

BAB IV

ANALISA DAN PEMBAHASAN

Dalam penelitian yang telah dilakukan maka ada beberapa hal yang penulis akan bahas pada bagian ini yakni diantaranya:

4.1. Pengujian Power Supply

Pada rangkaian power supply terdiri dari accu sebagai sumber listrik, diode sebagai penyearah tegangan, IC regulator LM7805 sebagai penyetabil tegangan, dan kapasitor sebagai filter. Rangkaian power supply menghasilkan tegangan keluaran sebesar +5V seperti yang ditunjukkan pada tabel 4.1.

Tabel 4.1 Pengujian Power Supply

No	Input Voltage (Volt DC)	Output Voltage (Volt DC)
1	12	4,9

Sumber : Peneliti (2020)

4.2. Pengujian Jarak dan Pembacaan RFID

Pengujian ini bertujuan mengetahui kemampuan custom RFID reader membaca RFID tag antara custom RFID reader dengan RFID tag yang dibaca.



Gambar 4.1 Pengujian Kartu RFID

Sumber : Peneliti (2020)

Tabel 4.2 Hasil pembacaan dan jarak RFID tag

Pengujian Ke	Jarak (cm)	Yang dibaca	Hasil Pembacaan
1	0	62 DE ED 1A	Berhasil
2	1	9A 6D EF 29	Berhasil
3	2	51 E7 48 1A	Berhasil
4	3	72 60 C4 1A	Berhasil
5	4	69 54 GA 6A	Tidak Berhasil

Sumber : Peneliti (2020)

Hasil pengujian yang diberikan oleh Tabel 4.2 terlihat jarak maksimum pembacaan RFID tag pada RFID reader adalah 3 cm. Seluruh tag yang digunakan dapat terbaca oleh reader dan reader mampu membaca seluruh nomor seri yang terdapat di dalam setiap tag dan meneruskannya ke database.

4.3. Pengujian Transmit Data RFID Arduino ke Raspberry

4.3.1 Inisialisasi Pin dan Variable

Berikut adalah inisialisasi Pin dan Variable pada tugas akhir ini :

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
#define Sole_ON digitalWrite(3,HIGH)
#define Sole_OFF digitalWrite(3,LOW)
#define Buzz_ON digitalWrite(2,HIGH)
#define Buzz_OFF digitalWrite(2,LOW)
String cardID = "", idLogin = "", idReg = "", inString = "", IDcard = "", sendIDcard = "";
char inChar;
boolean registNewID = false;
MFRC522 mfrc522(SS_PIN, RST_PIN); // Instance of the class
```

Gambar 4.2 Program Inisialisasi PIN dan Variable

Sumber : Peneliti (2020)

4.3.2 Proses baca RFID dan Penentuan Jenis ID

Arduino akan membaca ID Card RFID setelah proses TAP dilakukan. Pada tugas akhir ini terdapat dua tipe RFID yaitu satu untuk akses pintu masuk dan kedua untuk membuka menu register

ID baru. Syntax dari pembacaan RFID menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.3 dibawah ini.

```

if(mfrc522.PICC_IsNewCardPresent())
{
  if(mfrc522.PICC_ReadCardSerial())
  {
    cardID = "";
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
      cardID.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
      cardID.concat(String(mfrc522.uid.uidByte[i], HEX));
    }
    cardID.toUpperCase(); delay(1000);
    IDcard = cardID.substring(1);
  }
}

```

Gambar 4.3. Syntax Pembacaan Data RFID
Sumber : Peneliti (2020)

Jika yang terdeteksi ID client maka akan langsung dikirim ke Raspberry melalui data serial untuk mengetahui apakah ID tersebut sudah terdaftar pada database atau belum. Format data yang dikirim adalah SR# ID RFID. Syntax yang terbaca adalah ID Client menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.4 dibawah ini.

```

if(IDcard != "BD D8 45 83") { Serial.println("SR#" + IDcard); }

```

Gambar 4.4. Jika RFID yang terbaca adalah ID Client
Sumber : Peneliti (2020)

Jika terdeteksi master card maka sistem akan mengaktifkan menu registrasi ID baru serta menaktifkan buzzer sebanyak tiga kali seperti yang ditunjukkan pada gambar 4.5 berikut ini.

