

## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

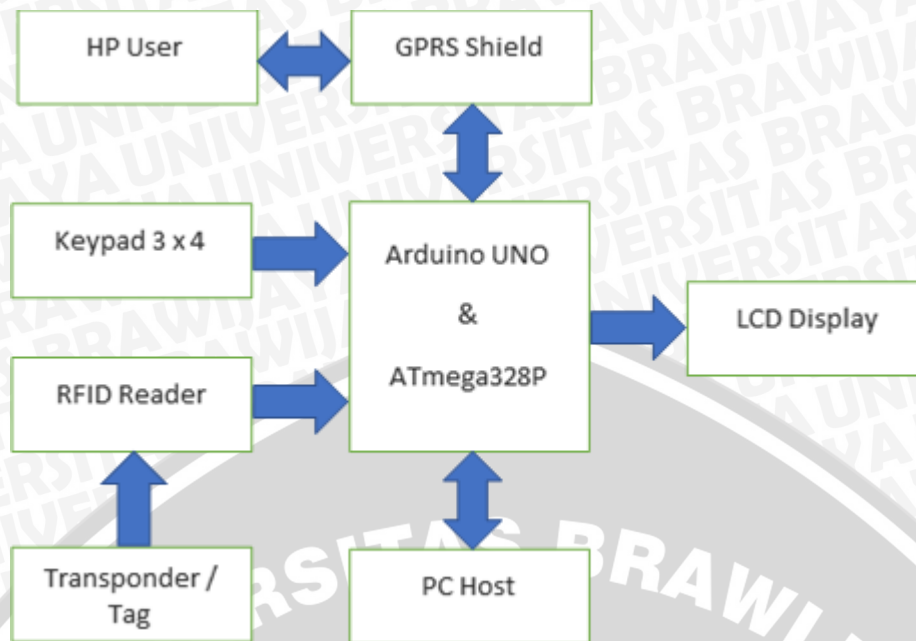
Perancangan Sistem Pemeriksaan Jumlah Ban ini dilakukan secara bertahap sehingga akan memudahkan analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari beberapa hal sebagai berikut :

- 1). Perancangan sistem pemeriksa jumlah ban.
- 2). Perancangan perangkat keras (*hardware*) terdiri dari rangkaian sistem RFID, rangkaian keypad, serta rangkaian LCD.
- 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 memasukkan password melalui keypad. Apabila angka yang dimasukkan sesuai dengan password yang telah disimpan sebelumnya, maka akan ada notifikasi “Dekatkan Tag” pada LCD, yang merupakan proses pendeteksian tag. Apabila ada tag yang didekatkan pada RFID *reader*, maka kode dari tag tersebut akan dimasukkan dan dicocokkan pada database yang disimpan pada PC, untuk menentukan apakah tag tersebut tag masuk atau tag keluar. Proses pendeteksian tag akan berakhir setelah 20 hitungan. Kemudian akan muncul notifikasi “wait : SMS” pada LCD, yang merupakan proses menunggu datangnya SMS. SMS yang masuk akan diterima oleh GPRS *Shield*, dan Arduino akan mengirimkan perintah ke program antarmuka untuk mengakses database dan menghitung jumlah ban yang masih tersimpan. Kemudian GPRS *Shield* akan mengirimkan SMS balasan berisi jumlah ban tersebut kepada *user*.

Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Diagram Blok Sistem

Fungsi masing-masing blok dalam diagram blok diatas adalah sebagai berikut :

- Tag**; dipakai sebagai penanda ban dengan cara ditempelkan. *Tag* akan mengalami perpindahan data (*coupling*) saat didekatkan ke *reader*. Ketika medan gelombang radio dari *reader* didekati oleh *tag*, koil antena yang terdapat pada *tag* pasif ini akan membentuk suatu medan magnet. Medan magnet ini akan menginduksi suatu tegangan listrik yang memberi tenaga pada *tag* pasif. Pada saat yang sama akan terjadi suatu tegangan jatuh pada beban *tag*. Tegangan jatuh ini akan terbaca oleh *reader*.
- RFID Reader**; berfungsi untuk mendeteksi *tag* yang telah ditempelkan pada ban. Kemudian data yang didapat, akan dikirimkan ke Arduino untuk dilakukan pengolahan lebih lanjut.
- Keypad**; sarana bagi user untuk memasukkan password pada alat.
- Arduino**; *board* mikrokontroller yang digunakan untuk menerjemahkan data yang diperoleh dari pembacaan *tag* RFID oleh RFID *reader* serta pengolah aktivitas alat secara keseluruhan.
- PC**; digunakan untuk menampilkan ID *tag* beserta tanggal masuk maupun tanggal keluar melalui sebuah antarmuka yang dibuat menggunakan software Microsoft Visual C#.

