

## BAB IV

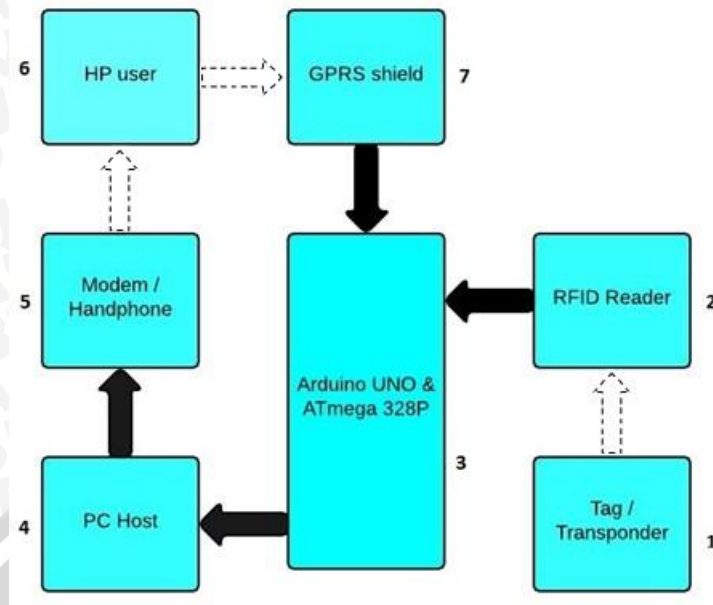
### PERANCANGAN DAN PEMBUATAN ALAT

Perancangan Sistem Pengecekan Jumlah Kendaraan ini dilakukan secara bertahap sehingga akan memudahkan analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan ini terdiri dari beberapa hal sebagai berikut :

- 1). Perancangan sistem pengecekan jumlah kendaraan.
- 2). Perancangan perangkat keras (*hardware*) terdiri dari rangkaian sistem RFID.
- 3). Perancangan perangkat lunak (*software*) terdiri dari penyimpanan *tag* ke dalam database, pengambilan data dari database, pengiriman data melalui SMS (*Short Message Service*), serta program antarmuka PC.

#### 4.1 Perancangan Sistem

Prinsip kerja alat ini dimulai dari proses pembacaan *tag* oleh RFID *reader*. Pembacaan *tag* oleh RFID *reader* hanya dapat dilakukan bila *tag* dalam posisi ON dan berada dalam jarak pembacaan RFID *reader*. Apabila hal tersebut sudah terpenuhi maka kode dari *tag* tersebut akan dimasukkan dan dicocokkan pada database yang disimpan pada PC, untuk menentukan apakah *tag* masuk atau *tag* keluar. Ketika *user* ingin mengakses sistem dengan SMS, maka SMS yang masuk akan diterima oleh GPRS *Shield* dan sistem akan mengakses database dan menghitung jumlah kendaraan yang tersedia dan jumlah kendaraan yang keluar. Kemudian *Handphone* yang difungsikan sebagai modem akan mengirimkan SMS balasan berisi jumlah kendaraan yang tersedia dan jumlah kendaraan yang keluar kepada *user*. Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Diagram Blok Sistem

Berikut penjelasan langkah-langkah alur blok sistem dan fungsi masing-masing setiap blok adalah

1. **Tag**; dipasang pada kendaraan atau mobil dengan cara ditempelkan. *Tag* akan mengalami perpindahan data (*coupling*) saat berada dalam jangkauan *reader*. Baterai yang terdapat di dalam *tag* digunakan untuk memancarkan gelombang radio kepada *reader* sehingga *reader* dapat membaca data yang terdapat pada *tag*.
2. **RFID reader**; berfungsi untuk membaca *tag* yang telah ditempelkan pada kendaraan. Kemudian data yang didapat, akan dikirimkan ke Arduino untuk dilakukan pengolahan lebih lanjut.
3. **Arduino**; *board* mikrokontroler yang digunakan untuk menerjemahkan data yang diperoleh dari pembacaan *tag* RFID oleh RFID *reader* serta pengolah aktivitas alat secara keseluruhan.
4. **PC**; digunakan untuk menampilkan ID *tag* beserta tanggal masuk maupun tanggal keluar melalui sebuah antarmuka yang dibuat menggunakan *software* Microsoft Visual Studio C# 2012.
5. **GPRS Shield**; dipakai sebagai sarana SMS gateway yang berfungsi menerima perintah dari *user* ataupun dari mikrokontroler yang berupa AT command untuk menyampaikan informasi.
6. **Modem atau Handphone**; dipakai sebagai sarana SMS gateway yang berfungsi mengirim informasi yang diperlukan oleh *user*.



