

DAFTAR PUSTAKA

- Abarca, R. M. (2021). Sistem Mikro Kontroler. *Nuevos Sistemas de Comunicación e Información*, 2013–2015.
- Abdalla, M. M., Darjat, D., & Zahra, A. A. (2015). RFID (*Radio Frequency Identification*) sebagai Teknologi Pendukung Pemilu Digital Legislatif. *Transient: Jurnal Ilmiah Teknik Elektro*, 4(3), 794–801.
- Bawono, M. R., Sipahutar, P. Y. C., Sari, N. A., & Natasha, F. (2017). Sistem Keamanan Pada Pintu Menggunakan *Keypad* Dengan Sensor Berbasis Mikrokontroler. *Researchgate.Net*, January.
- Efendi, Y. (2018). *Internet of Things (IoT)* Sistem Pengendalian Lampu Menggunakan *Raspberry Pi* Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer*, 4(2), 21–27.
- Eka Wardani, L. (2011). Prototipe Pemberian Pakan Ayam Berbasis Arduino. *NASPA Journal*, 42(4), 1.
- Firly, firly, Dewi, I. P., Mursyida, L., & Samala, A. D. (2018). Dasar-dasar Android Studio dan Membuat Aplikasi Mobile Sederhana. *Angewandte Chemie International Edition*, 6(11), 951–952., 10, 1–21.
- Huda, R. S., & Kurniawan, A. F. (2022). Penerapan Firebase Pada Aplikasi E-Wisata Berbasis Android. 1, 247–256.
- Kemenko. (2017). *Pedoman Umum Subsidi Rastra*.
- Kustiawan, E. (n.d.). Meningkatkan Efisiensi Peralatan Dengan Menggunakan *Solid State Relay (Ssr)* dalam Pengaturan Suhu Pack Pre-Heating Oven (Pho) (*Studi Kasus di PT Indonesia Toray Synthetics , Tangerang*). 9(1), 1–6.
- Mukti, A. A., Siswoko, S., & Subiyantoro, S. (2022). Sistem Akses Kontrol Kunci Elektrik Menggunakan Pembacaan E-Ktp Berbasis Mikrokontroler. *Jurnal Elektronika Dan Otomasi Industri*, 9(3), 270.
- Natsir, M., Rendra, D. B., & Anggara, A. D. Y. (2019). Implementasi *IoT* untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya. *Jurnal PROSISKO (Pengembangan Riset Dan Observasi Rekayasa Sistem Komputer)*, 6(1), 69–72.
- Prasetya, V. A., Piarsa, I. N., & Sri Arsa, D. M. (2021). Rancang Bangun Prototipe Sistem *Monitoring Vending Machine* Berbasis *Internet of Things*. *Jusikom : Jurnal Sistem Komputer Musirawas*, 6(1), 9–22.

- Purnomo, A. (2015). Perancangan dan Pembuatan Mesin Penjual Makanan Otomatis Menggunakan Relay Cerdas. *Fakultas Teknik Jurusan Teknik Elektro Universitas Muhammadiyah Surakarta*.
- Rahman, K. M., & Hiti, S. (2005). *Synchronous Motor*. 41(2), 557–565.
- Sibagariang, B. A. A. (2018). *Tinjauan Yuridis Pendistribusian Beras Miskin (RASKIN) Menurut Undang-Undang Nomor 7 Tahun 1996 Tentang Pangan Fakultas Hukum Universitas Medan Area Medan*.
- Syahriel, S., Lubis, A. P., & Fauziah, R. (2021). Perancangan Atm Raskin Berbasis Rfid dan *Internet of Things (Iot)* untuk Masyarakat Tidak Mampu. *Journal of Computer*, 1(3), 153–158.
- Tullah, M. H., Amran, T. G., & Sugiarto, D. (2016). Rancang Bangun Sistem Informasi Atm Beras Raskin dengan Menggunakan *Radio Frequency Identification (Rfid)*. *Jurnal Teknik Industri*, 6(2), 165–177.
- Yusniati. (2018). Penggunaan Sensor *Infrared Switching* Pada Motor DC Satu Phasa. *Journal of Electrical Technology*, 3(3), 90–96.

