colPins[COLS] = { 4,3,2,A1 };
```

Gambar 11. Bahasa Pemrograman untuk *Input Library* dan Pin

Setelah selesai meng-*inputlibrary* selanjutnya yaitu menentukan password yang digunakan untuk menentukan *inputan* data dari *keypad* terdaftar atau tidak. Dalam pengaplikasiannya di prototipe password ini digunakan untuk menentukan kode boking yang terdaftar.

```
Password password1 = Password( "23A1" );
Password password2 = Password( "1B34" );
```

Gambar 12. Bahasa Pemrograman *Input Password*

Setelah itu tentukan nama penyimpanan setiap logika yang dikerjakan dan tentukan juga kondisi

logika yang tercantum pada void loop. Pada program prototipe yang di buat, untuk penyimpanan membuat beberapa nama diantaranya:

1. “GoodTagSerialNumber” dan “GoodTagSerialNumber1” nama penyimpanan ini digunakan untuk menyimpan data berupa nomor ID RFID yang sudah ditentukan. Untuk “GoodTagSerialNumber” menyimpan nomor ID RFID 30 A1 CD A3 dan “GoodTagSerialNumber1” menyimpan nomor ID RFID A2 78 AC 79.
2. “ReadTag” nama penyimpanan ini digunakan untuk menyimpan informasi yang digunakan untuk membaca tag.
3. “TagData” nama penyimpanan yang digunakan untuk menyimpan perintah bahwa batas grup tag yang di izinkan yaitu 4 grup.
4. “TagSerialNumber” digunakan untuk menyimpan semua data nomor ID yang teridentifikasi oleh RFID.
5. “FoundTag” untuk menyimpan data bahwa RFID yang digunakan sudah sesuai dengan yang ditentukan.

Penentuan kondisi logika pada program arduino terdiri dari beberapa jenis dan yang digunakan pada program prototipe ini yaitu *int* dan *string*. Untuk *int* digunakan untuk menentukan kondisi logika dengan angka. Biasanya yang digunakan yaitu “1” dan “0”. Sedangkan untuk *string* digunakan untuk menentukan kondisi logika dengan menggunakan kata “TRUE” dan “FALSE”. Untuk kondisi logika yang digunakan pada program prototipe ini meliputi:

1. “PasswordTrue” digunakan untuk menentukan kondisi apakah data yang di masukan oleh keypad sesuai password yang sudah di tentukan.
2. “GoodTag” digunakan untuk menentukan kondisi apakah ID nomor yang teridentifikasi sudah sesuai dengan yang ditentukan.

3. “Reset” digunakan untuk menentukan kondisi akan di reset atau tidak.
4. “Pass1” digunakan untuk menentukan apakah Password 1 terdeteksi atau tidak.
5. “Pass2” digunakan untuk menentukan apakah Password 2 terdeteksi atau tidak.
6. “ID1” digunakan untuk menentuka apakah ID 1 terdeteksi atau tidak.
7. “ID2” digunakan untuk menentuka apakah ID 2 terdeteksi atau tidak.

```
byte GoodTagSerialNumber[4] = {0x30, 0xA1, 0xCD, 0xA3};
byte GoodTagSerialNumber1[4] = {0xA2, 0x78, 0xAC, 0x79};
byte ReadTag;
byte TagData[MAX_LEN];
byte TagSerialNumber[4];
byte FoundTag;
String PasswordTrue;
String GoodTag;

int Reset;
int Pass1;
int Pass2;
int ID1;
int ID2;
```

Gambar 13. Bahasa Pemrograman Nama Penyimpanan dan Kondisi Logika

Menentukan komponen yang digunakan sebagai keluaran (*output*). Untuk menentukan *output* pada program yang dibuat, kalimat harus ditulis pada bagian “void setup”. Pada program prototipe ini keluaran (*output*) terdiri dari 3 komponen yaitu LED, Buzzer , dan Vibrator.

