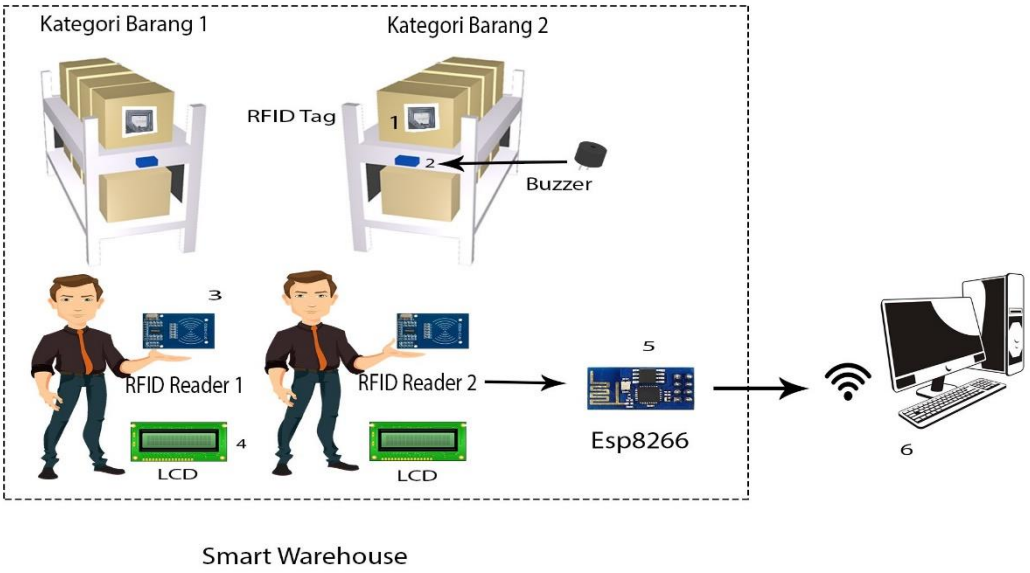
BAB III
PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Sistem *Smart Warehouse* ini memiliki beberapa komponen, setiap komponen saling terhubung dan mengirim informasi dengan komponen lainnya, berikut gambaran umum sistem *Smart Warehouse* menggunakan RFID.



Gambar 3.1 Gambar Umum Sistem

Pada Gambar 3.1 terdapat beberapa kardus yang tersimpan pada sebuah pergudangan dan memiliki kategori barang berbeda, setiap kardus akan ditempelkan RFID *tag*, data pada RFID *tag* dibaca oleh RFID *reader*, dan LCD akan menampilkan nomor uid apakah RFID *tag* sudah terbaca oleh RFID *reader* selanjutnya data akan dikirim ke database melalui media transmisi *wireless* ke sebuah *Personal Computer*. Data yang lebih detail akan ditampilkan oleh admin menggunakan *smartphone* berbasis Android. Buzzer yang tersimpan pada rak barang berfungsi untuk mengetahui letak keberadaan kategori barang yang ada di pergudangan.

Keterangan :

1. RFID *tag*, untuk menandai objek yang ingin dikenali
2. Buzzer digunakan untuk indikator keberadaan barang.
3. RFID *reader*, membaca data yang ada di RFID *tag*

4. LCD untuk menampilkan nomor uid yang pada RFID *tag*
5. Modul Wifi esp8266 sebagai pengirim data dari RFID *tag* ke *Personal Computer*
6. *Personal Computer* menampilkan data dari RFID *tag*

3.2 Komponen Sistem

Terdapat beberapa komponen yang digunakan dalam perancangan dan pembuatan implementasi *Smart Warehouse* Menggunakan RFID, komponen tersebut terbagi atas perangkat lunak dan komponen perangkat keras.

3.2.1 Perangkat Lunak

Pada bagian ini menjelaskan perangkat lunak yang dibutuhkan untuk membangun sistem ini yaitu, Arduino IDE yang berfungsi sebagai Compiler dan Editor Coding Modul Arduino.