**L
A
M
P
I
R
A
N**

Lampiran 1 Source Code Program

Mesin Vending

Kode inisialisasi library

```
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include "time.h"
#include <EEPROM.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
```

Kode konfigurasi pin sensor dan akuator

```
#define IR 34
#define motor 5
#define buzzer 13
#define SS_PIN 15
#define RST_PIN 2
```

Kode perintah aktifkan RFID dan setting jumlah karakter LCD

```
MFRC522 mfrc522(SS_PIN, RST_PIN);

LiquidCrystal_I2C lcd(0x27,20,4);
```

Kode mengkoneksikan jaringan Wi-Fi

```
#define WIFI_SSID "aynun"
#define WIFI_PASSWORD "namakuji"
```

Kode koneksi ke firebase database

```
#define FIREBASE_HOST "https://mesin-vending-default-rtddb.firebaseio.com/"
#define FIREBASE_AUTH
"6HEawSQ3vKkizxGPdwiAZB6itld5KAv3X5mRw6uf"
```

Kode setting karakter keypad

```
#define ROW_NUM 4
#define COLUMN_NUM 3

char keys[ROW_NUM][COLUMN_NUM] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
```

Kode inisialisasi pin dan perintah aktifkan *keypad*

```
byte pin_rows[ROW_NUM] = {4, 14, 27, 26};
byte pin_column[COLUMN_NUM] = {25, 33, 32};

Keypad keypad = Keypad( makeKeymap(keys), pin_rows,
pin_column, ROW_NUM, COLUMN_NUM );
```

Kode menentukan alamat akses waktu secara online

```
const char* ntpServer = "id.pool.ntp.org";
const long  gmtoffset_sec = 28800;
const int   daylightOffset_sec = 0;
```

Kode variabel yang digunakan

```
char key;
String input_pin;

String tag_id;
String current_date, current_time;
unsigned int ts_counter = 0, ts_counter_success = 0;
int total_rice, rice_final;

//mengatur nama bulan
String months[12]={"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
"Aug", "Sep", "Oct", "Nov", "Dec"};

FirebaseData fbdo;
```

Kode memulai komunikasi SPI pada ESP32

```
void setup() {
  Serial.begin(115200);
  SPI.begin();
  mfrc522.PCD_Init();
  input_pin.reserve(32);

  //mengaktifkan memory EEPROM ESP32
  EEPROM.begin(4096);
```

Kode konfigurasi input dan output, mengaktifkan LCD dan menyalakan backlight, serta mengatur tulisan pada LCD

```
pinMode(IR, INPUT);
pinMode(motor, OUTPUT); digitalWrite(motor, HIGH);
pinMode(buzzer, OUTPUT); digitalWrite(buzzer, LOW);
lcd.init();
lcd.backlight();

lcd.setCursor(0,0);
lcd.print(" MESIN VENDING ");
delay(500);
lcd.clear();
```

Kode koneksi ke Wi-Fi

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
int counterWifi = 0;
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{ counterWifi++;
  Serial.print(".");
  lcd.setCursor(0,1); lcd.print("WAITING.....");
  lcd.setCursor(0,2); lcd.print("CONNECTING TO SERVER");
  delay(500);
  if (counterWifi == 60) {break;}
}

Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
// Baca Counter Transaksi
ts_counter = EEPROM.read(1);
ts_counter_success = EEPROM.read(2);
```

Kode untuk mendapatkan waktu dan memulai koneksi ke firebase database

```
// Init and get the time
configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

//mendapatkan waktu
get_time();
lcd.clear();
```

NTP

Kode untuk mendapatkan nilai waktu realtime

```
void get_time(){
    struct tm timeinfo;
    if(!getLocalTime(&timeinfo)){
        Serial.println("Failed to obtain time");
        return;
    }
}
```

```
// Tanggal
char now_date[3];
strftime(now_date, 3, "%d", &timeinfo);
//Serial.println(now_date);

