Custom Keypad:

```
#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS]= {

   {'1','2','3','A'},

   {'4','5','6','B'},

   {'7','8','9','C'},

   {'*','0','#','D'}

   };

byte rowPins[ROWS] =
{A0,A1,A2,A3};

byte colPins[COLS] = {A4,A5,A6,A7};

Keypad keypad = Keypad(
makeKeymap(keys),

rowPins,colPins,ROWS,COLS);


void setup() {

   // put your setup code here, to
run once:

   Serial.begin(9600);

}

void loop() {

   // put your main code here, to
run repeatedly:

char key = keypad.getKey();

if(key != NO_KEY){

   Serial.println(key);

}

}

/*

PINOUT:

RC522 MODULE        Uno/Nano
2560

SDA                 D7/D6
D53/D43

SCK                 D13
D52

MOSI                D11
D51

MISO                D12
D50

IRQ                 N/C
N/C

GND                 GND
GND

RST                 D9/D8
D45/D44

3.3V                3.3V
3.3V

*/
```

```
//#include <EEPROM.h>
#include <Wire.h>    // i2C
Conection Library
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include
<LiquidCrystal_PCF8574.h>
#include <Password.h>
#include <Keypad.h>
#include <SPI.h>
#include <RFID.h>
#define SS_PIN 53
#define RST_PIN 49
RFID rfid(SS_PIN,RST_PIN);
LiquidCrystal_PCF8574 lcd(0x27);
//SDA & SCL ARduino >< LCD i2c
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&Serial3);
//RX TX PIN 14, 15
int serNum[5];
//Variable buffer Scan Card
int cards[][5] = {
//ID Kartu yang diperbolehkan
masuk
   {136, 4, 104, 147, 119},
//Ektp kakak
   {161, 25, 234, 169, 251},
//Gantungan kunci biru
   {217, 217, 214, 42, 252},
//Kartu yg Polos
   {151, 61, 132, 36, 10}
//Kartu yg Polos
};
bool access = false;
Password password =
Password("123456"); //<-- TULIS
PASSWORD SISTEM ANDA DISINI
const byte ROWS = 4; // Four rows
const byte COLS = 4; //    columns
// Define the Keymap
char keys[ROWS][COLS] = {
   {'1','2','3','A'},
   {'4','5','6','B'},
   {'7','8','9','C'},
   {'*','0','#','D'}
};
byte rowPins[ROWS] =
{A0,A1,A2,A3};
byte colPins[COLS] = {A4,A5,A6,A7};
// Create the Keypad
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );


c onst int buzzer = 29;
//Passive buzzer ke Pin D29 dan GND
const int sensorsentuh = 30;
//Sensor Sentuh D30
const int doorlock = 31;
//Driver Selenoid Door Lock/Relay input ke pin D31
int nilaisensor;
int alarm = 0;
int idFinger;
uint8_t alarmStat = 0;
uint8_t maxError = 5;
unsigned long previousMillis = 0;
const long interval = 1000;
uint8_t pwMode=0;
uint8_t pwPos=0;
void setup(){
        Serial.begin(9600);
        finger.begin(57600);
        lcd.begin(16, 2);
        lcd.setBacklight(255);
        SPI.begin();
        rfid.init();

keypad.addEventListener(keypadEvent); //add an event listener for this keypad
        pinMode(sensorsentuh, INPUT);
        pinMode(doorlock, OUTPUT);
                if (finger.verifyPassword()) {
            Serial.println("FingerPrint Sensor Ditemukan!");
        } else {
            Serial.println("FingerPrint Sensor Tidak Ditemukan! :(");
            while (1) { delay(1); }
        }
        tone (buzzer,1200);
        lcd.setCursor (0,0);
        lcd.print(F("DoorLock System "));
        lcd.setCursor (0,1);
        lcd.print(F(" Control Access "));
        delay (2000);
        lcd.clear();

        noTone (buzzer);

}

void loop(){

    FINGERPRINT();

    keypad.getKey();    //Reading Keypad

    nilaisensor= digitalRead(sensorsentuh);

    Serial.println(nilaisensor);

    if(idFinger >= 0){

        ACCEPT ();

    }

    if(nilaisensor == 1){

        ACCEPT ();