```

else
{
  Buzz_ON; delay(500); Buzz_OFF; delay(80);
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Serial.println("Register New ID Card Menu");
  registNewID = true; delay(500);
}

```

Gambar 4.5. Jika RFID yang terbaca adalah MASTER Card
Sumber : Peneliti (2020)

4.3.3 Menu Registrasi ID Baru

Setelah masuk ke dalam menu pendaftaran ID Card baru, TAP ID Card baru kemudian ID number baru, dikirim ke Raspberry untuk disimpan ke database. Jika Master Card di TAP satu kali lagi maka akan keluar dari menu registerasi. Format data yang dikirim adalah R# ID RFID. Syntax dari menu registrasi ID baru menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.6 dibawah ini.

```

//----- Register New ID -----
if(registNewID)
{
  if(IDcard == "BD D8 45 83")
  {
    Buzz_ON; delay(80); Buzz_OFF; delay(80);
    Buzz_ON; delay(80); Buzz_OFF; delay(80);
    Serial.println("Close New ID Card Register Menu");
    registNewID = false; IDcard = ""; delay(500);
  }
  if(sendIDcard != IDcard && IDcard != "BD D8 45 83") (Serial.println("R#" + IDcard); sendIDcard = IDcard);
}

```

Gambar 4.6. Menu Register ID baru setelah master card terbaca
Sumber : Peneliti (2020)

4.3.4 Proses Pembacaan Data Serial Dari Raspberry

Arduino akan memindai atau mempartisi data tersebut. Terdapat dua perintah data yang dipartisi yaitu RR# untuk respon akses pintu masuk dan S# untuk respon ke menu registrasi. Syntax dari proses pembacaan data serial dari Raspberry menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.7 dibawah ini.

```

void serialEvent()
{
  while(Serial.available() > 0)
  {
    inChar = (char)Serial.read();
    inString += inChar;
    if(inChar == '\n')
    {
      int L_StrIn = inString.length();
      int hIdxPos = inString.indexOf("#");
      String hIdx = inString.substring(0, hIdxPos+1);

      if(hIdx == "RR#") { idLogin = inString.substring(hIdxPos+1, L_StrIn); }
      if(hIdx == "S#") { idReg = inString.substring(hIdxPos+1, L_StrIn); }
      inString = "";
    }
  }
}

```

Gambar 4.7. Pembacaan Data Serial Dari Raspberry
Sumber : Peneliti (2020)

4.3.5 Jenis Respond Data Serial Dari Raspberry

Respon dari Raspberry jika terbaca S#1 maka pendaftaran ID baru berhasil maka buzzer akan bunyi sekali, jika gagal maka buzzer bunyi dua kali dengan respon S#2. Syntax dari respond data serial jika pendaftaran ID baru berhasil atau gagal menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.8 dibawah ini.

```
//Register New ID Respond - S#1 = Success, S#2 = Failed
if(idReg.toInt() == 1)
{
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Serial.println("Registering New ID Number " + cardID.substring(1) + " Succeeded*");
  idReg = ""; sendIDcard = "";
}
if(idReg.toInt() == 2)
{
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Serial.println("Registering ID Number " + cardID.substring(1) + " failed or has been registered*");
  idReg = ""; sendIDcard = "";
}
}
```

Gambar 4.8. Respond dari Raspberry jika Pendaftaran ID baru
Sumber : Peneliti (2020)

Respon dari Raspberry jika terbaca RR#1 maka buzzer akan bunyi sekali dan solenoid akan tertutup selama 2 detik, jika yang terbaca RR#2 akses solenoid ditolak dan buzzer akan bunyi dua kali. Syntax dari respond akses pintu masuk yang diizinkan atau ditolak menggunakan library dari Arduino seperti yang ditunjukkan pada gambar 4.9 dibawah ini.