- f. **LCD**; digunakan sebagai *visual* pada bagian alat yang akan menampilkan menu bagi *user*.
- g. **GPRS Shield**; dipakai sebagai sarana SMS gateway yang berfungsi menerima perintah dari user ataupun dari mikrokontroler yang berupa AT command untuk menyampaikan informasi.
- h. **HP User**; User cukup mengirimkan sebuah SMS yang berisi perintah kepada alat, untuk kemudian alat akan mengirimkan info mengenai jumlah ban.

## 4.2 Perancangan Perangkat Keras

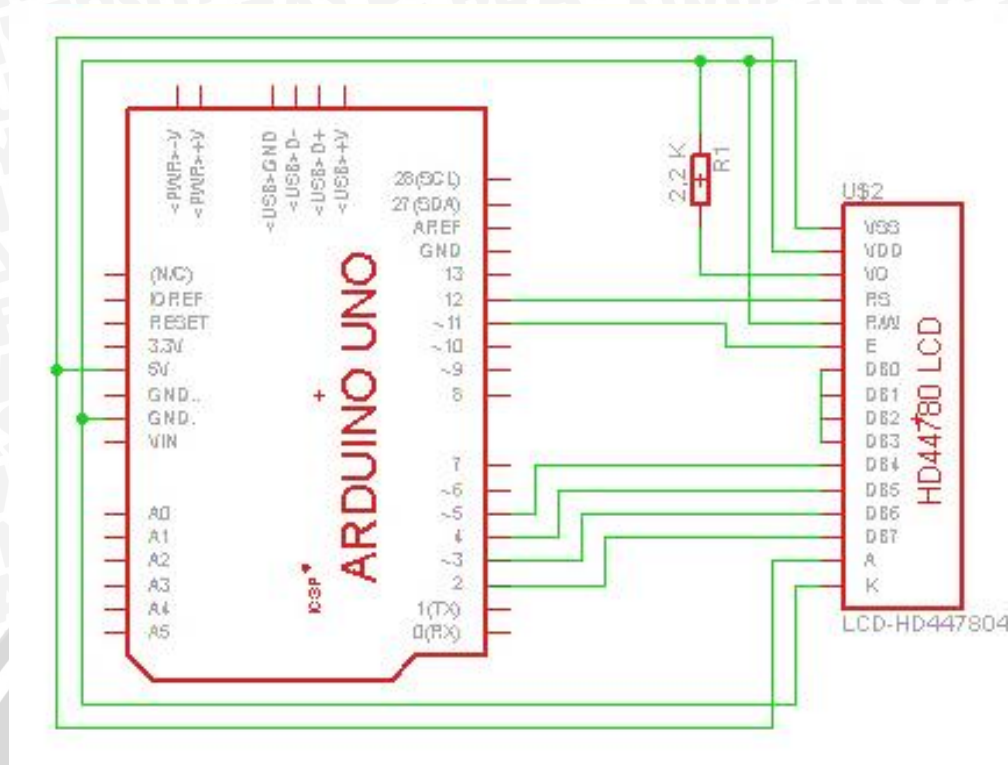
### 4.2.1 Perancangan Rangkaian LCD (*Liquid Crystal Display*) 2 x 16 Karakter

Pada perancangan sistem ini digunakan LCD modul HD44780 yang merupakan sebuah modul LCD *dot matrix* yang membutuhkan daya kecil. LCD modul HD44780 dilengkapi panel LCD dengan tingkat kontras yang cukup tinggi. Rangkaian koneksi LCD karakter 2 x 16 ke Arduino ditunjukkan dalam Gambar 4.2. LCD modul HD44780 mempunyai spesifikasi sebagai berikut :

- Memiliki 16 karakter dan 2 baris tampilan yang terdiri atas 5 x 8 dot matrik.
- Memerlukan catu daya DC 5 V.
- Menggunakan 4 bit data dan 3 bit kontrol.
- *Adjustable contrast*.





