

BAB 4

HASIL PENGUJIAN DAN PEMBAHASAN

4.1. Metode Pengujian

Tujuan pengujian alat adalah untuk membuktikan bahwa sistem yang diimplementasikan telah memenuhi spesifikasi yang telah direncanakan sebelumnya. Berdasarkan spesifikasi sistem yang telah dijalankan sebelumnya, maka dilakukan pengujian terhadap sistem menggunakan beberapa metode pengujian. Metode pengujian dipilih berdasarkan fungsi operasional dan beberapa parameter yang ingin diketahui dari sistem tersebut. Dalam penelitian ini dipilih 2 macam metode pengujian, yaitu pengujian fungsional dan pengujian kinerja sistem secara keseluruhan. Pengujian fungsional digunakan untuk membuktikan bahwa sistem yang diimplementasikan dapat memenuhi persyaratan fungsi operasional yang direncanakan sebelumnya. Sedangkan pengujian kinerja sistem secara keseluruhan bertujuan untuk memperoleh beberapa parameter yang dapat menunjukkan kemampuan dan kehandalan sistem dalam menjalankan fungsi operasionalnya. Dengan menggunakan 2 metode penelitian di atas, diharapkan pada sistem dapat ditemukan kelebihan dan kekurangan dari alat yang dibuat sehingga memudahkan jika dilakukan pengembangan nantinya.

4.2. Pengujian Fungsional

Pengujian fungsional dilakukan dengan cara menguji bagian - bagian dari sistem keseluruhan yang terdiri dari :

1. Pengujian NFC (*Near Field Communication*)

2. Pengujian mode pendaftaran dan mode penghapusan ID (*Identity*)
3. Pengujian penyimpanan data di MicroSD
4. Pengujian *solenoid* sebagai kunci pintu

Berikut hasil pengujian masing-masing bagian fungsional:

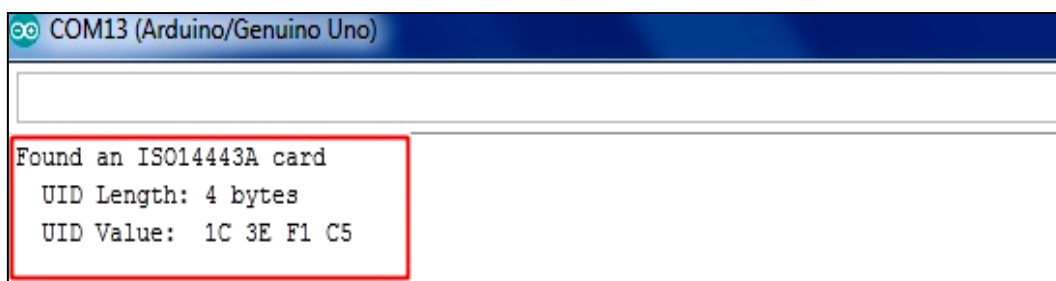
4.2.1. Pengujian NFC (*Near Field Communication*)

Pengujian NFC tipe PN532 dilakukan dengan menghubungkan NFC dengan Arduino Uno. NFC terhubung ke Arduino menggunakan komunikasi I2C dengan menyambungkan pin SDA (*Serial Data*) ke pin A4, pin SCL (*Serial Clock*) ke A5, VCC ke pin 5V dan *Ground* ke pin GND. Setelah terhubung, sebuah *tag* NFC didekatkan pada *reader* NFC untuk dibaca seperti yang ditunjukkan oleh Gambar 4.1.



Gambar 4.1 *Reader* membaca ID sebuah *tag* NFC

Reader NFC akan membaca nomor ID dari *tag* NFC yang di dekatkan ke *reader*, masing – masing *tag* memiliki nomor ID yang unik. Nomor ID tersebut dikonversikan dari bilangan desimal ke bilangan hexadesimal. Pengkonversian nomor ID dilakukan setelah nomor ID terbaca oleh *reader* NFC. Dalam 1 buah *tag* NFC memiliki nomor ID yang berukuran 4 *bytes*. Nomor ID *tag* yang dibaca akan ditampilkan pada layar *serial*. Gambar 4.2 adalah tampilan dari layar *serial* pada Arduino yang membaca nomor ID dari *tag* NFC.