7. **HP User**; User cukup mengirimkan sebuah SMS yang berisi perintah kepada alat, untuk kemudian alat akan mengirimkan info mengenai jumlah kendaraan yang tersedia dan jumlah kendaraan yang keluar.

## 4.2 Perancangan Perangkat Keras

### 4.2.1 Perancangan Rangkaian RFID reader

Perangkat ini digunakan sebagai pembaca *tag* yang terbaca yang terhubung secara serial dengan komputer. RFID *reader* ini nantinya akan mengirimkan data ASCII. Data yang akan dikirimkan tersebut mempunyai struktur sebagai berikut: STX(Awalan data) – MT(Tipe data) – RT(Tipe RFID *reader*) – 4 digit nomor kartu RFID *tag* dalam kode ASCII – 2 digit nomor proyek RFID *tag* dalam kode ASCII – ETX(Akhir data) – LRC1(byte pertama dari *checksum*) – LRC2(byte kedua dari *checksum*) – CR(*Carriage return*(0x0D)) – LR(*Line feed*(0x0A)).

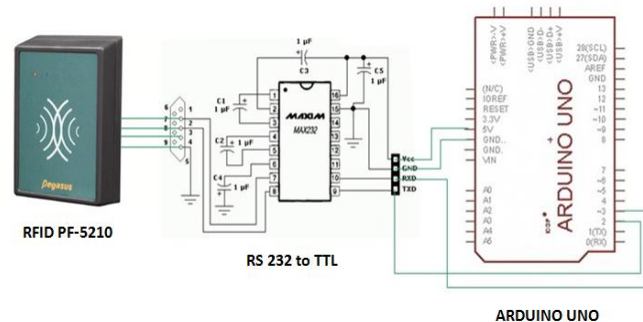
Rangkaian koneksi antara Arduino dengan RFID *reader* ditunjukkan dalam Gambar 4.2. Dalam perancangan ini, RFID *reader* dihubungkan dengan Arduino menggunakan converter RS232 to ttl. Dimana konfigurasi pinnya sebagai berikut :

Konfigurasi pin RFID *reader* ke RS232

- Pin A pada RFID *reader* dihubungkan ke Pin Rx pada RS232.
- Pin C pada RFID *reader* dihubungkan ke Pin Tx pada RS232.
- Pin G pada RFID *reader* dihubungkan ke pin DTR+ pada RS232.
- Pin H pada RFID *reader* dihubungkan ke pin GND pada RS232.

Konfigurasi pin RS232 ttl ke Arduino

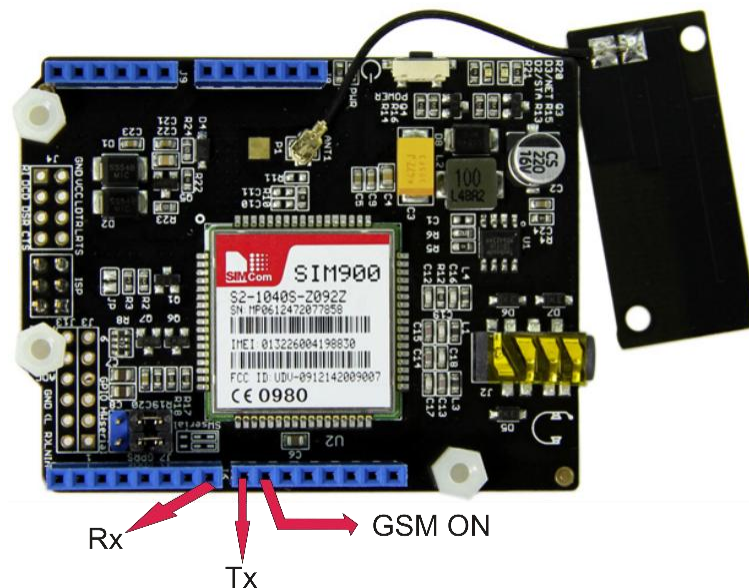
- Pin Rx pada RS232 dihubungkan ke pin 3 pada Arduino.
- Pin Tx pada RS232 dihubungkan ke pin 2 pada Arduino.
- Pin GND pada RS232 dihubungkan ke pin GND pada Arduino.