// bulan
char now_month[10];
strftime(now_month, 10, "%B", &timeinfo);
String abb_month = String(now_month);
if(abb_month=="Januari")abb_month="Jan";
if(abb_month=="Februari")abb_month="Feb";
if(abb_month=="Maret")abb_month="Mar";
if(abb_month=="April")abb_month="Apr";
if(abb_month=="Mei")abb_month="May";
if(abb_month=="Juni")abb_month="Jun";
if(abb_month=="Juli")abb_month="Jul";
if(abb_month=="Agustus")abb_month="Aug";
if(abb_month=="September")abb_month="Sep";
if(abb_month=="Oktober")abb_month="Oct";
if(abb_month=="November")abb_month="Nov";
if(now_month=="Desember")abb_month="Dec";
//Serial.println(now_month);

// Tahun
char now_year[5];
strftime(now_year, 5, "%Y", &timeinfo);
//Serial.println(now_year);

// Format ke tanggal lengkap
current_date = String(now_date) + " " + abb_month + " " +
String(now_year);
Serial.println(current_date);
```

```
// Jam
char now_hour[3];
strftime(now_hour, 3, "%H", &timeinfo);
//Serial.println(now_hour);

// Menit
char now_minute[3];
strftime(now_minute, 3, "%M", &timeinfo);
//Serial.println(now_minute);

// Format ke Waktu HH:mm
current_time = String(now_hour) + ":" + String(now_minute);
Serial.println(current_time);
```


Kode membaca aksi dari database

```

/** Baca Aksi dari database */
action_rw:
lcd.setCursor(0,0); lcd.print(" SELAMAT DATANG ");
lcd.setCursor(0,1); lcd.print(" DI MESIN VENDING ");
lcd.setCursor(0,3); lcd.print(" SCAN E-KTP ANDA! ");

if(WiFi.status() != WL_CONNECTED) {goto manual_mode;}

Firebase.getString(fbdo, "action/action_rw");
if(fbdo.stringData() == "read") {
  Serial.println(fbdo.stringData());
  goto read_tag;
}
if(fbdo.stringData() == "write") {
Serial.println(fbdo.stringData());
  goto read_tag;
}
if(fbdo.stringData() == "delete") {
  Serial.println(fbdo.stringData());
  lcd.clear();
  goto delete_counter;
}
goto action_rw;

// Delete Counter
delete_counter:
ts_counter=0; ts_counter_success=0;
EEPROM.write(1, ts_counter); EEPROM.commit();
EEPROM.write(2, ts_counter_success); EEPROM.commit();
Firebase.setString(fbdo, "action/action_rw", "read");
goto action_rw;

```

Kode untuk memasukkan pengguna baru ke firebase

```

// Masukkan pengguna baru ke database
write_action:
//Cek pengguna terdaftar atau tidak
lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN ");
lcd.setCursor(0, 1); lcd.print("ID : ");
lcd.setCursor(6, 1); lcd.print(tag_id);
lcd.setCursor(0, 2); lcd.print("DAFTAR PENGGUNA BARU");
for(int i=0; i<=19; i++){
  lcd.setCursor(i, 3); lcd.print("."); delay(100);
}

Firebase.setString(fbdo, "user/" + tag_id + "/id", tag_id);
Firebase.setString(fbdo, "user/" + tag_id + "/status", "Pengguna
baru");

```

```

Serial.print("Pengguna Berhasil didaftar dengan id: ");
Serial.println(tag_id);
tag_id = "";
Firebase.setString(fbdo, "action/action_rw", "read");
lcd.clear();
goto action_rw;

```

Kode sensor membaca kode kartu

```

// Baca (kode kartu) UID Tag
read_tag:
if(!mfr522.PICC_IsNewCardPresent()) { return; }
if(!mfr522.PICC_ReadCardSerial()) { return; }
Serial.print("UID tag : ");
String user_id;

for(byte i = 0; i < mfr522.uid.size; i++) {
    user_id.concat(String(mfr522.uid.uidByte[i], HEX));
}
tag_id = user_id;
Serial.println(tag_id);
if(tag_id){
    alarmSukses();
    lcd.setCursor(0, 0); lcd.print("  MESIN VENDING  ");
    lcd.setCursor(0, 1); lcd.print("  SILAHKAN TUNGGU  ");
    lcd.setCursor(0, 2); lcd.print("  MEMBACA ID E-KTP  ");
    lcd.setCursor(0, 3); lcd.print("                ");
    for(int i=0; i<=19; i++){
        lcd.setCursor(i, 3); lcd.print("."); delay(100);
    }
    lcd.clear();
    goto read_user;
}else{
    goto action_rw;}
goto read_tag;

```

Kode membaca pengguna dari database, id terdaftar atau tidak terdaftar