Gambar 4.2 Tampilan nomor ID *tag* NFC di layar *serial*

Hasil dari pengujian NFC menunjukkan bahwa *tag* NFC yang telah dibaca memiliki ID yaitu 1C3EF1C5 dengan panjang 4 *bytes* dan termasuk tipe kartu ISO 14443A. Kemudian dilakukan pengujian pada jarak baca *tag* NFC terhadap *reader* dengan 5 kartu yang berbeda. Tabel 4.1 merupakan hasil pengujian jarak *tag* NFC terhadap *reader*.

Tabel 4.1 Pengujian jarak baca *tag* NFC

<i>Tag</i> NFC	Hasil Pengujian (cm)					Rata-rata Jarak Baca (cm)
	Uji 1	Uji 2	Uji 3	Uji 4	Uji 5	
Kartu 1 Putih Polos	2	2,6	2,5	2,4	2,7	2,44
Kartu 2 E-Money	0,8	0,7	0,4	0,5	0,6	0,6
Kartu 3 KTM UMY	1,8	2	1,9	2,1	1,7	1.9
Kartu 4 Kimia Farm	1,7	1,5	1,9	1,9	1,6	1,72
Kartu 5 Flaurent	2	1,9	2,1	1,5	1,8	1,86

Maka dari hasil pengujian jarak baca pada Table 4.1, *tag* NFC dapat dibaca oleh *reader* dengan jarak baca maksimal yang berbeda-beda. Pada kartu 2 menunjukkan bahwa *tag* tersebut memiliki jarak baca maksimal hanya sekitar 0,6-0,8 cm. Lain halnya dengan kartu 1 yang memiliki jarak baca maksimal relatif jauh dibandingkan dengan kartu lainnya yaitu sekitar 2-2,7 cm. Selain itu dilakukan juga pengujian membaca *tag* NFC dengan adanya penghalang sebuah kertas pada *reader*, dan *tag* NFC masih bisa terbaca dengan dihalangi selebar kertas. NFC *reader* juga masih bisa membaca *tag* NFC dari belakang pintu yang tertutup triplek setebal 0,5 cm.

Setelah itu, dilakukan pengujian pada waktu tunda *reader* membaca *tag* NFC. Pengujian ini dilakukan untuk mengetahui kecepatan *reader* dalam membaca *tag* dalam cakupan frekuensi yang dipancarkan *reader*. Tabel 4.2 merupakan hasil pengujian waktu tunda (*delay*) *reader* membaca *tag*.

Tabel 4.2 Pengujian waktu tunda *reader* NFC

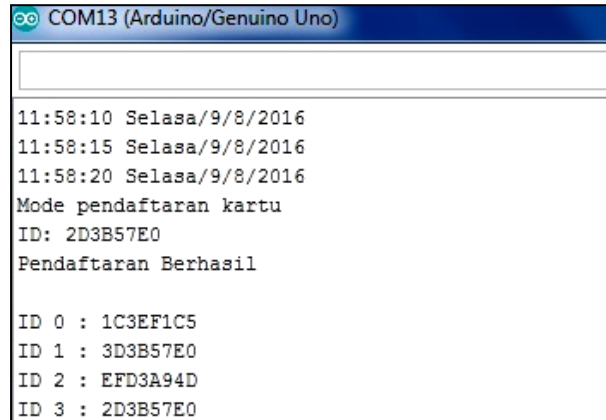
<i>Tag</i> NFC	Hasil Pengujian (milidetik)			Rata-rata Waktu Tunda (milidetik)
	Uji 1	Uji 2	Uji 3	
Kartu 1 Putih Polos	3	2	3	2,66
Kartu 2 E-Money	2	1	2	1,66
Kartu 3 KTM UMY	2	2	3	2,33
Kartu 4 Kimia Farm	2	2	3	2,33
Kartu 5 Flaurent	3	2	2	2,33

Hasil dari pengujian waktu tunda *reader* membaca *tag* pada Tabel 4.2 menunjukkan bahwa setiap kartu memiliki waktu tunda baca yang berbeda namun relatif sama sekitar 2,3 milidetik. Jadi, dari hasil pengujian NFC yang meliputi

membaca ID *tag*, pengujian jarak baca maksimal dan pengujian waktu tunda baca *tag* dapat disimpulkan bahwa *tag* NFC memiliki jarak baca maksimal 2 – 2,7 cm dengan waktu tunda baca sekitar 2,3 milidetik.