Gambar 4.2. Rangkaian Koneksi RFID *reader* dengan Arduino

#### 4.2.2 Rangkaian GPRS Shield

*GPRS Shield* digunakan sebagai sebuah ekspansi fitur dari Arduino untuk dapat menerima SMS dari *user*. Jenis *GPRS Shield* yang digunakan dalam perancangan ini sendiri adalah *GPRS Shield* dari SeeedStudio, yang didalamnya menggunakan modul SIM900 yang memiliki konsumsi daya yang cukup rendah. Pin yang digunakan dalam mengkomunikasikan *GPRS Shield* ditunjukkan dalam Gambar 4.3.

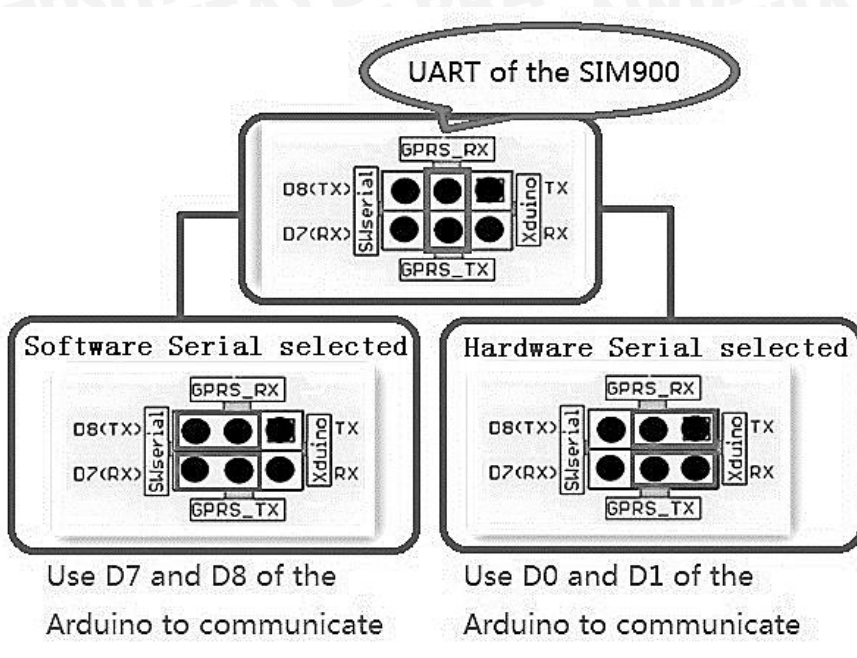


Gambar 4.3 Koneksi pin *GPRS Shield* dengan Arduino

- Pin Digital 0 sebagai *Hardware Serial Penerima* (Rx).
- Pin Digital 1 sebagai *Hardware Serial Pengirim* (Tx).
- Pin Digital 7 sebagai *Software Serial Penerima* (Rx).
- Pin Digital 8 sebagai *Software Serial Pengirim* (Tx).
- Pin Digital 9 sebagai GSM ON atau OFF.

*GPRS Shield* sendiri dapat dikontrol melalui *Hardware serial* ataupun *Software Serial*. Untuk perancangan ini, digunakan pengendalian melalui *Software Serial*, sehingga *GPRS Shield* akan aktif apabila dipicu melalui perangkat lunak. Penggunaan *Software Serial* port ditunjukkan dalam Gambar 4.4.