3.2.2 Perangkat Keras

Pada bagian ini menjelaskan perangkat keras yang dibutuhkan untuk membangun sistem ini yang terlampir pada Tabel 3.1

Tabel 3.1 Perangkat Keras

No	Perangkat Keras	Jumlah
1	RFID tipe MFRC522	2
2	Mifare RFID Tag	8
3	Arduino Uno	2
4	Modul WiFi esp 8266	2
5	Push Button	2
6	LCD 2x16	2
7.	Arduino nano	1
8.	buzzer	2

3.1 Instalasi Sistem

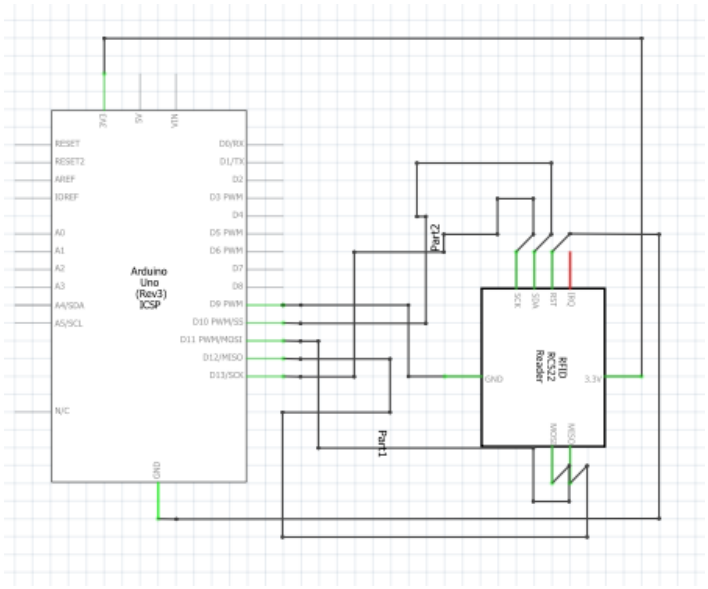
Dalam instalasi sistem diperlukan beberapa tahapan yang akan dilakukan mulai dari konfigurasi Arduino UnoR3, konfigurasi modul WiFi, konfigurasi RFID, konfigurasi LCD dan konfigurasi Buzzer.

3.1.1 Konfigurasi Arduino Uno R3

Konfigurasi Arduino Uno R3 dilakukan menggunakan *software* standar Arduino IDE. Pada sistem yang akan dibuat, Arduino Uno berfungsi sabagai mikrokontroler yang akan menerima data dari RFID *tag* yang dibaca oleh RFID *reader* dan mengirimkannya ke PC *server* melalui media transmisi *wireless* menggunakan modul ESP8266.

3.1.2 Konfigurasi Modul RFID MFRC522 dengan Arduino Uno

Konfigurasi Modul RFID MFRC522 juga dilakukan menggunakan Arduino IDE yang dirangkai bersmaan dengan Arduino Uno, berikut pada gambar 3.2 merupakan konfigurasi rangkaiannya.



Gambar 3.2 Rangkaian Modul RFID MFRC522 dengan Arduino Uno

Konfigurasi pengkabelan pada gambar 3.2 diatas adalah sebagai berikut :

1. Modul RFID pin 3.3V dihubungkan dengan pin 3.3V pada Arduno Uno.
2. Modul RFID pin RST dihubungkan dengan pin 9 pada Arduino Uno.
3. Modul RFID pin GND dihubungkan dengan pin GND pada Arduino Uno.
4. Modul RFID pin MISO dihbungkan dengan pin 12 pada Arduino Uno.

5. Modul RFID pin MOSI dihubungan dengan pin 11 pada Arduino Uno.
6. Modul RFID pin SCK dihubungkan dengan pin 13 pada Arduino Uno.
7. Modul RFID pin SDA dihubungkan dengan pin 10 pada Aduino Uno.
8. Catu Daya pada Arduino uno didapat dari regulator tegangan 5V DC.

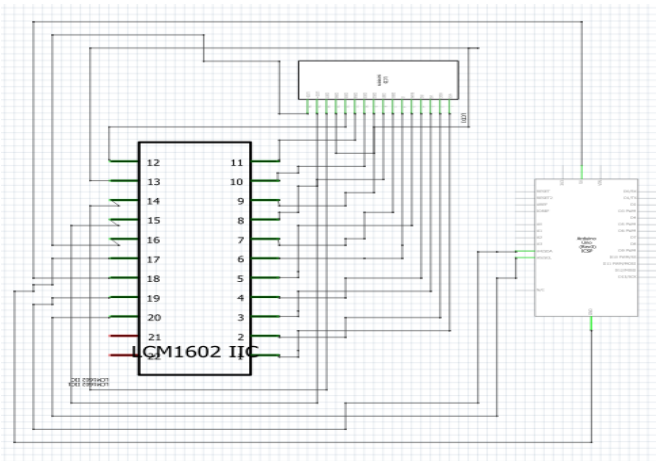
Setelah rangkaian selesai disusun, selanjutnya adalah dilakukan pemrograman agar RFID MFRC522 terhubung dengan Arduino. Berikut potongan *source code Arduino.*