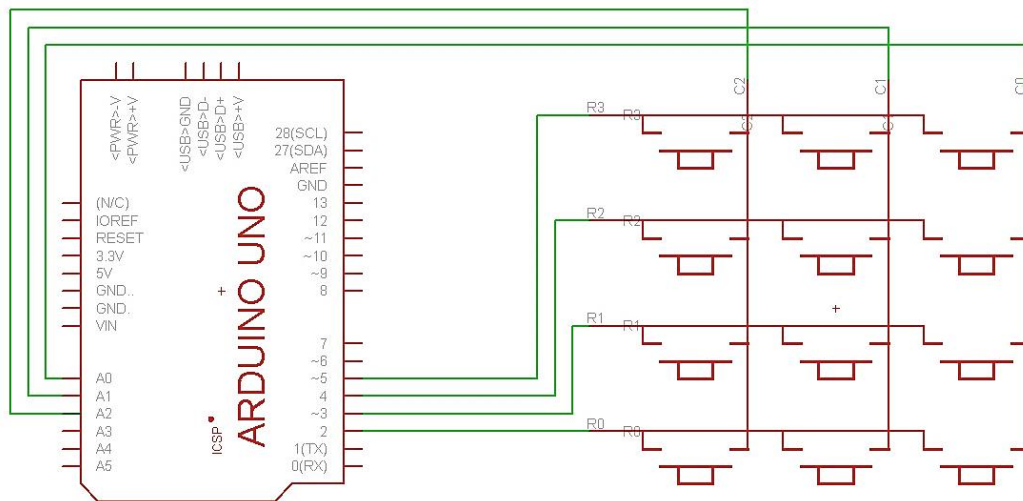


Gambar 4.2. Rangkaian Koneksi LCD 2 x 16 Karakter ke Arduino

Dalam perancangan ini agar didapat kontras dan *backlight* dari LCD yang sesuai, maka dipasang sebuah resistor dengan nilai 2,2 k $\Omega$  pada PIN *contrast* dari LCD 2 x 16 karakter untuk mengatur nilai tegangan yang masuk ke PIN *contrast*.

#### 4.2.2 Perancangan Rangkaian Keypad

Pada perancangan sistem ini, digunakan keypad dengan dimensi 3 x 4. Keypad ini akan difungsikan sebagai sebuah input bagi *user* untuk memasukkan *password* maupun mengakses menu pada alat, melalui tombol-tombol yang terdapat pada papan keypad tersebut. Port untuk baris dihubungkan pada pin 2, 3, 4, 5 Arduino. Sedangkan untuk port kolom dihubungkan pada pin analog Arduino, yaitu pin A0, A1, dan A2, seperti yang terlihat pada Gambar 4.2.



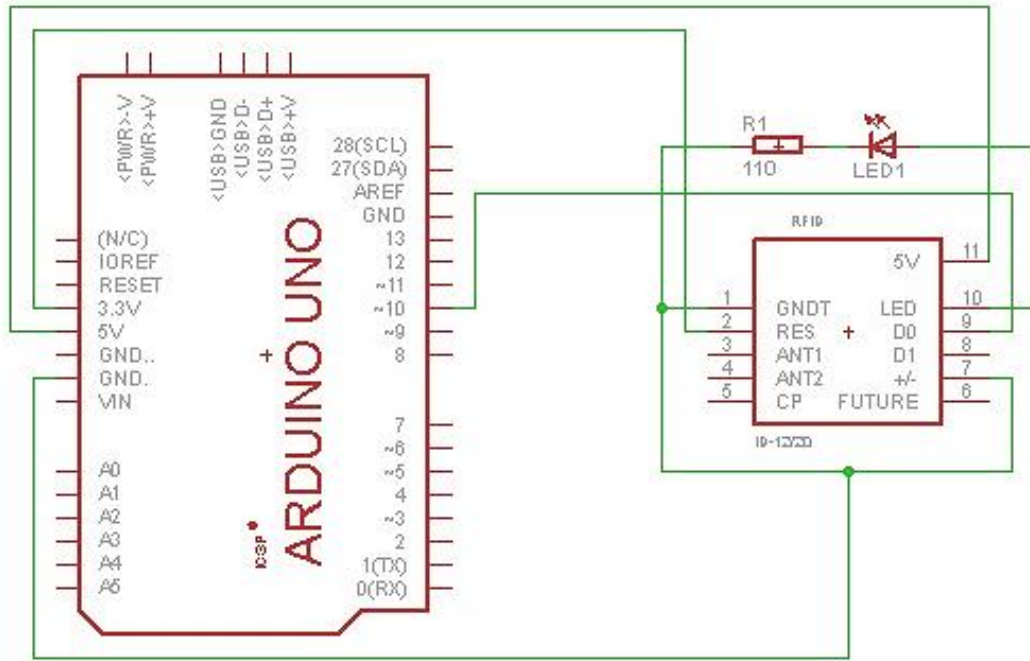
Gambar 4.3. Rangkaian Koneksi Keypad 3 x 4 ke Arduino

#### 4.2.3 Perancangan Rangkaian RFID Reader

Perangkat ini digunakan sebagai pembaca tag yang terdeteksi yang terhubung secara serial dengan komputer. RFID Reader ini nantinya akan mengirimkan data ASCII. Data yang akan dikirimkan tersebut terdiri dari start data (02h), 12 data ASCII (10 data inti dan 2 data checksum) dan stop data (03h).

Rangkaian koneksi antara Arduino dengan RFID Reader ditunjukkan pada Gambar 4.4. Dalam perancangan ini, pin-pin yang harus dihubungkan dengan pin Arduino antara lain :

- Pin 1 dan 7 dihubungkan ke pin GND Arduino
- Pin 2 dapat dihubungkan pada salah satu pin digital I/O ataupun pin 3,3V pada Arduino. Namun untuk perancangan ini, dihubungkan dengan pin 3,3V.
- Pin 9 dihubungkan pada pin 10 digital I/O Arduino.
- Pin 11 dihubungkan dengan Vcc Arduino, 5V.