```
//Access ID Respond - RR#1 = Allow, RR#2 = Deny
if(idLogin.toInt() == 1)
{
  Serial.println("Acces Allow For " +String(cardID.substring(1)));
  idLogin = "";
  Buzz_ON; delay(80); Buzz_OFF; delay(80); //Buzzer
  Sole_ON; delay(2000); Sole_OFF; delay(80); //Solenoid ON 2 detik
}
if(idLogin.toInt() == 2)
{
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Buzz_ON; delay(80); Buzz_OFF; delay(80);
  Serial.println("Acces Deny For " + cardID.substring(1));
  idLogin = "";
}
}
```

Gambar 4.9. Respond dari Raspberry jika ID diizinkan atau ditolak
Sumber : Peneliti (2020)

4.3.6 Pembuatan Database Xampp Pada Raspberry

Pertama sebelum menginstall, pastikan daftar software kita uptodate, update dengan cara :

```
$sudo apt-get update
```

```
$sudo apt-get upgrade -y
```

a. Menginstall web server Apache 2

Apache2 dapat diinstall dengan menjalankan perintah:

```
$ sudo apt-get install apache2 libapache2-mod-php5
```

Untuk konfigurasi seperti virtualhost, terletak pada direktori /etc/apache2/site-available, jika kalian ingin membuat virtualhost, tinggal di copy saja file konfigurasi default pada direktori tersebut kemudian beri nama sesuai situs, lakukan penyetingan server name dan directory root (sama seperti di Windows) kemudian enable site tersebut dan reload atau restart service apache2 dengan perintah :

```
$ sudo a2ensite namasitus
```

```
$ sudo service apache2 restart
```

b. Menginstall modul PHP5

Modul PHP5 dapat diinstall dengan perintah:

```
$ sudo apt-get install php5 php5-mysql php-imagick
```

Untuk melihat daftar paket lengkap, gunakan perintah:

```
$ sudo apt-cache search php
```

maka akan muncul semua paket PHP yang mungkin kalian butuhkan.

c. Menginstall MySQL Server

Cara menginstall MySQL di Raspberry juga sangat mudah, versi server dan client dapat diinstall dengan perintah berikut:

```
$ sudo apt-get install mysql-server mysql-client
```

d. Menginstall PHPMyAdmin

Seperti XAMPP, kita biasanya bisa dengan mudah memanipulasi database melalui antarmuka web menggunakan PHPMyAdmin, begitupula di Linux dan Raspberry Pi PHPMyAdmin sudah pasti tersedia, caranya install:

```
$ sudo apt-get install phpmyadmin
```

Lakukan sedikit konfigurasi khususnya akun database ketika installasi, setelah selesai dapat di akses melalui <http://127.0.0.1/phpmyadmin> (<http://127.0.0.1/phpmyadmin>) atau ganti IP localhost dengan IP Raspberry Pi di jaringan.

Tabel 4.3 Pengujian Transmit data RFID Arduino ke Raspberry

No.	Arduino	Raspberry	Error
	Transmit	Receive	
1	SR#9A 6D EF 29	SR#9A 6D EF 29	0
2	R#9A 6D EF 29	R#9A 6D EF 29	0
3	SR#62 DE ED 1A	SR#62 DE ED 1A	0
4	R#62 DE ED 1A	R#62 DE ED 1A	0
5	SR#51 E7 48 1A	SR#51 E7 48 1A	0
6	R#51 E7 48 1A	R#51 E7 48 1A	0
7	SR#72 60 C4 1A	SR#72 60 C4 1A	0
8	R#72 60 C4 1A	R#72 60 C4 1A	0
Total Error			0

Sumber : Peneliti (2020)

Dari pengujian diatas ID Card yang dikirim dan diterima sesuai dengan nomor ID tidak hilang atau bertambah satupun, jika yang dikirim adalah SR# maka yang diterima pun harus SR#, dan juga yang dikirim R# maka yang diterima pun harus R#, tingkat keberhasilan 100% dan error 0%.

4.4. Pengujian Respond data Raspberry ke Arduino

4.4.1. Pembacaan Serial Dari Arduino

Pada bagian ini raspberry menerima data request dari Arduino yang kemudian akan dilakukan pemilahan data atau *parsing* dengan cara mendeteksi *header* data. Seperti yang telah dijelaskan pada bagian sebelumnya *header* data tersebut yaitu **SR#ID_RFID** sebagai request akses pintu masuk dan **R#ID_RFID** sebagai request penambahan ID baru. Berikut merupakan syntax program yang terdapat pada raspberry untuk proses pembacaan *header* data serial dari Arduino.