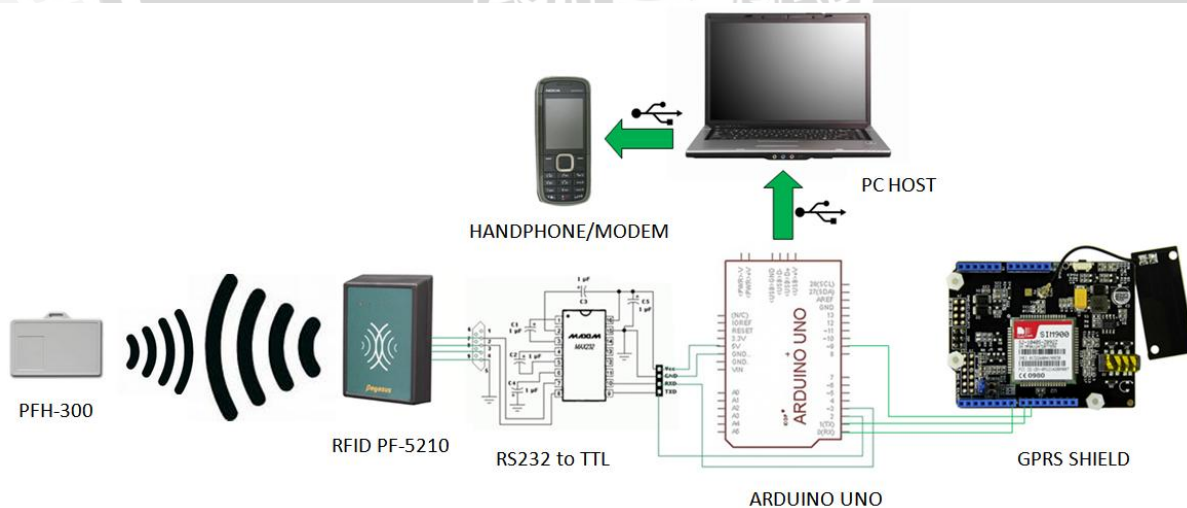




Gambar 4.4 Penggunaan Port *Software Serial* SIM900

### 4.2.3 Perancangan Rangkaian Sistem Keseluruhan

Rangkaian sistem keseluruhan adalah hasil kombinasi antara rangkaian RFID reader, GPRS shield dan handphone atau modem yang sudah dihubungkan ke PC host atau komputer untuk menjadikan sebuah rancangan alat pengecekan mobil menggunakan RFID dengan SMS sebagai media transmisi data. Rangkaian sistem keseluruhan ditunjukkan dalam Gambar 4.5.



Gambar 4.5. Rangkaian Sistem Keseluruhan

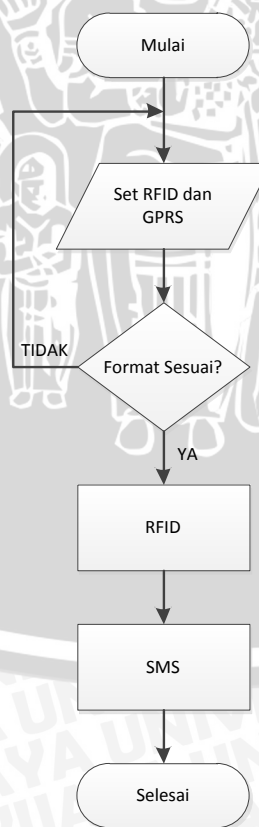
### 4.3 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak, digunakan *software* Arduino IDE dengan versi 1.0.4 untuk pemrograman Arduino Uno dan Microsoft Visual Studio C# 2012 untuk pemrograman antarmuka pada PC.

#### 4.3.1 Perancangan *Flowchart* Program Utama

Perancangan *flowchart* program utama, meliputi tahapan-tahapan atau proses apa saja yang akan dijalankan oleh Arduino. Perancangan ini dapat dipisah menjadi dua sub program, yaitu pembacaan *tag* menggunakan RFID *reader* serta pengolahan SMS yang diterima dari *user* untuk kemudian diproses oleh Arduino. *Flowchart* perancangan program utama secara keseluruhan ditunjukkan dalam Gambar 4.6.

Langkah awal proses alat akan dimulai dengan memastikan baud rate RFID dan GPRS *Shield*. Apabila baud rate yang dimasukkan sesuai, maka proses pembacaan akan dimulai. Proses pembacaan akan terus berjalan dan ketika ada SMS yang masuk maka alat akan memproses pengiriman SMS balasan kembali pada *user*, demikian seterusnya.