4.2.2. Pengujian Mode Pendaftaran dan Mode Penghapusan ID

Pengujian pendaftaran dan penghapusan ID dilakukan dengan menekan tombol nomor 2 yaitu Daftar ID selama beberapa detik hingga muncul tulisan Mode Pendaftaran ID pada layar serial. Setelah itu *tag* NFC didekatkan ke *reader* untuk dibaca ID yang akan didaftarkan di memori Arduino. Jika ID sudah didaftarkan, maka akan tertampil tulisan “ID sudah terdaftar” pada layar serial, namun jika belum terdaftar maka akan tertampil “Pendaftaran Berhasil”. Kemudian menampilkan daftar ID yang ada di memori seperti Gambar 4.3.



```
COM13 (Arduino/Genuino Uno)
11:58:10 Selasa/9/8/2016
11:58:15 Selasa/9/8/2016
11:58:20 Selasa/9/8/2016
Mode pendaftaran kartu
ID: 2D3B57E0
Pendaftaran Berhasil

ID 0 : 1C3EF1C5
ID 1 : 3D3B57E0
ID 2 : EFD3A94D
ID 3 : 2D3B57E0
```

Gambar 4.3 Mode pendaftaran ID

Hasil dari pengujian Mode Pendaftaran ID pada Gambar 4.3 menunjukkan bahwa *tag* NFC dengan ID: 2D3B57E0 telah berhasil didaftarkan dan disimpan di memori Arduino pada slot ID ke 3.

Selanjutnya dilakukan pengujian mode penghapusan ID dari memori Arduino dengan menekan tombol nomor 3 yaitu hapus ID. Tombol ditekan selama

beberapa detik, hingga muncul tulisan Mode Hapus ID pada layar serial. Setelah mode hapus ID aktif, isi memori akan ditampilkan kembali seperti yang ditunjukkan pada Gambar 4.4.

```
12:6:41 Selasa/9/8/2016
Mode hapus kartu
ID 0 : 1C3EF1C5
ID 1 : 3D3B57E0
ID 2 : EFD3A94D
ID 3 : 2D3B57E0
```

Gambar 4.4 Mode penghapusan ID aktif

Untuk memilih slot ID yang akan dihapus dengan menekan tombol 2 dan tombol 3 sebagai menu ke atas dan ke bawah. Setelah memilih slot yang akan dihapus, tombol 4 ditekan untuk mengeksekusi perintah menghapus ID yang telah dipilih seperti yang ditunjukkan oleh Gambar 4.5.

```
ID 3 : 2D3B57E0
ID 3 : 2D3B57E0
Terhapus !
12:7:8 Selasa/9/8/2016
12:7:13 Selasa/9/8/2016
Start
MicroSD Ok
Setup Done !
ID 0 : 1C3EF1C5
ID 1 : 3D3B57E0
ID 2 : EFD3A94D
ID 3 : 0
ID 4 : 0
```

Gambar 4.5 Menghapus ID di memori Arduino

Hasil dari pengujian Mode Penghapusan ID menunjukkan bahwa slot ID ke 3 dengan ID: 2D3B57E0 telah berhasil dihapus dari memori Arduino. Maka slot ID ke 3 di memori Arduino akan kembali kosong seperti yang ditunjukkan pada Gambar 4.5.

4.2.3. Pengujian Penyimpanan data di MicroSD

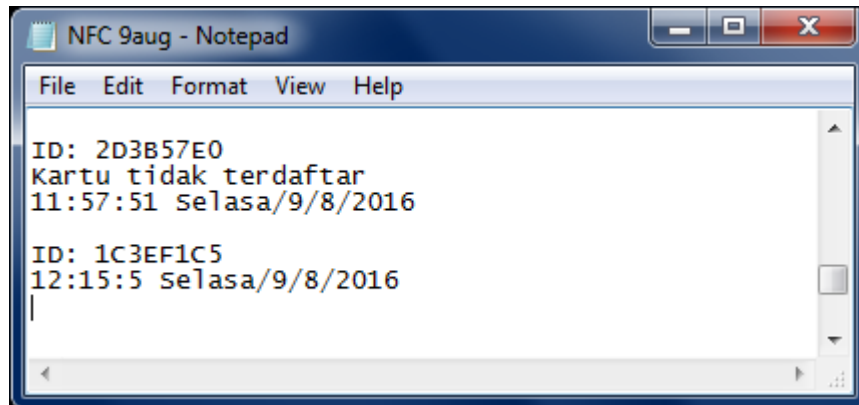
Pengujian fungsional penyimpanan data di MicroSD dilakukan untuk menguji pada sistem penyimpanan atau pencatatan data di MicroSD melalui Ethernet Shield. Berikut potongan program yang digunakan untuk memberikan perintah menyimpan data pada MicroSD apabila ada ID yang dibaca oleh *reader*, baik yang belum terdaftar atau yang sudah terdaftar di memori Arduino dengan keterangan waktu baca ID secara *real time*:

```

dataFile.print("ID: ");
dataFile.println(data_nfc,HEX);
{if(data_nfc==memory_nfc[i]){
    dataFile.print(jam);
    dataFile.print(":");
    dataFile.print(menit);
    dataFile.print(":");
    dataFile.print(detik);
    dataFile.print(" ");
    dataFile.print(font[hari]);
    dataFile.print("/");
    dataFile.print(tanggal);
    dataFile.print("/");
    dataFile.print(bulan);
    dataFile.print("/");
    dataFile.println(tahun+2000);
    dataFile.println();
    dataFile.close();}
i++;if(i>29){Serial.println("Kartu Tidak
Terdaftar");dataFile.println("Kartu tidak terdaftar");}