```
pinMode(LED, OUTPUT);
pinMode(Buzzer, OUTPUT);
pinMode(Vibrator, OUTPUT);
}
```

Gambar 14. Bahasa Pemrograman untuk Menentukan Output

Selanjutnya yaitu memasuki pada bagian *void keypad* yang menjelaskan proses kerja dari keypad. Dalam program prototipe ini menggunakan keypad 4x4

yang terdiri dari 10 angka, 4 huruf, pagar dan bintang. *Keypad* ini digunakan sebagai *input* untuk kode boking penumpang. Kerja *keypad* ini nanti didukung dengan adanya buzzer sebagai informasi tambahan untuk penumpang. Dengan logika yang sudah di buat nantinya jika penumpang menekan *keypad* buzzer akan berbunyi selama 50 ms. Selain itu dengan logika yang dibuat, gambar bintang “*” pada *keypad* berfungsi sebagai tombol untuk memeriksa kode yang sudah di *input* sedangkan gambar pagar “#” pada *keypad* digunakan untuk mereset kode *inputan*. Untuk mengetahui kode *inputan* teridentifikasi atau tidak, logika yang dibuat akan membunyikan buzzer menjadi 2 jenis suara yang pertama buzzer akan berbunyi selama 150 ms sebanyak 2 kali jika kode *inputan* teridentifikasi dan kedua buzzer akan berbunyi selama 2 detik jika kode *inputan* tidak teridentifikasi.

```
void keypadEvent(KeypadEvent eKey){
  switch (keypad.getState()){
    case PRESSED:
      Serial.print("Pressed: ");
      Serial.println(eKey);
      digitalWrite(Buzzer, HIGH);
      delay(50);
      digitalWrite(Buzzer, LOW);
  }
  switch (eKey){
    case '*': checkPassword(); password2.reset(); password1.reset(); break;
    case '#': password2.reset(); password1.reset(); break;
    default: password2.append(eKey); password1.append(eKey);
  }
}
}
```

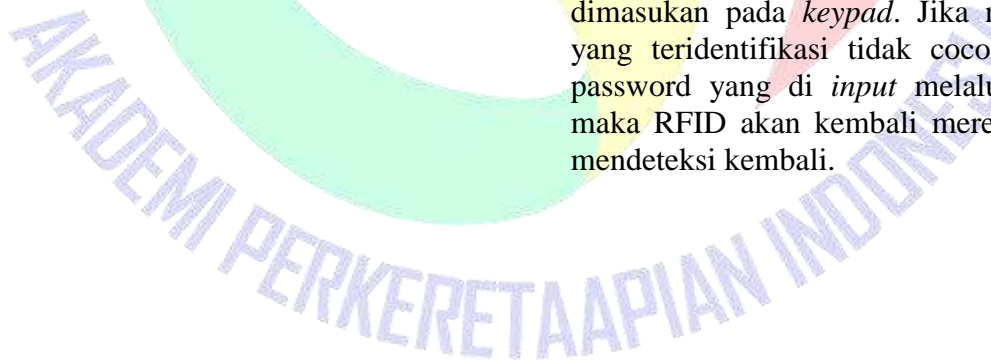
(a)

```
void checkPassword(){
  if (password2.evaluate()){
    Serial.println("Success");
    PasswordTrue="TRUE";
    Pass2 = 1;
    digitalWrite(Buzzer, HIGH);
    delay(150);
    digitalWrite(Buzzer, LOW);
    delay(150);
    digitalWrite(Buzzer, HIGH);
    delay(150);
    digitalWrite(Buzzer, LOW);
    //Add code to run if it works
  }
  if (password1.evaluate()){
    Serial.println("Success");
    PasswordTrue="TRUE";
    Pass1 = 1;
    digitalWrite(Buzzer, HIGH);
    delay(150);
    digitalWrite(Buzzer, LOW);
    delay(150);
    digitalWrite(Buzzer, HIGH);
    delay(150);
    digitalWrite(Buzzer, LOW);
  }
  if ( Pass1 != 1 && Pass2 != 1){
    digitalWrite(Buzzer, HIGH);
    delay(2000);
    digitalWrite(Buzzer, LOW);
  }
}
```

(b)

Gambar 15. (a) Bahasa Pemrograman untuk *Keypad*, dan (b) Pemrograman dengan Logika untuk *Check Password* dan Pengaturan Bunyi *Buzzer*

Selanjutnya yaitu memasuki bagian *void* RFID yang menjelaskan tentang proses kerja RFID. RFID bekerja dengan mendeteksi *tag* atau kartu yang memiliki frekuensi dengan besar yang dapat diterima oleh RFID. Hasil pendeteksian *tag* atau kartu nantinya akan disimpan dan di olah untuk dicocokkan dengan password yang dimasukan pada *keypad*. Jika nomor ID yang teridentifikasi tidak cocok dengan password yang di *input* melalui *keypad* maka RFID akan kembali mereset untuk mendeteksi kembali.