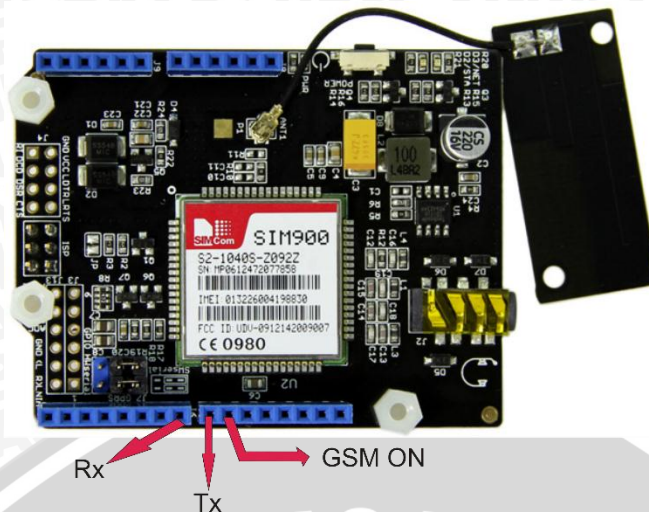


Gambar 4.4. Rangkaian Koneksi RFID Reader dengan Arduino

#### 4.2.4 Rangkaian GPRS Shield

*GPRS Shield* digunakan sebagai sebuah ekspansi fitur dari Arduino untuk dapat menerima serta mengirimkan SMS dari/kembali ke *user*. Jenis GPRS Shield yang digunakan dalam perancangan ini sendiri adalah GPRS Shield dari SeedStudio, yang didalamnya menggunakan modul SIM900 yang memiliki konsumsi daya yang cukup rendah. Pin yang digunakan dalam mengkomunikasikan GPRS Shield ditunjukkan pada Gambar 4.5.

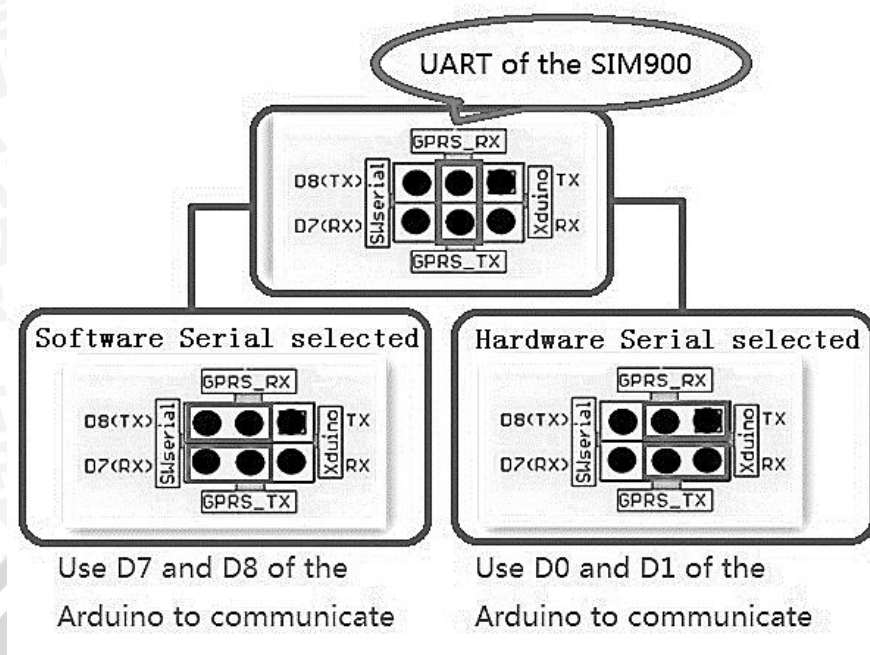




Gambar 4.5 Koneksi pin GPRS Shield dengan Arduino

- 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 Gambar 4.6.



Gambar 4.6 Penggunaan Port Software Serial SIM900

### 4.3 Perancangan Perangkat Lunak

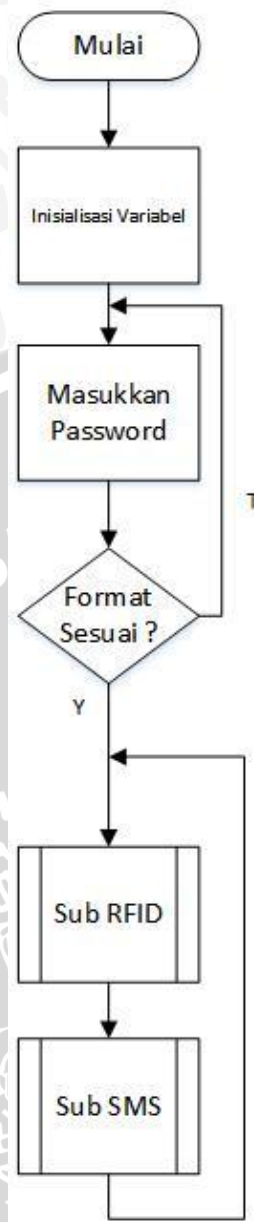
Dalam perancangan perangkat lunak, digunakan *software* Arduino IDE dengan versi 1.0.4 untuk pemrograman Arduino Uno dan Microsoft Visual Studio C# 2010 untuk pemrograman antarmuka pada PC.