```

Maka hasil pengujian penyimpanan data di MicroSD untuk ID yang sudah terdaftar atau yang belum terdaftar ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Penyimpanan data di MicroSD

Hasil dari pengujian penyimpanan data di MicroSD menunjukkan bahwa ID: 2D3B57E0 belum terdaftar di memori Arduino dan dibaca oleh *reader* pada pukul 11:57:51, Selasa 9/8/2016 dan ID: 1C3EF1C5 sudah terdaftar dan dibaca *reader* pada pukul 12:15:5, Selasa 9/8/2016. Data tersebut telah tersimpan pada file NFC.txt yang berada pada MicroSD.

Kemudian pengujian penyimpanan data pada microSD dilanjutkan dengan membaca 5 buah *tag* dengan ID yang berbeda dan dicatat di file NFC.txt. Tabel 4.3 merupakan hasil pengujian penyimpanan data pada MicroSD.

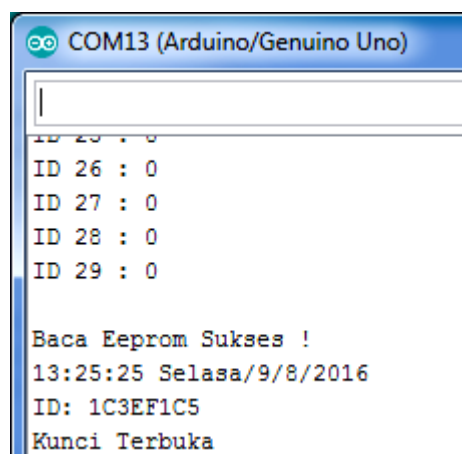
Tabel 4.3 Pengujian penyimpanan data di MicroSD

Tag NFC	Hasil Pengujian Penyimpanan Data				
	Uji 1	Uji 2	Uji 3	Uji 4	Uji 5
Kartu 1C3EF1C5	Tersimpan	Tersimpan	Tersimpan	Tersimpan	Tersimpan
Kartu EFD3A94D	Gagal	Gagal	Gagal	Tersimpan	Tersimpan
Kartu 2D3B57E0	Tersimpan	Gagal	Tersimpan	Gagal	Tersimpan
Kartu 5D3A57E0	Tersimpan	Tersimpan	Tersimpan	Tersimpan	Tersimpan
Kartu 3D3B57E0	Tersimpan	Tersimpan	Tersimpan	Tersimpan	Tersimpan

Hasil dari pengujian penyimpanan data pada Tabel 4.3 menunjukkan bahwa dari 5 buah *tag* NFC yang diuji, beberapa *tag* ada yang mengalami kegagalan menyimpan data di MicroSD setelah ID dibaca oleh *reader*. Indikator data tersimpan pada microSD ditunjukkan oleh lampu LED pada *Ethernet shield* yang berada diantara pin AREF dan pin GND. Kegagalan penyimpanan data di MicroSD bisa disebabkan oleh beberapa hal, yaitu pada bagian *hardware* yang tidak terpasang secara benar atau jarak dari *reader* dan *tag* yang kurang dekat. Maka dapat disimpulkan bahwa penyimpanan data di MicroSD dapat berkerja sesuai fungsinya sebesar 80%.