```

// Baca Pengguna dari database, apabila ID terdaftar atau tidak terdaftar
read_user:
Firebase.getString(fbdo, "action/action_rw");
if(fbdo.stringData() == "read") {
    if(Firebase.getString(fbdo, "user/" + tag_id + "/id")){
        if(fbdo.stringData() == tag_id){
            lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN  ");
            lcd.setCursor(0, 1); lcd.print("ID : ");
            lcd.setCursor(6, 1); lcd.print(tag_id);

```

```

delay(1000);
lcd.setCursor(0, 3); lcd.print("  ID TERDAFTAR  ");

Serial.print("Pengguna Terdaftar : ");
Serial.println(fbdo.stringData());
Serial.println("> Masukkan PIN");
delay(2000); lcd.clear();
goto insert_pin;
}else{
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN  ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  delay(1000);
  lcd.setCursor(0, 3); lcd.print(" ID TIDAK TERDAFTAR ");
  Serial.println("Pengguna Tidak tidak terdaftar");
  alarmGagal();
  delay(2000); lcd.clear();
  goto action_rw;
}
}
}
if(fbdo.stringData() == "write") {
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN  ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 2); lcd.print("CEK PENGGUNA  ");
  for(int i=0; i<=19; i++){
    lcd.setCursor(i, 3); lcd.print("."); delay(100);
  }
  //lcd.clear();
  Serial.println("Cek pengguna....");
  goto check_user;
}
goto read_user;

// Cek Pengguna terdaftar/tidak
check_user:
if(Firebase.getString(fbdo, "user/" + tag_id + "/id")){
  if(fbdo.stringData() == tag_id){
    lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN  ");
    lcd.setCursor(0, 1); lcd.print("ID : ");
    lcd.setCursor(6, 1); lcd.print(tag_id);
    lcd.setCursor(0, 2); lcd.print("TERDAFTAR  ");
    delay(3000);
    lcd.clear();
    Serial.print("Pengguna telah terdaftar");
    Firebase.setString(fbdo, "action/action_rw", "read");
    goto action_rw;
  }
}

```

```

}else{
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 2); lcd.print("BELUM TERDAFTAR   ");
  delay(3000);
  lcd.clear();
  Serial.print("Mendaftarkan pengguna baru");
  goto write_action;
}
}
goto check_user;

```

Kode untuk memasukkan PIN

```

// Masukkan PIN
insert_pin:
  key = keypad.getKey();

  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 2); lcd.print("PIN : ");
  lcd.setCursor(6, 2); lcd.print(input_pin);

  if(key){
    digitalWrite(buzzer,1);delay(100);digitalWrite(buzzer,0);
    Serial.println(key);
    if(key == '*') {
      input_pin = ""; // reset input keypad
      lcd.setCursor(6,2); lcd.print("          ");
    }else if(key=='#'){
      //lcd.setCursor(6,0); lcd.print("*****");
      for(int i=0; i<=19; i++){
        lcd.setCursor(i, 3); lcd.print("."); delay(100);
      }
      goto read_pin;
    }else{
      input_pin += key;
    }
  }