```

serial_port = serial.Serial('/dev/ttyACM0',9600)
def read_from_port(ser):
    while True:
        reading = ser.readline().decode().rstrip()
        paket_ser = reading.split('#')
        if paket_ser[0] == "SR":
            cekrf(paket_ser[1])
        elif paket_ser[0] == "R":
            tambahrf(paket_ser[1])

```

Gambar 4.10. Pembacaan Serial

Sumber : Peneliti (2020)

Setelah *header* data dan jenis request data dapat diketahui, maka proses selanjutnya yang dilakukan yaitu melakukan pengecekan ID RFID pada database. Untuk proses pengecekan ID dan penambahan ID dijelaskan pada bagian selanjutnya.

4.4.2. Pembacaan dan Perekaman Data ID RFID Baru Pada Database

Setelah proses pemilahan data request diatas telah dilakukan, pada bagian ini merupakan proses pengecekan data ID RFID pada database. Pengecekan ID dilakukan dengan cara membandingkan ID tersebut dengan yang sudah terdaftar pada database seperti yang ditunjukkan pada syntax program berikut.

```
mycursor = mydb.cursor()
query="SELECT COUNT(*) from rfid where data= %s "
val=(data,)
mycursor.execute(query,val)
result=mycursor.fetchone()
```

Gambar 4.11 Pembacaan Pada Database
Sumber : Peneliti (2020)

Result pada syntax program paling bawah pada gambar diatas akan menunjukkan respond bahwa ID RFID yang dilakukan pengecekan pada database sudah terdaftar atau belum. Jika sudah terdaftar maka *result* = 1, selain *result* = 1 maka ID dinyatakan belum terdaftar.

4.4.3. Pengiriman Respond Data Ke Arduino

Pada bagian ini merupakan detail pengiriman respond data dari raspberry ke Arduino setelah dilakukan pengecekan data ID RFID pada database yang telah dijelaskan pada bagian sebelumnya dengan hasil pengecekan yang ditunjukkan pada kondisi *result*, jika = 1 maka data ID telah terdaftar dan jika belum terdaftar maka *result* akan tidak sama dengan 1 atau != 1.

Pada proses pengecekan ID RFID akses pintu masuk atau disaat raspberry mendeteksi *header* data dari Arduino adalah SR#, jika *result* = 1 maka raspberry akan mengirimkan respond RR#1 ke Arduino sebagai repond bahwa ID RFID yang ingin mengakses pintu masuk telah terdaftar, namun jika belum terdaftar atau *result* tidak sama dengan (!=) 1 maka respond dari raspberry yang

dikirimkan yaitu RR#2 seperti yang ditunjukkan pada syntax program berikut

```

result=mycursor.fetchone()
if result[0]==1:
    print("RR#1") #ada data
    serial_port.write('RR#1'.encode())
    serial_port.write('\n'.encode())
else:
    print("RR#2") #tidak ada data
    serial_port.write('RR#2'.encode())
    serial_port.write('\n'.encode())

```

Gambar 4.12. Respond akses pintu masuk
Sumber : Peneliti (2020)

Hal yang sama ditunjukkan pada proses pendaftaran ID RFID baru atau disaat raspberry mendeteksi *header* data dari Arduino adalah R#, namun pada proses ini mempunyai perbedaan respond data yang dikirim ke Arduino. Jika *result* = 1 maka respond yang dikirim raspberry yaitu S#2 sebagai indikasi jika ID RFID sudah didaftarkan, namun *result* != 1 maka respond yang dikirim raspberry yaitu S#1 sebagai indikasi jika data ID RFID baru telah selesai disimpan dan sebelumnya memang belum didaftarkan seperti yang ditunjukkan pada gambar syntax program berikut