    }

    if (alarm >= maxError){

        alarmStat = 1;        }

    if (alarmStat == 0 && pwMode == 0){

    lcd.setCursor (0,0);

    lcd.print(F(" -System Ready- "));

    unsigned long currentMillis = millis();

//Membuat perubahan tulisan pada LCD tanpa menggunakan delay

//Pada sistem stanby sama sekali tidak boleh ada delay untuk menghindari Keypad NGADAT

    if (currentMillis - previousMillis >= interval) {

            //previousMillis = currentMillis;

    lcd.setCursor (0,1);

    lcd.print(F("Scan Card/Finger"));

    }

    if (currentMillis - previousMillis >= (2*interval)) {

            previousMillis = currentMillis;

    lcd.setCursor (0,1);

    lcd.print(F(" Enter Password "));

    }

        if(rfid.isCard()){

                if(rfid.readCardSerial()){

Serial.print(rfid.serNum[0]);

                        Serial.print(" ");

Serial.print(rfid.serNum[1]);
```

```
              Serial.print(" ");
Serial.print(rfid.serNum[2]);
              Serial.print(" ");
Serial.print(rfid.serNum[3]);
              Serial.print(" ");
Serial.print(rfid.serNum[4]);
              Serial.println("");
              for(int x = 0; x <
sizeof(cards); x++){
                for(int i = 0; i <
sizeof(rfid.serNum); i++ ){

if(rfid.serNum[i] != cards[x][i]) {
                          access
= false;
                          break;
                     } else {
                          access
= true;
                     }
                }
                if(access) break;
              }
          }
        if(access){
```

```
Serial.println("Welcome!");
            lcd.setCursor (0,0);
            lcd.print(F(" Akses
diterima "));
            lcd.setCursor (0,1);
            lcd.print("ID:");
lcd.print(rfid.serNum[0]);
lcd.print(rfid.serNum[1]);
lcd.print(rfid.serNum[2]);
lcd.print(rfid.serNum[3]);
lcd.print(rfid.serNum[4]);
            ACCEPT ();
             }
       else {
             alarm = alarm+1;
             Serial.println("Not
allowed!");
             lcd.setCursor (0,0);
             lcd.print(F(" Akses
ditolak    "));
             lcd.setCursor (0,1);
             lcd.print("ID:");

lcd.print(rfid.serNum[0]);
lcd.print(rfid.serNum[1]);
lcd.print(rfid.serNum[2]);
lcd.print(rfid.serNum[3]);

lcd.print(rfid.serNum[4]);
                    RIJECT ();
                }
        }
        rfid.halt();
}
if (alarmStat == 1) {
    lcd.setCursor (0,0);
    lcd.print(F("-System LOCKED- "));
    lcd.setCursor (0,1);
    lcd.print(F(" Please Wait "));
for(int i=10; i>0; i--){
tone (buzzer,1800);
lcd.setCursor (13,1); lcd.print(i);
lcd.print(F("    "));delay (1000);}
noTone (buzzer);
alarmStat = 0;
alarm = 0;
    }
}
```

```
void keypadEvent(KeypadEvent
eKey){
    switch (keypad.getState()){
    case PRESSED:
    Serial.print("Pressed: ");
    Serial.println(eKey);
        pwMode =1;
    pwPos=pwPos+1;
    if(pwPos==1){lcd.clear();}
    lcd.setCursor (0,0);
    lcd.print(F("Password Entered"));
    if(pwPos<7){lcd.setCursor
(4+pwPos,1);
    lcd.print(F("*"));}
    switch (eKey){
        case '*': checkPassword();
break;
        //case '#': password.reset();
break;

        default:
password.append(eKey);
          }
    }
}
```

```
void checkPassword(){
  if (password.evaluate()){
    Serial.println("Password OK");
    lcd.setCursor (0,1);
    lcd.print(F("      -Accepted-
"));
    ACCEPT ();
    password.reset();
    pwPos=0;
    //Add code to run if it works
      }else{
    Serial.println("Password
SALAH");