#### 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. Gambar 4.7 menunjukkan *flowchart* perancangan program utama secara keseluruhan.

Proses alat akan dimulai dengan proses memasukkan *password*. Apabila *password* yang dimasukkan sesuai, maka proses pendeteksian akan dimulai selama 20 hitungan. Apabila telah menyelesaikan 20 hitungan maka proses menunggu datangnya SMS akan dimulai. Jika tidak ada SMS yang masuk atau alat sudah menyelesaikan proses mengirim SMS balasan kembali pada *user*, maka akan terjadi pengulangan kembali ke proses pendeteksian tag oleh RFID *reader*, demikian seterusnya.





Gambar 4.7. Flowchart Perancangan Pada Arduino secara Keseluruhan

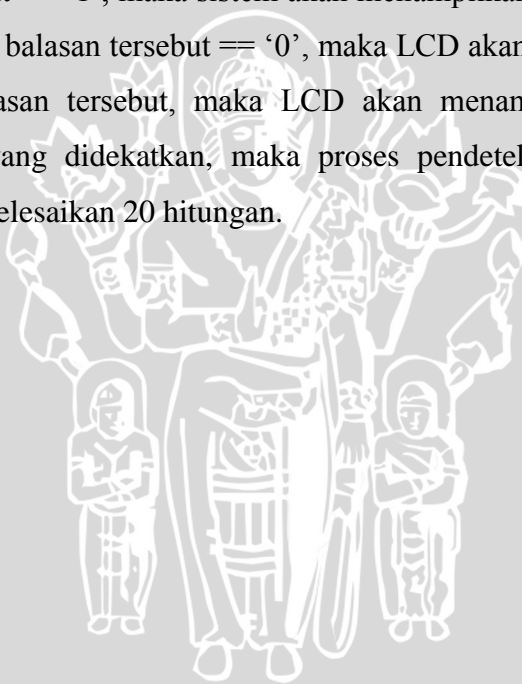
#### 4.3.1.1 Sub Program Pembacaan RFID

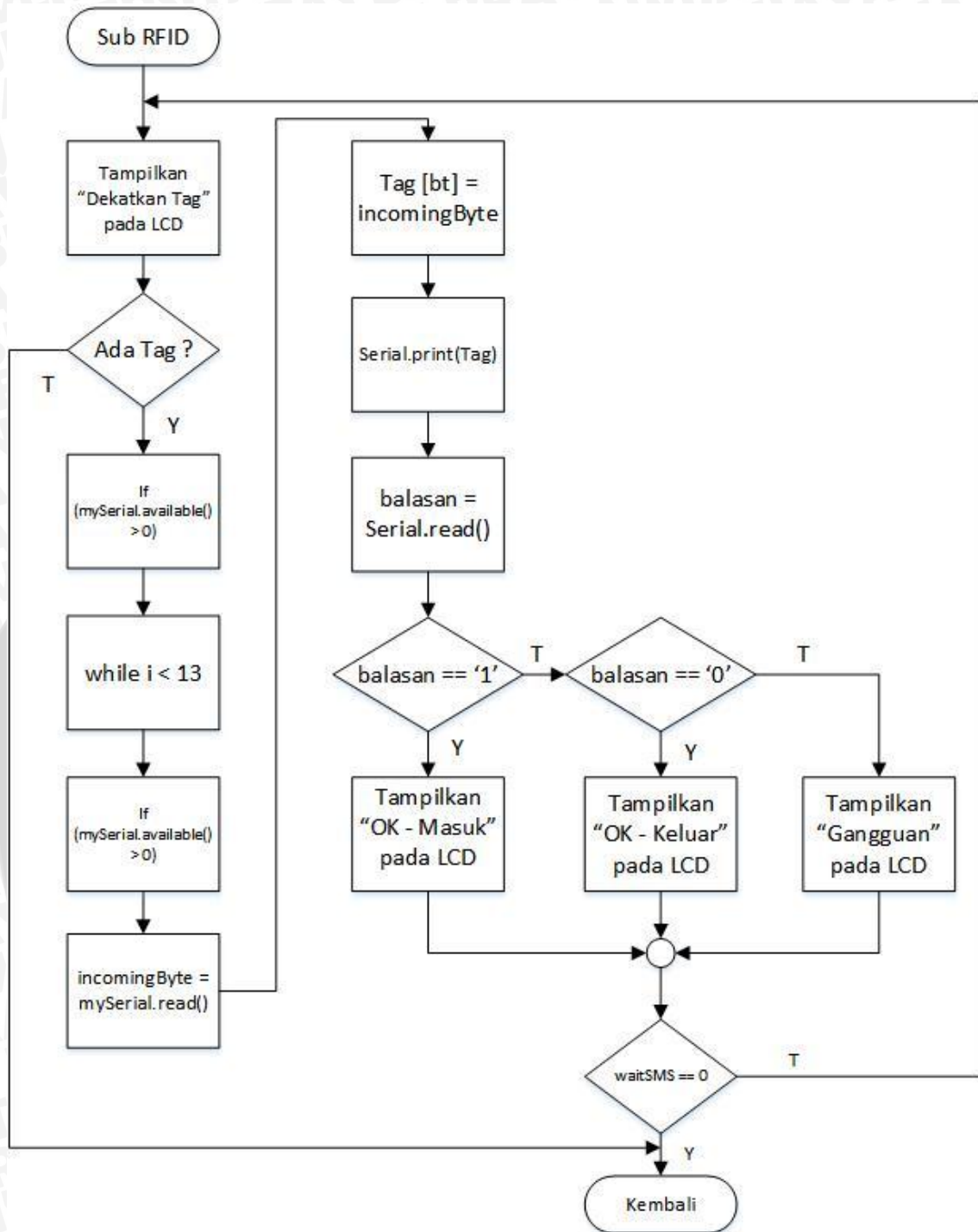
Sub program pembacaan tag RFID dirancang dengan tujuan mendeteksi adanya tag yang didekatkan pada sistem, dan disimpan dalam sebuah database. *Flowchart* perancangan pembacaan RFID ini ditunjukkan pada Gambar 4.8.