```

void setup() {
  Serial.begin(9600); // Variable used to configure your tag detected
  // Check to see if a Tag was detected
  // If yes, then the variable foundTag will become "true"
  foundTag = true; requestTag(001, 002); TagData;
}

// (foundTag == 01_001)
delay(1000);

// Get serial number value to properly send information from the Tag
foundTag = true; sendSerial(TagData);
Serial.print(TagSerialNumber, TagData, 0); // Print the Tag information on the TagSerialNumber variable

Serial.println("Tag Detected");
Serial.print("Serial Number: ");
// Now store it in ID 1 & ID 2, after 1 of loop we print serial number to serial monitor
Serial.print(TagSerialNumber[0], HEX);
Serial.print(", ");
}

Serial.println("");
Serial.println("");

// Check if detected Tag has the right Serial number we are looking for
// (ID1 == 01 & ID2 == 02)
if ((foundTagSerialNumber[0] == TagSerialNumber[0]) && (foundTagSerialNumber[1] == TagSerialNumber[1])) {
  Serial.println("IF not equal, then break out of the 'while' loop");
}

// If ID1 == 01 & ID2 == 02, if we want to do 7 loops then the Tag Serial number are matching
Serial.println("MATCH");
if (foundTagSerialNumber[0] == TagSerialNumber[0]) {
  // (foundTagSerialNumber[1] == TagSerialNumber[1])
  // ID2=0;
}

// (foundTag == "00000")
Serial.println("Found Password");
Serial.println("");
Serial.println("");
Serial.println("");

// (foundTag == "00000")
Serial.println("Found Password");
Serial.println("");
Serial.println("");
Serial.println("");
}

// (foundTag == "00000")
Serial.println("Found Password");
Serial.println("");
Serial.println("");
Serial.println("");
}
    
```

(a)

```

// (foundTag == "00000")
Serial.println("Found Password");
Serial.println("");
Serial.println("");
Serial.println("");
}

// (foundTag == "00000")
Serial.println("Found Password");
Serial.println("");
Serial.println("");
Serial.println("");
}
    
```

(b)

Gambar 16. (a) Bahasa Pemrograman untuk Kerja RFID, dan (b) Bahasa Pemrograman untuk Menyeleksi Nomor ID yang Terdeteksi dengan yang Sudah Ditetapkan

Setelah semua *inputan* di program dengan logika yang dibutuhkan maka selanjutnya menghubungkan kedua *inputan* tersebut agar menghasilkan *output* yang kita inginkan. Untuk menghubungkan kedua *inputan* tersebut maka akan dibuat logika pada bagian *void loop*. Bagian ini menjelaskan tentang proses dari kerja prototipe tersebut. Namun pada program ini, logika di sederhanakan sehingga menjadi 2 bagian yang sudah di jelaskan sebelumnya yaitu *void keypad* dan *void RFID*. Pada bagian *void loop* dibuat logika yang menghasilkan *output* akhir berupa getaran dan cahaya lampu. *Output* berupa getaran akan bekerja selama 15 detik dan cahaya lampu akan bekerja selama 5 menit.

```

void loop() {
  if (Reset == 1)
  {
    Pass1 = 0;
    Pass2 = 0;
    ID1 = 0;
    ID2 = 0;
    delay (50);
    Reset = 0;
  }
  keypad.getKey();

  if (PasswordTrue == "TRUE"){
    rfid ();
  }
  else{
    PasswordTrue == "FALSE";
  }
  if ((ID1 == 1) && (Pass1 == 1)){
    digitalWrite(Vibrator, HIGH);
    delay(15000);
    digitalWrite(Vibrator, LOW);
    digitalWrite(LED, HIGH);
    delay(300000);
    digitalWrite(LED, LOW);
    Reset = 1;
  }
  if ((ID2 == 1) && (Pass2 == 1)){
    digitalWrite(Vibrator, HIGH);
    delay(15000);
    digitalWrite(Vibrator, LOW);
    digitalWrite(LED, HIGH);
  }
}
    