```

result=mycursor.fetchone()
if result[0]==1:
    print("S#2") #data sudah ada
    serial_port.write('S#2'.encode())
    serial_port.write('\n'.encode())
else:
    sql = "INSERT INTO rfid (data) VALUES (%s)"
    val = (data,)
    mycursor.execute(sql, val)
    mydb.commit()
    print("S#1") #sukses nambah data
    serial_port.write('S#1'.encode())
    serial_port.write('\n'.encode())

```

Gambar 4.13. Respond pendaftaran ID baru
Sumber : Peneliti (2020)

Tabel 4.4 Pengujian Respond data Raspberry ke Arduino

No.	Raspberry	Arduino	Error
	<i>Transmit</i>	<i>Receive</i>	
1	S#9A 6D EF 29	S#9A 6D EF 29	0
2	RR#9A 6D EF 29	RR#9A 6D EF 29	0
3	S#62 DE ED 1A	S#62 DE ED 1A	0
4	RR#62 DE ED 1A	RR#62 DE ED 1A	0
5	S#51 E7 48 1A	S#51 E7 48 1A	0
6	RR#51 E7 48 1A	RR#51 E7 48 1A	0
7	S#72 60 C4 1A	S#72 60 C4 1A	0
8	RR#72 60 C4 1A	RR#72 60 C4 1A	0
Total Error			0

Sumber : Peneliti (2020)

Dari pengujian diatas ID Card yang dikirim dan diterima sesuai dengan nomor ID tidak hilang atau bertambah satupun, jika yang dikirim adalah RR# maka yang diterima pun harus RR#, dan juga yang dikirim S# maka yang diterima pun harus S#, tingkat keberhasilan 100% dan error 0%.

4.5. Pengujian Pembacaan Barcode

Untuk detail proses rancangan software absensi menggunakan barcode pada raspberry dijelaskan sebagai berikut :

1. Pembacaan Barcode

Proses pembacaan barcode dilakukan untuk mengetahui digit barcode tersebut. Perangkat barcode terhubung dengan raspberry menggunakan USB sehingga data hasil scan dapat diakses menggunakan program serial seperti yang ditunjukkan pada syntax program berikut.

```

def key(event):
    global all
    if event.char != '\r':
        all = all + event.char
    else:
        cekbarcode(all)
        all=""

```

Gambar 4.14. Syntax program pembacaan barcode
Sumber : Peneliti (2020)

Setelah digit barcode diketahui selanjutnya dilakukan perekaman daftar hadir pada database. ID barcode pada bagian ini sebelumnya telah dilakukan registrasi secara manual pada database dan menjadi 1 bagian dengan kartu RFID yang telah didaftarkan sebagai akses pintu masuk.

2. Perekaman Daftar Hadir Laboratorium Menggunakan Barcode

Pada proses pencatatan atau perekaman absensi menggunakan barcode terdapat 2 kondisi yaitu kondisi jam kedatangan dan jam pulang pada setiap ID dan data barcode yang telah melakukan proses absensi dengan tabel format perekaman seperti yang ditunjukkan pada gambar berikut.

<i>id</i>	<i>Data</i>	<i>timein</i>	<i>timeout</i>
-----------	-------------	---------------	----------------

Gambar 4.15. Tabel format perekaman absensi
Sumber : Peneliti (2020)

Proses scanning barcode juga dilakukan pengecekan pada database sebagai indikasi bahwa ID barcode telah terdaftar. Proses pengecekan sama halnya seperti yang ditunjukkan pada gambar 3.21 diatas, jika *result* = 1 maka ID terdaftar namun dengan penambahan *counter* sebagai indikasi kondisi *scanning* pertama atau kedua.

Pada proses *scanning* barcode yang pertama (*counter* = 1) sistem akan langsung mencatat waktu dan tanggal kemudian akan disimpan pada kolom *timein* pada database sebagai kondisi jam

kedatangan dan pada proses *scanning* kedua (*counter* = 2) waktu dan tanggal akan disimpan pada kolom *timeout* pada database sebagai kondisi jam pulang seperti yang ditunjukkan pada gambar syntax program berikut.