Komunikasi antara RFID Reader ID-12 dengan Arduino menggunakan komunikasi Serial dengan *baudrate* sebesar 9600 bps (*bit per second*). RFID reader ini dikontrol menggunakan Software Serial. Proses pendeteksian tag pada proses *loop*

Arduino akan bergantian dengan proses menunggu SMS. Proses pendeteksian akan diberikan waktu selama 20 hitungan. Setelah proses hitungan selesai, maka untuk sementara proses pendeteksian akan di-*postpone*.

Pada saat proses pendeteksian, akan muncul notifikasi pada LCD “Dekatkan Tag”. Jika ada tag yang berhasil terdeteksi “(if(mySerial.available() > 0))”, Arduino akan membaca data tag dari serial buffer melalui perintah “incomingByte = mySerial.read()”. Arduino hanya akan membaca data sebanyak 12 byte saja, dimana data yang tersimpan pada variabel incomingByte tadi akan dimasukkan pada array Tag[]. Saat berhasil terdeteksi, LED akan menyala, dan Arduino akan mengirimkan data tag ke program antarmuka untuk dicocokkan ke database melalui perintah “Serial.print(Tag)”, dan menunggu balasan dari komputer dengan perintah “balasan=Serial.read()”. Dengan pemisalan balasan tersebut == '1', maka sistem akan menampilkan “OK - Masuk” pada LCD. Apabila dimisalkan balasan tersebut == '0', maka LCD akan menampilkan “OK - Keluar”. Selain dua balasan tersebut, maka LCD akan menampilkan “Gangguan”. Apabila tidak ada tag yang didekatkan, maka proses pendeteksian tag akan terus berlangsung hingga menyelesaikan 20 hitungan.