4.2.4. Pengujian *Solenoid* Sebagai kunci pintu

Pengujian fungsional *solenoid* sebagai kunci pintu dilakukan untuk memastikan *solenoid* dapat bekerja sebagai kunci pintu sesuai dengan rancangan. *Solenoid* diberi catu daya 12 V melalui sebuah modul *relay* yang telah terhubung ke Arduino. *Solenoid* akan aktif menarik *plunger* saat *reader* membaca ID yang telah terdaftar di memori Arduino seperti yang ditunjukkan oleh Gambar 4.7.



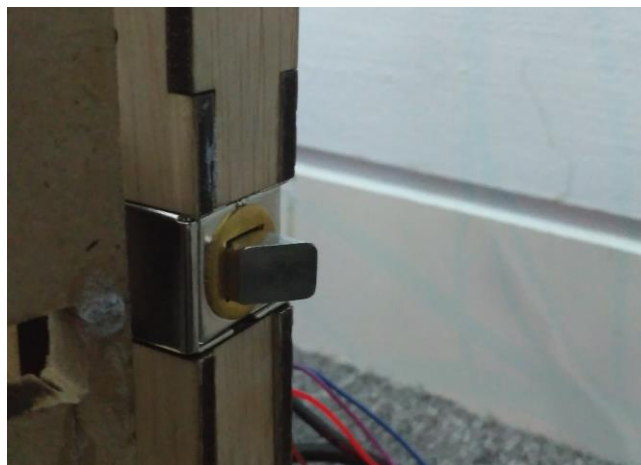
Gambar 4.7 ID yang dapat membuka kunci pintu

Gambar 4.7 menunjukkan bahwa ID: 1C3EF1C5 yang dibaca oleh *reader* NFC dapat mengaktifkan *solenoid* untuk membuka kunci pintu karena ID tersebut telah terdaftar di memori Arduino. Berikut Gambar 4.8 menunjukkan keadaan saat *solenoid* aktif dan kunci terbuka.



Gambar 4.8 Solenoid aktif dan kunci terbuka

Setelah *solenoid* membuka kunci pintu, apabila *limit switch* aktif, maka *solenoid* akan mengunci pintu kembali seperti yang ditunjukkan oleh Gambar 4.9.



Gambar 4.9 Solenoid tidak aktif dan mengunci pintu

Begitu pula halnya saat ID yang dibaca belum terdaftar di memori Arduino, *solenoid* akan tetap dalam posisi mengunci pintu. Gambar 4.10 menunjukkan saat ID yang tidak terdaftar dibaca oleh *reader* dan tidak dapat membuka kunci pintu.

```
ID: 3D3B57E0
Kartu tidak terdaftar
21:26:34 Senin/8/8/2016

ID: 2D3B57E0
Kartu tidak terdaftar
21:26:39 Senin/8/8/2016

ID: 5D3A57E0
Kartu tidak terdaftar
21:26:42 Senin/8/8/2016
```

Gambar 4.10 ID yang tidak terdaftar di memori Arduino

Hasil dari pengujian *solenoid* sebagai pembuka kunci pintu menunjukkan bahwa ID: 1C3EF1C5 telah terdaftar di memori Arduino dan dapat mengaktifkan *solenoid* untuk membuka kunci. Kemudian pada saat *tag* NFC dengan ID: 3D3B57E0 , 2D3B57E0 dan 5D3A57E0 dibaca, *solenoid* tidak aktif dan pintu masih terkunci.

4.3. Pengujian Alat Keseluruhan

Pengujian alat secara keseluruhan dilakukan untuk menunjukkan bahwa alat dapat berfungsi sesuai dengan rancangan awal dan dilakukan secara sistematis berdasarkan rancangan. *Tag* NFC yang terdaftar di memori Arduino sebanyak 3 buah dan 2 buah *tag* lainnya tidak terdaftar. ID yang telah terdaftar adalah ID 1C3EF1C5, 2D3B57E0 dan 3D3B57E0. Sedangkan ID yang tidak terdaftar adalah EFD3A94D dan 5D3A57E0.