```

if result[0]==1 and counter == 1:
    print("Kode Barcode : ")
    print(data)
    label.config(text=data)
    sql = "UPDATE barcode SET timein = %s WHERE data = %s"
    val = (time,data)
    mycursor.execute(sql, val)
    mydb.commit()
    counter = 2
elif result[0]==1 and counter == 2:
    print("Kode Barcode : ")
    print(data)
    label.config(text=data)
    sql = "UPDATE barcode SET timeout = %s WHERE data = %s"
    val = (time,data)
    mycursor.execute(sql, val)
    mydb.commit()
    counter = 1
else:
    print("Kode Barcode Tidak Tersedia")
    label.config(text="Kode Barcode Tidak Tersedia")
print(counter)

```

Gambar 4.16. Syntax pemrograman untuk perekaman daftar hadir
Sumber : Peneliti (2020)

Pengujian Barcode ini bertujuan untuk menguji Barcode tersebut dapat dipindai atau tidak, pengujian ini meliputi pencahayaan dalam Barcode, jarak Barcode, dan bentuk dari Barcode.

a) Pencahayaan

Pada proses peletakan Barcode, pencahayaan tidak mempengaruhi proses pemindaian karena walaupun tempat tersebut kurang cahaya Barcode tetap dapat dipindai melalui Barcode Scanner. Hasil pengujian pencahayaan dapat dilihat pada Tabel 4.5.

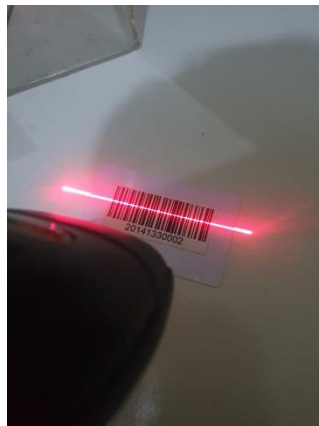
Tabel 4.5 Pengujian Pencahayaan Pemindaian Barcode

Pencahayaan	Hasil Pembacaan
Sangat Gelap	Berhasil
Gelap	Berhasil
Redup	Berhasil
Terang	Berhasil
Sangat Terang	Berhasil

Sumber : Peneliti (2020)

b) Jarak Barcode dengan Barcode Scanner

Pengujian sistem telah dilakukan berdasarkan jarak dan sudut pada saat pembacaan barcode untuk mendapatkan posisi terbaik pembacaan barcode terhadap Barcode Scanner. Hasil sebagaimana ditunjukkan pada Tabel 4.6.



Gambar 4.17. Pengujian Jarak Barcode Scanner
Sumber : Peneliti (2020)

Tabel 4.6 Pengujian jarak pemindaian Barcode pada Barcode Scanner

No	Jarak (cm)	Hasil Pembacaan
1	2	Berhasil
2	4	Berhasil
3	6	Berhasil
4	8	Tidak Berhasil
5	10	Tidak Berhasil

Sumber : Peneliti (2020)

Dari tabel 4.6 diatas diperoleh hasil pembacaan jarak yang optimal yaitu untuk jarak 2 cm .

c) Bentuk Barcode

Bentuk dari Barcode yang dapat dipindai oleh Barcode Scanner yaitu harus utuh dan tidak rusak, dan Barcode tetap bisa dipindai walaupun peletakkan Barcode tersebut terbalik. Hasil dari pengujian bentuk Barcode dapat dilihat pada Tabel 4.7.