Gambar 4.8. Flowchart Pembacaan RFID

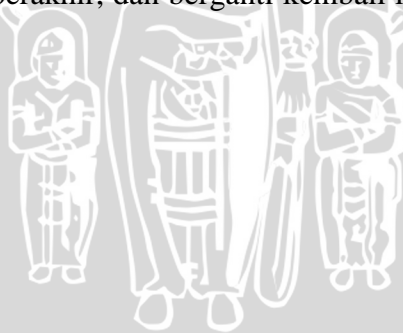
#### 4.3.1.2 Sub Program SMS

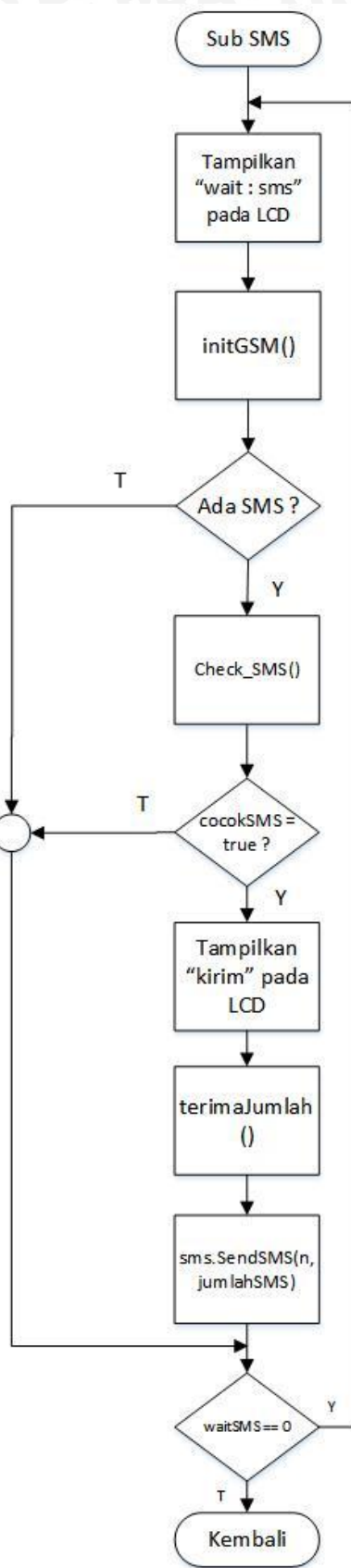
Sub program pengiriman dan penerimaan SMS dirancang agar sistem mampu mentransmisikan data dengan jangkauan yang jauh. *Flowchart* pengiriman data melalui SMS ditunjukkan pada Gambar 4.9.



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 pendeteksian tag, seperti yang telah disebutkan sebelumnya.

Ketika tiba giliran pemeriksaan SMS, akan muncul notifikasi pada LCD berupa “wait : SMS”. Pada tahapan ini, GPRS shield akan melakukan insialisasi melalui perintah “initGSM()”. Jika ada SMS yang masuk ke sistem, maka sistem akan mencatat nomor SMS yang masuk tersebut terlebih dahulu melalui perintah “Check\_SMS()”. Kemudian sistem akan melihat format SMS yang masuk sesuai dengan format yang ada pada sistem atau tidak. Apabila SMS yang dikirimkan *user* memiliki format ‘CHECK’, maka sistem akan memproses permintaan *user* lebih lanjut “(cocokSMS = true)”. Kemudian akan ada notifikasi “kirim” pada LCD ketika sistem memproses pengiriman SMS balasan. Saat proses tersebut berlangsung, Arduino akan menerima informasi dari program antarmuka mengenai jumlah ban yang masih ada, melalui perintah terimaJumlah(). Kemudian SMS berisi jumlah ban akan dikirim melalui perintah “sms.SendSMS(n, JumlahSMS). Notifikasi SMS terkirim akan muncul apabila SMS telah berhasil dikirim. Setelah itu, sistem akan menghapus inbox yang berisi permintaan jumlah tag tadi, agar tidak membebani memori yang ada pada SIM Card. Apabila tidak ada SMS yang masuk, maka proses memeriksa SMS akan berakhir, dan berganti kembali ke proses pendeteksian tag oleh RFID reader.