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

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);  //     Create
MFRC522 instance.
void setup() {
       Serial.begin(9600);  // Initialize serial
communications with the PC
       SPI.begin();               //  Init  SPI
bus
       mfrc522.PCD_Init();  // Init MFRC522 card
       Serial.println("Scan PICC to see UID and
type...");
}
void loop() {
       // Look for new cards
       if ( ! mfrc522.PICC_IsNewCardPresent()) {
              return;
       }
       // Select one of the cards
       if ( ! mfrc522.PICC_ReadCardSerial()) {
              return;
       }
}
```

3.1.3 Konfgurasi LCD dengan Arduino Uno

LCD akan dihubungkan dengan arduino uno seperti yang ditunjukan pada gambar 3.4 dibawah ini.



Gambar 3.3 Rangkaian LCD dengan Arduino Uno

Konfigurasi pengkabelan LCD dengan i2C ke Arduino pada gambar 3.3 diatas adalah sebagai berikut :

1. Hubungkan pin VCC pada LCD i2C ke pin 5V Arduino
2. Hubungkan pin GND pada LCD i2C ke pin GND Arduino
3. Hubungkan pin SCL pada LCD i2C ke pin SCL (atau bisa pake pin A5) Arduino
4. Hubungkan pin SDA pada LCD i2C ke pin SDA (atau bisa pake pin A4) Arduino.

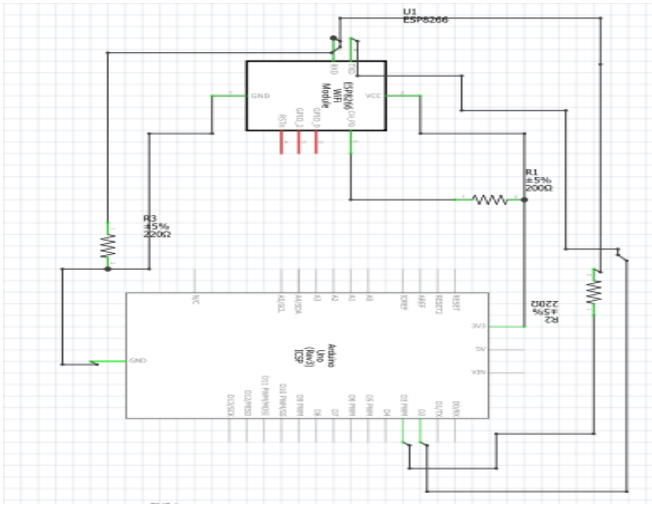
Setelah rangkaian selesai disusun, selanjutnya adalah dilakukan pemrograman agar LCD terhubung dengan Arduino. Berikut potongan *source code Arduino*

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27 ,2,1,0,4,5,6,7,3,
POSITIVE);
//Ubah alamat 0x27 dengan alamat i2C kamu
void setup(){
  lcd.begin (16,2); //LCD untuk ukuran 16x2
  }
```

```
void loop(){
  lcd.setCursor(0, 0); //baris pertama
  lcd.print("TEST LCD i2C");
  lcd.setCursor(0, 1); //baris kedua
  lcd.print("TEST LCD i2C");
}
```

3.1.4 Konfigurasi Esp8266 dengan Arduino Uno

Modul WiFi ESP 8266 dikonfigurasi menggunakan *software* Arduino IDE yang mana modul WiFi dirangkai terlebih dahulu dan terhubung dengan Arduino. Berikut gambaran rangkaian Modul WiFi ESP8266 dan langkah yang harus diperhatikan.