```

Gambar 17. Bahasa Pemrograman pada Void Loop

4. Kesimpulan dan Saran

Berdasarkan penelitian yang sudah dilakukan dihasilkan sebuah prototipe pengingat penumpang sampai stasiun tujuan dengan penggetar pada tempat duduk penumpang. Prototipe ini diharapkan dapat diaplikasikan di kereta penumpang. Sehingga perlu adanya penelitian lebih lanjut berkaitan dengan perencanaan penerapan alat dipengoperasian kereta penumpang, serta penggunaan mikrokontroler yang dapat menampung lebih banyak data daripada Arduino.

DAFTAR PUSTAKA

Sugianto Ahmad Hadi. (2013). *Kajian Sistem Standar Operasional Prosedur (Sop)E-Procurement.*

- Tugas Akhir Jurusan Teknik Sipil.
Padang: Universitas Bung Hatta
- Irma Puspitasari.(2016).*Modifikasi Kursi Penumpang Kereta Api Ekonomi yang Ergonomis dengan Metode Ergonomic Function Deployment* (Skripsi).Jurusan Teknik Mesin Universitas Jember:Jember
- MENLH.1996. *Tentang Baku Tingkat Getaran*. Jakarta : Kementrian Lingkungan Hidup
- Novi Dwi Ira Suryani.2015.*Analisis Pengaruh Tingkat Kebisingan dan Getaran Kereta Api Terhadap Tekanan Darah Ibu Rumah Tangga di Pemukiman Pinggiran Rel Kereta Api Jalan Ambengan Surabaya* (Skripsi).Fakultas Kesehatan Masyarakat Universitas Airlangga:Surabaya
- Priscalia Denni Yantri.2017.*Getaran Seluruh Tubuh (Whole Body Vibration) dan Keluhan Nyeri Punggung Bawah pada Operator Alat Berat di Instansi Pemerintah Kabupaten Jember* (Skripsi). Fakultas Kesehatan Masyarakat Universitas Jember:Jember
- Chen Yangyi (2013). Vibration Motor.
- Manual Book *Keypad* 4×4, tentang Additional Board
- Manual Book Arduino uno, tentang overview Arduino uno
- Muhidin Asep. (2010). Pemrograman Bahasa C++. Bekasi. Zeyrank Offset.
- Lutfi Muhammad. (2017). Pintu Pagar Rumah Otomatis Dan Dilengkapi Sistem Keamanan (Kontrol Motor). Tugas Akhir Jurusan Teknik Elektro. Palembang: Politeknik Negeri Sriwijaya.
- Santoso Hari. (2015). *Panduan Praktis Arduino Untuk Pemula*. Trenggalek. Elang Sakti.
- Saputra Chandra Bayu KM. (2016). *Aplikasi Rfid Sebagai Identifikasi Pada Prototipe Pengatur Solenoid Valve Berbasis Arduino Uno (Atmega328)*. Tugas Akhir Jurusan Teknik Elektro. Palembang: Politeknik Negeri Sriwijaya.
- Sugianto Ahmad Hadi. (2013). *Kajian Sistem Standar Operasional Prosedur (Sop)E-Procurement*. Tugas Akhir Jurusan Teknik Sipil. Padang: Universitas Bung Hatta
- Turang Daniel Alexander O. (2015). *Pengembangan Sistem Relay Pengendalian Dan Penghematan Pemakaian Lampu Berbasis Mobile*. Seminar Nasional Informatika. Sekolah Tinggi Teknologi Bontang. ISSN: 1979-2328.
- <https://teknikelektronika.com/pengertian-relay-fungsi-relay>. Diakses pada tanggal Rabu, 16 Mei 2018 pukul 10.34 WIB
- <https://teknikelektronika.com/pengertian-piezoelectric-buzzer-cara-kerja-buzzer/> . Diakses pada tanggal Rabu, 16 Mei 2018 pukul 11.27 WIB
- https://id.wikipedia.org/wiki/Bahasa_pemrograman. Kamis, 17 Mei 2018 pukul 10.44 WIB
- <http://terminaltechno.blog.uns.ac.id/2011/03/13/rfid-radio-frequency-identification>. Diakses pada tanggal Senin, 4 Juni 2018 pukul 14.24 WIB
- <https://depokinstruments.com/>. Kamis, 26 April 2018 pukul 08.00 WIB