    lcd.setCursor (0,1);
    lcd.print(F("      -Rejected-
"));
    alarm = alarm+1;
    RIJECT ();
    password.reset();
    pwPos=0;
    //add code to run if it did not
work
  }
}
void ACCEPT () {

digitalWrite(doorlock, HIGH);
           tone (buzzer,900);
           delay(100);
           tone (buzzer,1200);
           delay(100);
           tone (buzzer,1800);
           delay(200);
           noTone(buzzer);
           delay(600);
           lcd.setCursor (0,0);
           lcd.print(F(" Silahkan
Masuk "));
           lcd.setCursor (0,1);
           lcd.print(F("AutoLock
after "));
           for(int i=5; i>0; i--){
            lcd.setCursor (15,1);
lcd.print(i);
            delay (1000);
           }

digitalWrite(doorlock, LOW);
           pwMode =0;
           lcd.clear();}
```

```
void RIJECT () {

                tone (buzzer,900);

                delay(200);

                noTone(buzzer);

                delay(200);

                tone (buzzer,900);

                delay(200);

                noTone (buzzer);

                delay(500);

                pwMode =0;

                lcd.clear();

}
```

Koding Keseluruhan Alat

```
#include <Wire.h>   // i2C
Conection Library
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include
<LiquidCrystal_PCF8574.h>
#include <Password.h>
#include <Keypad.h>
#include <SPI.h>
#include <RFID.h>
#define SS_PIN 53
#define RST_PIN 49
RFID rfid(SS_PIN,RST_PIN);
LiquidCrystal_PCF8574 lcd(0x27);
//SDA & SCL ARduino >< LCD i2c
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&Serial3);
//RX TX PIN 14, 15
int serNum[5];   //Variable buffer
Scan Card
int cards[][5] = {   //ID Kartu yang
diperbolehkan masuk
   {136, 4, 61, 96, 209},            //
ktp reno
   {136, 4, 92, 40, 248},            //ktp
zopa
   {136, 4, 45, 23, 182},         // ktp
zur
   {151, 61, 132, 36, 10}      //Kartu
yg Polos
};
bool access = false;
Password password =
Password("123456"); //<-- TULIS
PASSWORD SISTEM ANDA
DISINI
const byte ROWS = 4; // Four rows
const byte COLS = 4; //   columns
// Define the Keymap
char keys[ROWS][COLS] = {
   {'1','2','3','A'},
   {'4','5','6','B'},
   {'7','8','9','C'},
   {'*','0','#','D'}
};
byte rowPins[ROWS] =
{A0,A1,A2,A3};
byte colPins[COLS] =
{A4,A5,A6,A7};
// Create the Keypad
Keypad keypad = Keypad(
makeKeymap(keys), rowPins,
colPins, ROWS, COLS );
```

```
const int buzzer = 29;
//Passive buzzer ke Pin D29 dan
GND
const int sensorsentuh = 30;
//Sensor Sentuh D30
const int doorlock = 31;
//Driver Selenoid Door Lock/Relay
input ke pin D31
int nilaisensor;
int alarm = 0;
int idFinger;
uint8_t alarmStat = 0;
uint8_t maxError = 5;
unsigned long previousMillis = 0;
const long interval = 1000;
uint8_t pwMode=0;
uint8_t pwPos=0;
void setup(){
      Serial.begin(9600);
      finger.begin(57600);
      lcd.begin(16, 2);
      lcd.setBacklight(255);
      SPI.begin();
      rfid.init();
keypad.addEventListener(keypadE
vent); //add an event listener for this
keypad
      pinMode(sensorsentuh,
INPUT);
      pinMode(doorlock, OUTPUT);
            if (finger.verifyPassword())
{
         Serial.println("FingerPrint
Sensor Ditemukan!");
      } else {
         Serial.println("FingerPrint
Sensor Tidak Ditemukan! :(");
         while (1) { delay(1); }
      }
      tone (buzzer,1200);
      lcd.setCursor (0,0);
      lcd.print(F("DoorLock System
"));
      lcd.setCursor (0,1);
      lcd.print(F(" Control Access "));
      delay (2000);
      lcd.clear();