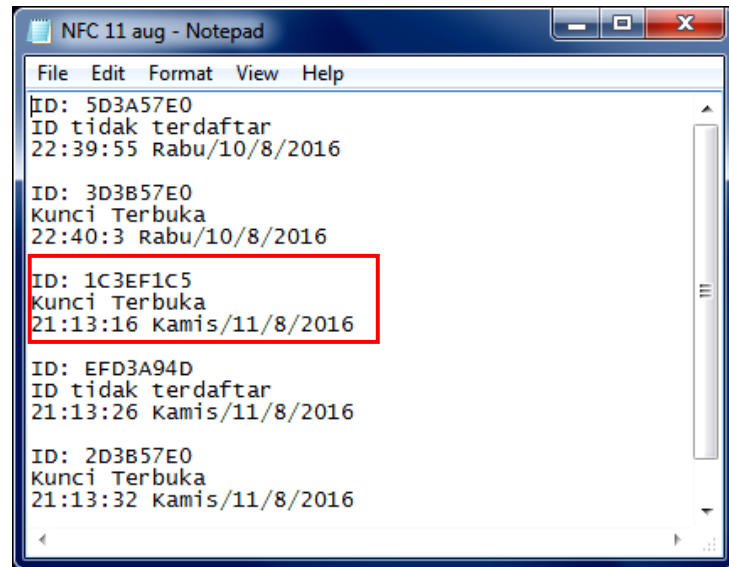
Pengujian pertama dilakukan pembacaan *tag* dengan ID 1C3EF1C5 yang didekatkan ke *reader* NFC. ID tersebut dapat membuka kunci pintu, karena sudah

terdaftar di memori Arduino, maka pintu bisa dibuka seperti yang ditunjukkan oleh Gambar 4.11.



Gambar 4.11 Kunci pintu terbuka

Saat pintu dalam keadaan terbuka, *solenoid* akan tetap dalam kondisi aktif hingga pintu tertutup kembali dengan indikator *limit switch* aktif dan pintu akan terkunci kembali. Selama solenoid aktif pun, *reader* tidak dapat membaca *tag* lain sebelum pintu dalam keadaan terkunci kembali. Saat ID 1C3EF1C5 dapat membuka kunci pintu, pada microSD disimpan data berupa keterangan kunci terbuka dan waktu akses seperti yang ditunjukkan oleh Gambar 4.12.



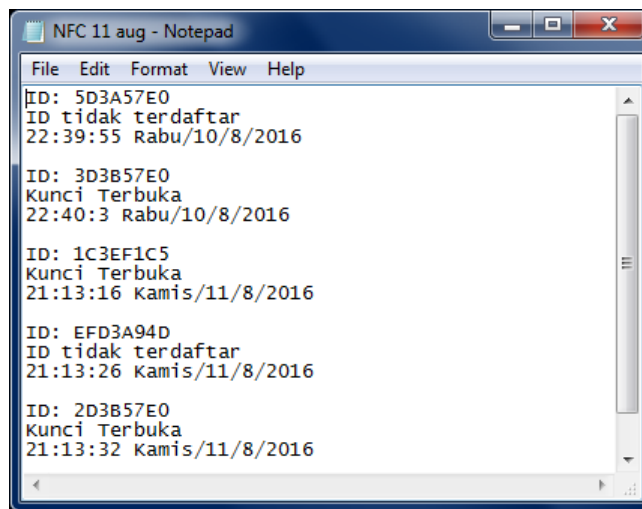
Gambar 4.12 ID 1C3EF1C5 tercatat di MicroSD

Pengujian selanjutnya dilakukan pembacaan *tag* dengan nomor ID EFD3A94D yang dibaca oleh *reader* NFC. *Buzzer* mengeluarkan suara *beep* 2 kali sebagai tanda bahwa ID tersebut tidak terdaftar di memori Arduino. Maka ID EFD3A94D tidak dapat membuka kunci pintu dan pintu masih dalam keadaan tertutup seperti yang ditunjukkan oleh Gambar 4.13.



Gambar 4.13 Pintu tertutup dan terkunci

Selama pintu dalam keadaan tertutup dan *solenoid* tidak aktif, maka ID lain dapat dibaca oleh *reader*. Setiap ID yang mencoba untuk membuka kunci pintu, baik yang terdaftar atau tidak terdaftar akan selalu dicatat dan disimpan datanya di microSD dengan rincian keterangan waktu dan keterangan kondisi kunci pintu seperti yang ditunjukkan oleh Gambar 4.14.



Gambar 4.14 Rekap data yang tersimpan di MicroSD

Hasil dari pengujian alat keseluruhan menunjukkan bahwa NFC dapat memberikan keamanan dalam mengakses pintu dengan mencatat ID *tag*, waktu akses pintu dan keterangan kondisi kunci pintu di sebuah MicroSD. Untuk ID yang sudah terdaftar maupun yang tidak terdaftar, sistem akan selalu mencatat dan menyimpan datanya di MicroSD.