Gambar 3.4 Rangkaian Modul WiFi ESP8266

Beberapa langkah konfigurasi ESP8266 pada gambar 3.4 diatas adalah sebagai berikut :

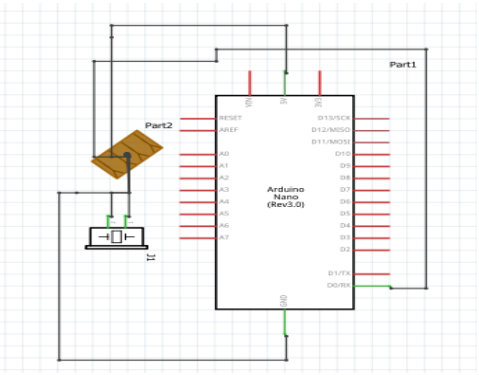
1. Pertama, rangkailah modul WiFi seperti gambar 3.4
2. Kemudian jalankan program Arduino IDE, lalu Upload program kosong ke Arduino.
3. Setelah berhasil melakukan upload program kosong, Buka serial monitor dan set Newline pada pojok kanan bawah menjadi Both NL & CR, dan baudrate menjadi 115200.
4. Terakhir, lakukan konfigurasi modul WiFi menggunakan AT Command ESP 8266

Setelah rangkaian selesai disusun, selanjutnya adalah dilakukan konfigurasi WiFi menggunakan AT command ESP 8266 agar terhubung dengan Arduino.

1. Masukan AT Command ESP8266 melalui *serial* monitor.
2. Melakukan *general test* dengan mengirimkan *command* **“**AT”
3. Melakukan *restart* modul ESP8266 dengan mengirimkan *command* “AT+RST”
4. Mengecek versi *firmware* dengan mengirimkan *command* “AT+GMR”
5. Mengubah mode operasi modul ESP8266 menjadi mode 3 dengan mengirimkan *command* “AT+CWMODE=3”. (1 = Sta, 2 = AP, 3 = keduanya, Sta merupakan mode *default* untuk *router*, AP merupakan mode normal untuk *devices*)
6. Melihat semua *list access point* menggunakan *command* “AT+CWLAP”
7. Melakukan *join* ke *access point* yang diinginkan (misalnya *access point*Hotspot UNIKOM) menggunakan *command* “AT+CWJAP=”Hotspot UNIKOM”, “hotspotunikom”
8. Mendapatkan dan menampilkan IP Address menggunakan *command* “AT+CIFSR”

3.1.5 Konfigurasi Buzzer dengan Arduino nano

Buzzer pada sistem ini berfungsi untuk mencari keberadaan suatu kategori barang yang ada dipergudangan. Konfigurasinya menggunakan *software* Arduino nano yang mana buzzer akan dirangkai terlebih dahulu dan terhubung dengan Arduino. Berikut konfigurasi rangkaianya.



Gambar 3.5 Rangkaian Buzzer

Konfigurasi pengkabelan Buzzer dan Bluetooth ke Arduino pada gambar 3.5 diatas adalah sebagai berikut:

1. Hubungkan Vcc buzzer dan Bluetooth ke pin 5v pada arduino nano.
2. Hubungkan ground buzzer ke gnd arduino nano.
3. Hubungkan Rx pada Bluetooth ke Tx arduino
4. Hubungkan Tx pada Bluetooth ke Rx arduino.

Setelah rangkaian selesai disusun, selanjutnya adalah dilakukan konfigurasi *coding* buzzer dengan arduino.

```
void setup() {
  // initialize serial:
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(11,OUTPUT);
  digitalWrite(13,LOW);
  digitalWrite(11,LOW);
  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}
void loop() {
  // print the string when a newline arrives:
  if (stringComplete) {
    Serial.println(inputString);
    if(inputString=="a9")
    {
      digitalWrite(13,HIGH);  }
    else if(inputString=="b9")
    {
      digitalWrite(13,LOW); }
    else if(inputString=="c9")
    {
      digitalWrite(11,HIGH); }
```

```
    else if(inputString=="d9")
    {
      digitalWrite(11,LOW); }
    // clear the string:
    inputString = "";
    stringComplete = false;
  }
}
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    Serial.print(inChar);
    inputString += inChar;
    if (inChar == '9') {
      stringComplete = true;
    }
  }
```

Pada konfigurasi *coding* buzzer dengan arduino pada a9 untuk mengaktifkan buzzer pada kategori barang 1 dan b9 untuk mematikan bunyi buzzer pada kategori 1, sedangkan untuk *codingan* c9 mengaktifkan bunyi buzzer pada kategori barang 2 dan d9 untuk mematikan bunyi buzzer pada kategori barang 2.