      noTone (buzzer);
}
void loop(){
   FINGERPRINT();
   keypad.getKey();   //Reading
Keypad
   nilaisensor=
digitalRead(sensorsentuh);
   Serial.println(nilaisensor);
   if(idFinger >= 0){
      ACCEPT ();
   }
   if(nilaisensor == 1){
      ACCEPT ();
   }
   if (alarm >= maxError){
      alarmStat = 1;      }
   if (alarmStat == 0 && pwMode
== 0){
   lcd.setCursor (0,0);
   lcd.print(F(" -System Ready- "));
   unsigned long currentMillis =
millis();
//Membuat perubahan tulisan pada
LCD tanpa menggunakan delay
//Pada sistem stanby sama sekali
tidak boleh ada delay untuk
menghindari Keypad NGADAT
   if (currentMillis - previousMillis
>= interval) {
      //previousMillis =
currentMillis;
  lcd.setCursor (0,1);
  lcd.print(F("Scan Card/Finger"));
  }
  if (currentMillis - previousMillis
>= (2*interval)) {
      previousMillis =
currentMillis;
  lcd.setCursor (0,1);
  lcd.print(F(" Enter Password "));
  }
    if(rfid.isCard()){
        if(rfid.readCardSerial()){

Serial.print(rfid.serNum[0]);
            Serial.print(" ");

Serial.print(rfid.serNum[1]);
            Serial.print(" ");

Serial.print(rfid.serNum[2]);
            Serial.print(" ");

Serial.print(rfid.serNum[3]);
            Serial.print(" ");

Serial.print(rfid.serNum[4]);
            Serial.println("");
            for(int x = 0; x <
sizeof(cards); x++){
              for(int i = 0; i <
sizeof(rfid.serNum); i++ ){
                  if(rfid.serNum[i]
!= cards[x][i]) {
                      access = false;
                      break;
                  } else {
                      access = true;
                  }
              }
              if(access) break;
            }
        }
        if(access){

Serial.println("Welcome!");
          lcd.setCursor (0,0);
          lcd.print(F(" Akses
diterima "));
          lcd.setCursor (0,1);
          lcd.print("ID:");
          lcd.print(rfid.serNum[0]);
lcd.print(rfid.serNum[1]);
          lcd.print(rfid.serNum[2]);
lcd.print(rfid.serNum[3]);
          lcd.print(rfid.serNum[4]);
          ACCEPT ();
           }
      else {
           alarm = alarm+1;
           Serial.println("Not
allowed!");
           lcd.setCursor (0,0);
           lcd.print(F(" Akses
ditolak   "));
           lcd.setCursor (0,1);
           lcd.print("ID:");

lcd.print(rfid.serNum[0]);
lcd.print(rfid.serNum[1]);

lcd.print(rfid.serNum[2]);
lcd.print(rfid.serNum[3]);

lcd.print(rfid.serNum[4]);
           RIJECT ();
        }
    }
    rfid.halt();
}
if (alarmStat == 1) {
  lcd.setCursor (0,0);
  lcd.print(F("-System LOCKED-
"));
  lcd.setCursor (0,1);
  lcd.print(F(" Please Wait "));
for(int i=10; i>0; i--){
tone (buzzer,1800);
lcd.setCursor (13,1); lcd.print(i);
lcd.print(F("   "));delay (1000);}
noTone (buzzer);
alarmStat = 0;
alarm = 0;
  }
}
void keypadEvent(KeypadEvent
eKey){
  switch (keypad.getState()){
  case PRESSED:
  Serial.print("Pressed: ");
  Serial.println(eKey);
  pwMode =1;
  pwPos=pwPos+1;
  if(pwPos==1){lcd.clear();}
  lcd.setCursor (0,0);
  lcd.print(F("Password Entered"));
  if(pwPos<7){lcd.setCursor
(4+pwPos,1);
  lcd.print(F("*"));}
  switch (eKey){
    case '*': checkPassword();
break;
    //case '#': password.reset();
break;
    default:
password.append(eKey);
     }
  }
}
void checkPassword(){
  if (password.evaluate()){
    Serial.println("Password OK");
    lcd.setCursor (0,1);
    lcd.print(F("     -Accepted-
"));
    ACCEPT ();
    password.reset();
    pwPos=0;
    //Add code to run if it works
  }else{