Gambar 4.6. Flowchart Perancangan Pada Arduino secara Keseluruhan

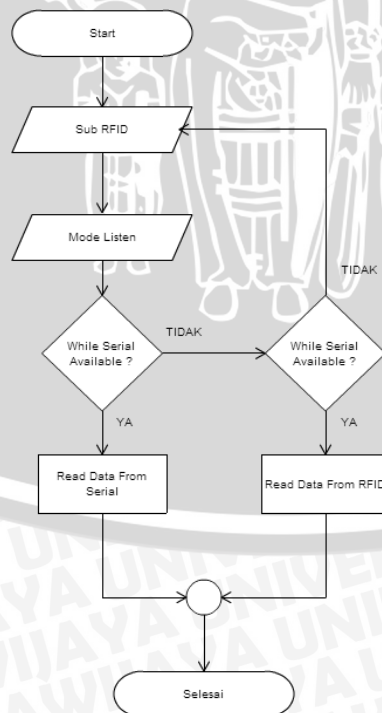


#### 4.3.1.1 Sub Program Pembacaan RFID

Sub program pembacaan *tag* RFID dirancang dengan tujuan membaca adanya *tag* yang berada pada jangkauan sistem, dan disimpan dalam sebuah database. *Flowchart* perancangan pembacaan RFID ini ditunjukkan dalam Gambar 4.7.

Komunikasi antara RFID *reader* PF-5210 dengan Arduino menggunakan komunikasi Serial dengan *baudrate* sebesar 4800 bps (*bit per second*). RFID *reader* ini dihubungkan dengan sistem kontroler melalui port RS-232. Proses pembacaan *tag* pada proses *loop* Arduino akan bergantian dengan proses menunggu SMS. Waktu pembacaan pada masing-masing *tag* yang tidak sama disebabkan oleh karakteristik *reader* RFID PF-5210, yang mana hanya diketahui oleh pihak pembuat.

Pada saat proses pembacaan *tag*, Arduino akan dalam mode listen dan menunggu *tag* yang masuk dalam jangkauan pembacaan *reader*. Jika ada *tag* yang berhasil dibaca “(while (BacaRFID.available() > 0))”, Arduino akan membaca data *tag* dari serial buffer melalui perintah “char inByte = BacaRFID.read()”. Saat berhasil terbaca, LED pada *reader* akan menyala, dan Arduino akan mengirimkan data *tag* ke program antarmuka untuk dicocokkan ke database melalui perintah “Serial.write(inByte)”, dan sistem akan menentukan apakah status *tag* tersebut Masuk atau Keluar.

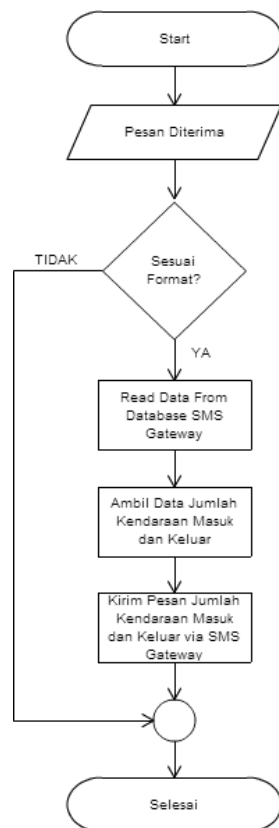


Gambar 4.7. Flowchart Pembacaan RFID

#### 4.3.1.2 Sub Program SMS

Sub program pengiriman dan penerimaan SMS dirancang agar sistem mampu mentransmisikan data dengan jangkauan yang jauh. *Flowchart* pengiriman data melalui SMS ditunjukkan dalam Gambar 4.8. Komunikasi antara GPRS *Shield* dan Arduino menggunakan komunikasi serial UART dengan *baud rate* 19200 bps (*bit per second*), 8 data, 1 stop, *no parity*. Proses memeriksa SMS akan bergantian dengan proses pembacaan *tag*, seperti yang telah disebutkan sebelumnya.

Ketika ada SMS yang masuk ke sistem, maka sistem akan membaca dan mencatat nomor SMS yang masuk tersebut terlebih dahulu melalui perintah “SMSGateway.read()”. Kemudian sistem secara otomatis akan memproses permintaan *user* lebih lanjut. Kemudian akan ada notifikasi “Pesan berhasil dikirim ke nomor” pada layar PC ketika sistem memproses pengiriman SMS balasan. Saat proses tersebut berlangsung, sistem akan sekaligus mengakses informasi jumlah kendaraan yang tersedia. Kemudian SMS berisi jumlah kendaraan akan dikirim melalui perintah “SendSMS(),”Jumlah kendaraan yang tersedia” & iPos1 & “,Jumlah kendaraan yang keluar “ & iPos2 & “.”). Dan bila SMS telah terkirim akan muncul notifikasi apabila SMS telah berhasil dikirim.