goto insert_pin;

```

Kode verifikasi PIN dari database

```

// Verifikasi PIN
read_pin:
if(Firebase.getString(fbdo, "user/" + tag_id + "/pin")){
  Serial.print("PIN : ");
  Serial.println(fbdo.stringData());
  if(fbdo.stringData() == input_pin){
    alarmSukses();
    lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
    lcd.setCursor(0, 1); lcd.print("ID : ");
    lcd.setCursor(6, 1); lcd.print(tag_id);
    lcd.setCursor(0, 2); lcd.print("PIN : ");
    lcd.setCursor(6, 2); lcd.print(input_pin);
    lcd.setCursor(0, 3); lcd.print("  PIN BENAR   ");
    delay(2000);
    lcd.clear();
    Serial.println("PIN Benar");
    input_pin = "";
    goto read_status;
  }else{
    alarmGagal();
    lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
    lcd.setCursor(0, 1); lcd.print("ID : ");
    lcd.setCursor(6, 1); lcd.print(tag_id);
    lcd.setCursor(0, 2); lcd.print("PIN : ");
    lcd.setCursor(6, 2); lcd.print(input_pin);
    lcd.setCursor(0, 3); lcd.print("  PIN SALAH   ");
    delay(2000);
    lcd.clear();
    Serial.println("PIN Salah");
    tag_id = "";
    input_pin = "";
    goto action_rw;
  }
}
}
goto read_pin;

```

Kode membaca status pengambilan beras

```
// Baca status pengambilan beras
read_status:
if(Firebase.getString(fbdo, "user/" + tag_id + "/status")){
    if(fbdo.stringData() == "Belum mengambil beras"){
        goto read_total_rice;
    }
    if(fbdo.stringData() == "Telah mengambil beras"){
        goto process_failed;
    }
    if(fbdo.stringData() == "Pengguna baru"){
        goto new_user;
    }
}
goto read_status;
```

Kode membaca total beras

```
read_total_rice:
if(Firebase.getInt(fbdo, "action/total-rice")){
    if(fbdo.to<int>() >= 1){
        Serial.println("Proses : Beras Tersedia");
        goto process_successful;
    } else {
        Serial.println("Proses : Beras Habis");
        goto process_failed;
    }
}
goto read_total_rice;
```

Kode membaca transaksi sukses dan dikirim ke firebase

```
// Transaksi Berhasil **/
process_successful:
  alarmSukses();
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 3); lcd.print("TRANSAKSI SUKSES....");
  //auth_success();
  delay(2000);

  if(Firebase.getInt(fbdo, "action/total-rice")){
    total_rice = fbdo.to<int>();
    rice_final = total_rice - 1;
    Firebase.setInt(fbdo, "action/total-rice", rice_final);
  }

  Firebase.setString(fbdo, "user/" + tag_id + "/status", "Telah
mengambil beras");
  // Counter Transaksi dan simpan ke EEPROM
  ts_counter++; ts_counter_success++; get_time();
  EEPROM.write(1, ts_counter); EEPROM.commit();
  EEPROM.write(2, ts_counter_success); EEPROM.commit();
  // Mengambil nama pengguna
  if(Firebase.getString(fbdo, "user/" + tag_id + "/nama")){
    String username = fbdo.stringData();
    // Kirim data Transaksi
    FirebaseJson json;
    json.add("id", tag_id);
    json.add("nama", username);
    json.add("jam", current_time);
    json.add("tanggal", current_date);
    json.add("status", "Sukses");
    Firebase.setJSON(fbdo, "transaction/" + String(ts_counter), json);

    FirebaseJson json_sf;
    json_sf.add("id", tag_id);
    json_sf.add("nama", username);
    json_sf.add("jam", current_time);
    json_sf.add("tanggal", current_date);
    Firebase.setJSON(fbdo, "transaction-success/" +
String(ts_counter_success), json_sf);
    motorON();
    goto action_rw;
  }
  goto process_successful;
```

Kode membaca transaksi yang dilakukan saat status pengambilan “telah mengambil beras” dan dikirim ke firebase

```
process_failed:
  alarmGagal();
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 3); lcd.print("SILAHKAN COBA LAGI..");
  //auth_failed();
  // Counter Transaksi dan simpan ke EEPROM
  ts_counter++; get_time();
  EEPROM.write(1, ts_counter); EEPROM.commit();
  // Mengambil nama pengguna
  if(Firebase.getString(fbdo, "user/" + tag_id + "/nama")){
    String username = fbdo.stringData();
    // Kirim data Transaksi
    FirebaseJson json;
    json.add("id", tag_id);
    json.add("nama", username);
    json.add("jam", current_time);
    json.add("tanggal", current_date);
    json.add("status", "Telah mengambil beras");
    Firebase.setJSON(fbdo, "transaction/" + String(ts_counter), json);