Gambar 4.9 Flowchart Penerimaan dan Pengiriman SMS

#### 4.3.2 Perancangan *Flowchart* Antarmuka PC

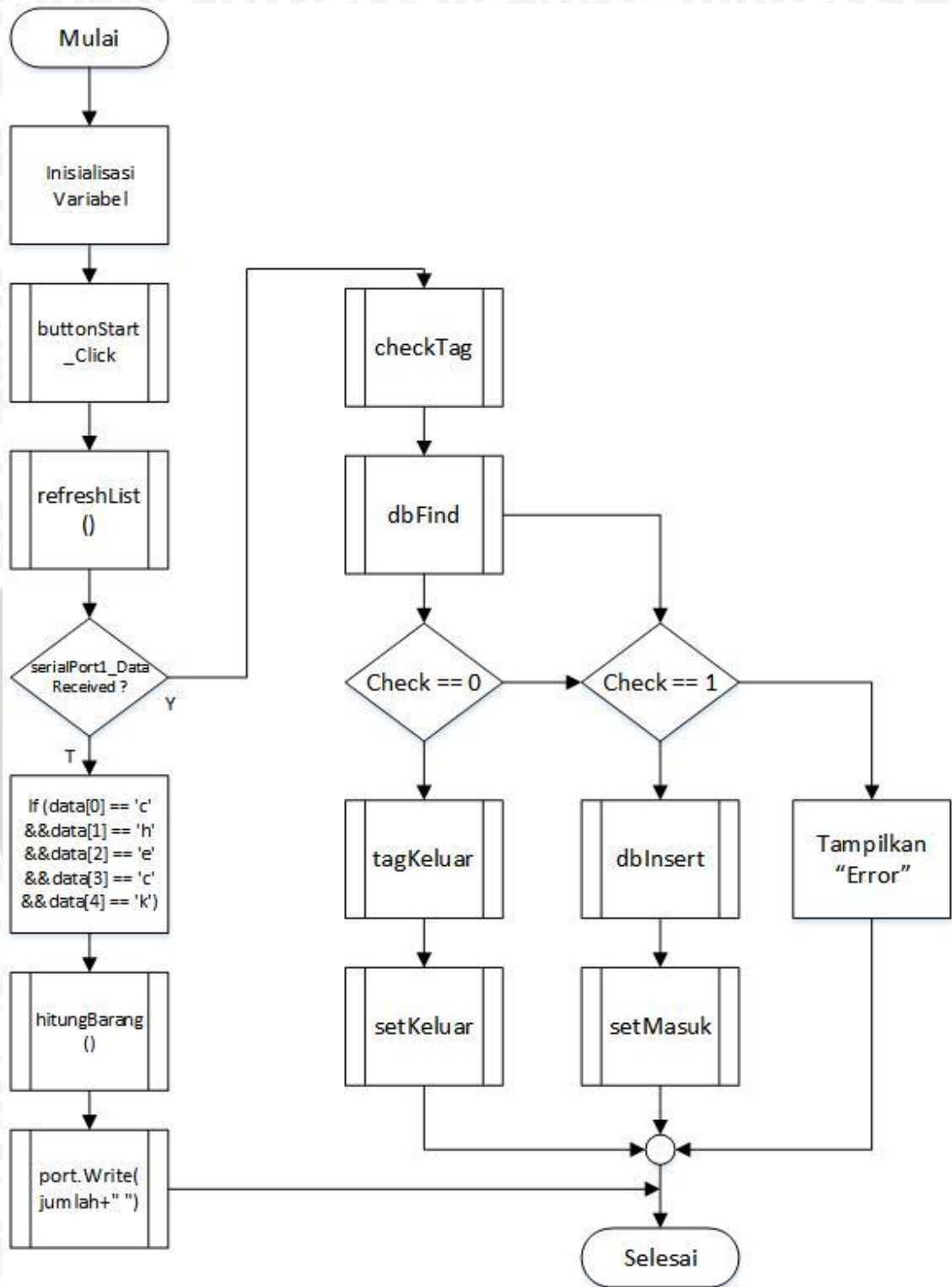
Program antarmuka pada PC ini bertujuan untuk mensinergikan komunikasi antara komputer dengan Arduino. Antarmuka program PC ini terdiri dari tag, tanggal masuk, tanggal keluar, port komunikasi, nomor HP yang ada pada SIM900, serta tombol start, dan stop. Database yang digunakan dalam perancangan ini disimpan dengan menggunakan *filename* db\_tag.mdb dengan *address* D:\Database. *Flowchart* keseluruhan dari program ini dapat dilihat pada Gambar 4.10.

Sebelum memulai menggunakan program ini, *user* harus memilih com port terlebih dahulu, sesuai dengan letak com port Arduino yang telah ter-*install* pada PC. Di sini telah ditentukan *baudrate* secara otomatis pada 9600 bps (*bit per second*). Untuk memulai menggunakan program, tekan tombol ‘Start’.

Apabila terdeteksi sebuah tag, program akan menerima data tag dari Arduino dan memproses data tag tersebut terlebih dahulu (*checkTag*). Program akan mencari dan mencocokkan tag tersebut melalui perintah *dbFind*. Dengan menggunakan pemisalan yang hampir sama, yaitu untuk tag ada tapi belum keluar = 0 dan tag ada sudah keluar = 1, maka pendeteksian tag ini dapat dikelompokkan sesuai dengan kolomnya masing-masing. Apabila tag tersebut adalah baru (*Check == 1*), maka kolom tag dan *tglMasuk* akan terisi dalam database *Tb\_Tag* melalui perintah “(*dbInsert*)”, masing-masing dengan kode beserta tanggal pendeteksian pada waktu kini, dan terdapat label tulisan “MASUK” melalui perintah “(*setMasuk*)” di bawah tulisan tombol “STOP”. Apabila tag tersebut terdeteksi untuk kedua kalinya (*Check == 0*), maka kolom tag, *tglMasuk*, dan *TglKeluar* akan langsung terisi masing-masing kode tag, tanggal tag tersebut pertama kali dikategorikan “baru”, dan tanggal pendeteksian kini, dan database akan diperbaharui (*update*) melalui perintah “*tagKeluar*”. Selain itu, maka akan muncul label “Error”.

Apabila ada SMS yang masuk ke alat, maka Arduino akan mengirim permintaan pada program untuk melihat database. Program akan melihat parameter *TglKeluar*, apakah kosong atau tidak. Jumlah ban yang masih tersimpan dihitung berdasarkan parameter tersebut masih dalam keadaan kosong melalui perintah “*hitungBarang()*”. Setelah itu, program akan kembali mengirimkan informasi berisi jumlah tersebut kembali ke Arduino untuk pengolahan lebih lanjut dengan perintah “*port.Write(jumlah+” “)*”.





Gambar 4.10 Flowchart Program PC