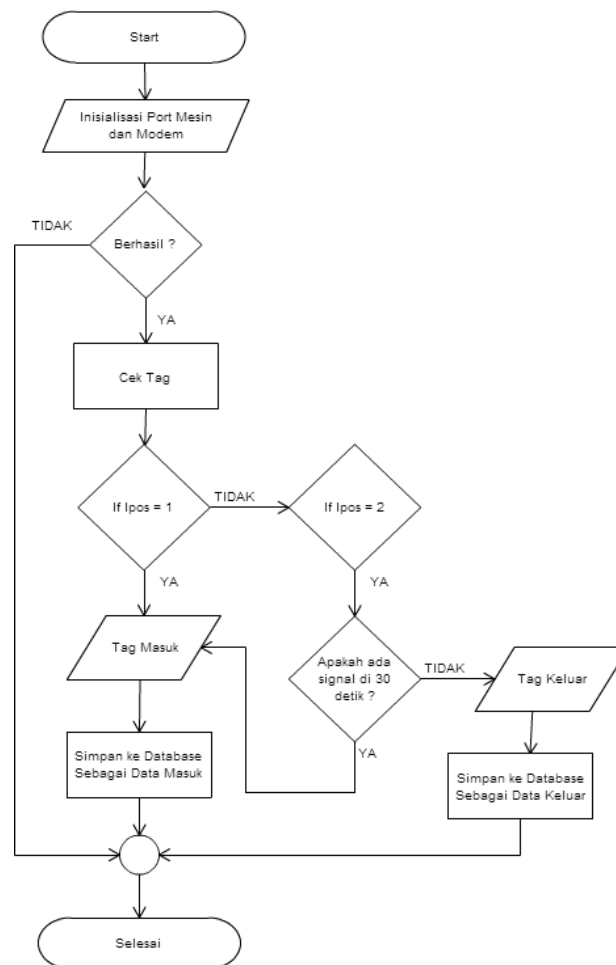
Gambar 4.8. Flowchart SMS Gateway



### 4.3.2 Perancangan *Flowchart* Antarmuka PC

Program antarmuka pada PC ini bertujuan untuk mensinergikan komunikasi antara komputer dengan Arduino. Antarmuka program PC ini terdiri dari nomor *tag*, plat kendaraan, jenis kendaraan, tanggal masuk, tanggal keluar, port komunikasi, serta tombol start, reset, tambah baru dan hapus. Database yang digunakan dalam perancangan ini disimpan dengan menggunakan *filename* *tblMasterMobil*. *Flowchart* keseluruhan dari program ini ditunjukkan dalam Gambar 4.9.

Sebelum memulai menggunakan program ini, *user* harus memilih com port Arduino atau mesin dan com port modem terlebih dahulu, sesuai dengan letak com port Arduino dan modem yang telah ter-*install* pada PC. Di sini telah ditentukan *baudrate* secara otomatis pada 19200 bps (*bit per second*). Untuk memulai menggunakan program, tekan tombol 'Start'.



Gambar 4.9. Flowchart Program PC

Pada Gambar 4.9. mengecek apakah port mesin dan modem sama, misalkan jika port mesin dan modem menggunakan port 7 maka dianggap sama sehingga dapat berkomunikasi antara modem dan mesin. Jika inialisasi port berhasil, langkah selanjutnya adalah cek *tag*. Cek *tag* berfungsi untuk membaca sinyal *tag* yang dipasang pada mobil. Apabila RFID membaca keberadaan *tag*, maka masuk ke ipos 1 dan dianggap sebagai *tag* masuk dan atribut data *tag* id, nomer kendaraan, jenis kendaraan dan tgl atau jam masuk tersimpan ke database. Sebaliknya, apabila RFID sudah tidak menjangkau *tag*, akan masuk ke ipos 2 dilanjutkan dengan mengecek sistem apakah selama 30 detik terdapat sinyal yang masuk dari *tag*. Jika tidak ada, maka dianggap sebagai *tag* keluar artinya mobil sudah meninggalkan area parkir atau keluar. Langkah selanjutnya, data keluar tersebut tersimpan di database dengan atribut data *tag* id, nomer kendaraan, jenis kendaraan dan tgl atau jam keluar.