    Serial.println("Proses : Gagal Mengambil Beras");
    delay(2000); lcd.clear();
    goto action_rw;
  }
  goto process_failed;
```

Kode perintah aktivasi kartu

```
new_user:
  alarmAktivasi();
  lcd.setCursor(0, 0); lcd.print("PENGGUNA MESIN   ");
  lcd.setCursor(0, 1); lcd.print("ID : ");
  lcd.setCursor(6, 1); lcd.print(tag_id);
  lcd.setCursor(0, 3); lcd.print("AKTIVASI E-KTP ANDA!");
  delay(2000); lcd.clear();
  Serial.println("Silahkan aktivasi kartu");
  goto action_rw;
```


Kode aktivasi alarm, deteksi sensor IR, dan motor on

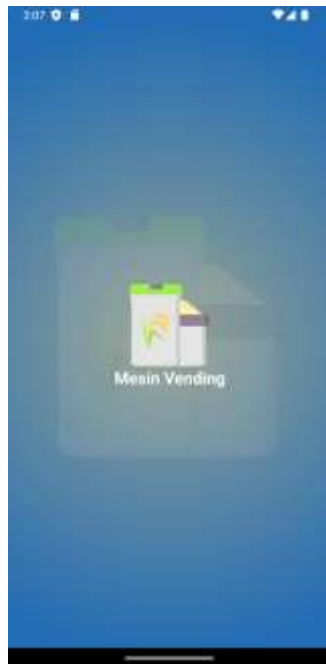
```
void alarmAktivasi(){
  for(int x=0;x<2;x++){
    digitalWrite(buzzer,1);delay(1000);
    digitalWrite(buzzer,0);delay(100);}
  }

void alarmGagal(){
  for(int x=0;x<6;x++){
    digitalWrite(buzzer,1);delay(100);
    digitalWrite(buzzer,0);delay(100);}
  }

void alarmSukses(){
  for(int x=0;x<2;x++){
    digitalWrite(buzzer,1);delay(100);
    digitalWrite(buzzer,0);delay(100);}
  }

void motorON(){
  while(digitalRead(IR)!=0)
  { digitalWrite(buzzer,HIGH);digitalWrite(motor,LOW); }
  digitalWrite(buzzer,HIGH);digitalWrite(motor,LOW);delay(2000);
  digitalWrite(motor,HIGH);digitalWrite(buzzer,LOW);
}
```

Lampiran 2 Tampilan Aplikasi Android



Tampilan Splash Screen



Tampilan Login



Tampilan Menu Dashboard



Tampilan Menu Pengguna



Tampilan Menu Transaksi

Lampiran 3 Gambar Sistem Mesin Vending



Tampak Depan



Tampak Samping



Rangkaian Alat



Tampak Atas



Tampak Dalam



Synchronous Motor AC

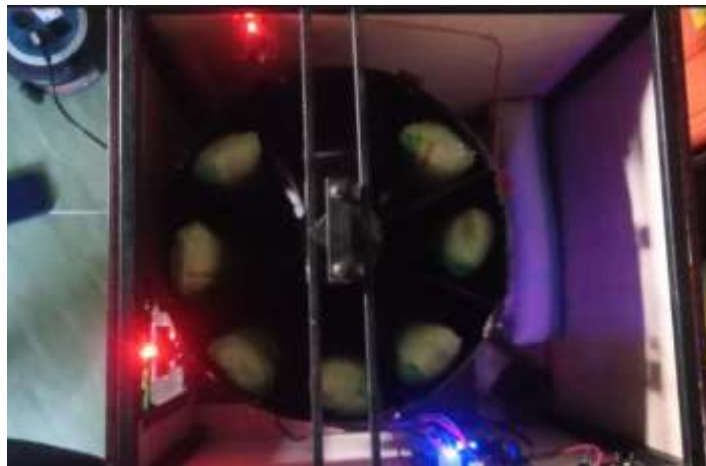
Lampiran 4 Gambar Sistem Saat Proses Transaksi



Saat Tap/Scan E-KTP



Saat Transaksi Sukses



Saat Proses Pemutaran Wadah