    Serial.println("Password
SALAH");
    lcd.setCursor (0,1);
    lcd.print(F("     -Rejected-     "));
    alarm = alarm+1;
    RIJECT ();
    password.reset();
    pwPos=0;
    //add code to run if it did not
work
  }
}
void ACCEPT () {
           digitalWrite(doorlock,
HIGH);
           tone (buzzer,900);
           delay(100);
           tone (buzzer,1200);
           delay(100);
           tone (buzzer,1800);
           delay(200);
           noTone(buzzer);
           delay(600);
           lcd.setCursor (0,0);
```

```
                lcd.print(F(" Silahkan
Masuk "));
                lcd.setCursor (0,1);
                lcd.print(F("AutoLock
after "));
                for(int i=5; i>0; i--){
                 lcd.setCursor (15,1);
lcd.print(i);
                 delay (1000);
                }
                digitalWrite(doorlock,
LOW);
                pwMode =0;
                lcd.clear();}
void RIJECT () {
                tone (buzzer,900);
                delay(200);
                noTone(buzzer);
                delay(200);
                tone (buzzer,900);
                delay(200);
                noTone (buzzer);
                delay(500);
                pwMode =0;
                lcd.clear();
}
void FINGERPRINT(){
  idFinger= getFingerprintIDez();
  delay(50);
}
//-----PROSES
FINGERPRINT-----//
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case
FINGERPRINT_NOFINGER:
      Serial.println("No finger
detected");
      lcd.setCursor (0,0);
     lcd.print("Finger Tidak");
     lcd.setCursor (0,1);
     lcd.print(" Terdaftar");
      return p;
    case
FINGERPRINT_PACKETRECIEV
EERR:

Serial.println("Communication
error");
      return p;
    case
FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging
error");
      return p;
    default:
      Serial.println("Unknown
error");
      return p;}
  // OK success!
  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image
converted");
      break;
    case
FINGERPRINT_IMAGEMESS:
      Serial.println("Image too
messy");
      return p;
    case
FINGERPRINT_PACKETRECIEV
EERR:

Serial.println("Communication
error");
      return p;
    case
FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find
fingerprint features");
      return p;
    case
FINGERPRINT_INVALIDIMAGE
:
      Serial.println("Could not find
fingerprint features");
      return p;
    default:
      Serial.println("Unknown
error");
      return p;
  }
  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print
match!");
  } else if (p ==
FINGERPRINT_PACKETRECIEV
EERR) {
    Serial.println("Communication
error");
    return p;
  } else if (p ==
FINGERPRINT_NOTFOUND) {
```

```
    Serial.println("Did not find a
match");
    return p;
  } else {
    Serial.println("Unknown
error");
    return p;
  }
  // found a match!
  Serial.print("Found ID #");
Serial.print(finger.fingerID);
  Serial.print(" with confidence of
"); Serial.println(finger.confidence);
  return finger.fingerID;}
// returns -1 if failed, otherwise
returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)
return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)
return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)
return -1;
  Serial.print("Found ID #");
Serial.print(finger.fingerID);
  Serial.print(" with confidence of
"); Serial.println(finger.confidence);
  lcd.setCursor (0,0);
  lcd.print("FingerTerdeteksi");
  lcd.setCursor (0,1);
  lcd.print(" Silahkan Masuk ");
  delay(1000);
  return finger.fingerID;
}
```