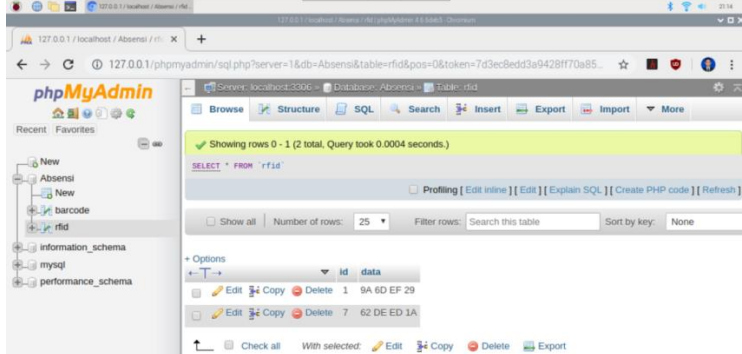
Tabel 4.7 Pengujian bentuk Barcode

Bentuk Barcode	Hasil Pemindaian
25 %	Tidak Terbaca
50 %	Tidak Terbaca
75 %	Tidak Terbaca
100 %	Terbaca

Sumber : Peneliti (2020)

4.6. Pengujian simpan dan baca data RFID pada database

Pengujian pada program tampilan dalam komputer fungsi yang diuji adalah Hasil pembacaan ID RFID yang sudah terdaftar dalam database melalui komputer serta hasil yang bisa disimpan,



Gambar 4.18. Pengujian Web Daftar ID Card RFID
Sumber : Peneliti (2020)

	id	data
<input type="checkbox"/> Edit Copy Delete	1	9A 6D EF 29
<input type="checkbox"/> Edit Copy Delete	7	62 DE ED 1A
<input type="checkbox"/> Edit Copy Delete	8	51 E7 48 1A
<input type="checkbox"/> Edit Copy Delete	9	72 60 C4 1A

Gambar 4.19. Hasil RFID untuk melihat ID Card
Sumber : Peneliti (2020)

Tabel 4.8 Hasil Pengujian ID yang terdaftar di database

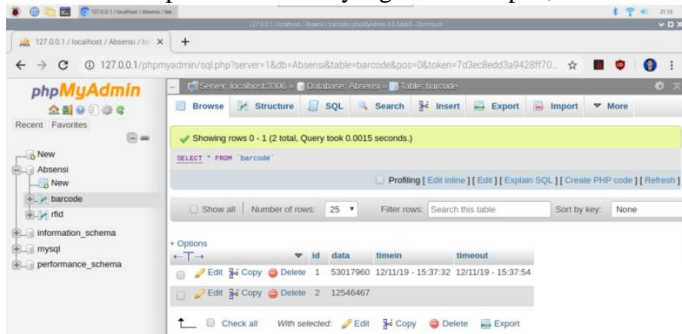
ID	DATA
1	9A 6D EF 29
2	62 DE ED 1A

3	51 E7 48 1A
4	72 60 C4 1A

Sumber : Peneliti (2020)

4.7. Hasil penyimpanan database perekaman absensi dengan barcode

Pengujian pada program tampilan dalam komputer fungsi yang diuji adalah hasil pembacaan ID barcode yang sudah terdaftar dalam database melalui komputer serta hasil yang bisa disimpan,



Gambar 4.20. Pengujian Tampilan Web Daftar ID Barcode Mahasiswa

Sumber : Peneliti (2020)

id	data	timein	timeout	state
3	20141330001	01/27/20 - 20:14:54	01/27/20 - 0 20:16:06	0
4	20141330002	01/27/20 - 20:02:53	01/27/20 - 0 20:15:03	0
5	20141330003	01/27/20 - 20:02:56	01/27/20 - 0 20:16:12	0


```

6 20141330004 01/27/20 01/27/20 - 0
    -          20:16:21
    20:02:59
7 20141330005 01/27/20 01/27/20 - 0
    -          20:16:17
    20:03:02

```

Gambar 4.21. Hasil Scan Barcode untuk absensi
Sumber : Peneliti (2020)

Tabel 4.9 Hasil penyimpanan absensi dengan barcode

NO.	ID Barcode	Jam Masuk	Jam Keluar
1	20141330001	01/27/20 – 20:14:54	01/27/20 – 20:16:06
2	20141330002	01/27/20 – 20:02:53	01/27/20 – 20:15:03
3	20141330003	01/27/20 – 20:02:56	01/27/20 – 20:16:12
4	20141330004	01/27/20 – 20:02:59	01/27/20 – 20:16:21
5	20141330005	01/27/20 – 20:03:02	01/27/20 – 20:16:17

Sumber : Peneliti (2020)