LAMPIRAN-LAMPIRAN

PROTOTIPE SISTEM KEAMANAN DAN OTOMATISASI RUMAH PINTAR BERBASIS INTERNET OF THINGS (IOT)

ORIGINALITY REPORT

15%	16%	3%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.uii.ac.id Internet Source	3%
2	repository.its.ac.id Internet Source	1%
3	j-ptiik.ub.ac.id Internet Source	1%
4	eprints.umm.ac.id Internet Source	1%
5	eprints.ums.ac.id Internet Source	1%
6	media.neliti.com Internet Source	1%
7	jurnal.umj.ac.id Internet Source	1%
8	www.scribd.com Internet Source	1%
9	journal.upgris.ac.id	

	Internet Source	1%
10	Submitted to Universitas Negeri Surabaya The State University of Surabaya Student Paper	1%
11	Submitted to Universitas Muhammadiyah Makassar Student Paper	1%
12	123dok.com Internet Source	1%
13	core.ac.uk Internet Source	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		

Link video Youtube demo alat :

https://youtu.be/EdC0EM1TTvY

Kode Program Arduino Mega

```
#include <NewPing.h> //library ultrasonik
#include <Adafruit_Fingerprint.h> // library fingerprint

//pembuatan object sensor ultrasonik
NewPing s_kmr_tidurA(12, 12);
NewPing s_kmr_tidurB(11, 11);
NewPing s_ruang_tengahA(10, 10);
NewPing s_ruang_tengahB(9, 9);
NewPing s_kmr_mandiA(8, 8);
NewPing s_kmr_mandiB(7, 7);

//pembuatan object fingerprint
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);

//variabel deteksi orang masuk & keluar
int kmtA, kmtB, rtA, rtB, kmA, kmB;
int n_kmr_tidurA, n_kmr_tidurB, n_kmr_mandiA, n_kmr_mandiB,
n_ruang_tengahA, n_ruang_tengahB;
boolean fkmtA, fkmtB = false;
int kmtIN, kmtOUT, rtIN, rtOUT, kmIN, kmOUT;

//waktu tunda finger kembali ke MENUNGGU
unsigned long timerFinger = 0;
#define waitFinger 3000

//batas nilai minimal dan maksimal deteksi gas
#define minGas 47
```

```
#define maxGas 300
//batas deteksi minimal sensor ultrasonik
#define batas 5

//status sensor pintu
#define DITUTUP 0
#define DIBUKA 1
//status finger print
#define DITERIMA 2
#define DITOLAK 1
#define MENUNGGU 0
//status alarm maling dan gas bocor
#define AMAN 0
#define DIBOBOL 1
#define BOCOR 1
//status lampu dan kipas
#define ON 0
#define OFF 1

//perangkat kendali
#define BYPASS 3
#define KMR_TIDUR 1
#define RUANG_TENGAH 2
#define KMR_MANDI 3
#define OUTDOOR1 4
#define OUTDOOR2 5
#define LOCK 6
#define FAN_KMR_TIDUR 7
#define FAN_R_TENGAH 8
#define ALARM_MALING 9
#define ALARM_GAS 10
```

```
//alokasi pin arduino (input)
#define sensorGas A0
#define resetAlarm 33
#define resetPintu 35
#define sensorPintu 37

//alokasi pin arduino (output)
#define buzzer 31
#define r_doorlock 39
#define r_fan_kmr_tidur 41
#define r_fan_ruang_tengah 43
#define r_outdoor1 45
#define r_outdoor2 47
#define r_kmr_tidur 49
#define r_ruang_tengah 51
#define r_kmr_mandi 53

//variabel perintah dari nodemcu
int p_alarm, p_bypass, p_lock, p_kmr_tidur, p_ruang_tengah, p_kmr_mandi,
p_outdoor1, p_outdoor2, p_timer;
//variabel status kondisi
int s_alarm = 0;
int s_alarm_maling, s_alarm_gas, s_bypass, s_pintu, s_finger, s_lock,
s_kmr_tidur, s_ruang_tengah, s_kmr_mandi, s_outdoor1, s_outdoor2;
int n_resetAlarm, n_resetPintu;
unsigned long t_kmr_tidurA, t_kmr_tidurB, t_kmr_mandiA, t_kmr_mandiB,
t_ruang_tengahA, t_ruang_tengahB;
int n_kmr_tidur, n_kmr_mandi, n_ruang_tengah = 0;
//id finger print
uint8_t id;
```

```
//timer sebagai non-blocking delay
unsigned long kirimStart, kirimStop, tampilStart, tampilStop, start, stop = 0;
boolean flagHapus = false;
int nilaiGas;
int flag = 0;
//status orang masuk & keluar setelah fingerprint akses
int status_orang;
#define STANDBY 0
#define MAU_MASUK 1
#define SEDANG_MASUK 2
#define SUDAH_MASUK 3
#define MAU_KELUAR 4
#define SEDANG_KELUAR 5
#define SUDAH_KELUAR 6

void setup() {
  //inisiasi mode pin = input/output.
  pinMode(sensorPintu, INPUT_PULLUP);
  pinMode(resetAlarm, INPUT_PULLUP);
  pinMode(resetPintu, INPUT_PULLUP);
  pinMode(buzzer, OUTPUT);
  pinMode(r_doorlock, OUTPUT);
  pinMode(r_fan_kmr_tidur, OUTPUT);
  pinMode(r_fan_ruang_tengah, OUTPUT);
  pinMode(r_outdoor1, OUTPUT);
  pinMode(r_outdoor2, OUTPUT);
  pinMode(r_kmr_tidur, OUTPUT);
  pinMode(r_ruang_tengah, OUTPUT);
  pinMode(r_kmr_mandi, OUTPUT);
  //inisiasi kondisi lampu dan aktuator
  lampu(KMR_TIDUR, OFF);
```

```
  lampu(RUANG_TENGAH, OFF);
  lampu(KMR_MANDI, OFF);
  lampu(OUTDOOR1, OFF);
  lampu(OUTDOOR2, OFF);
  kendali(FAN_KMR_TIDUR, OFF);
  kendali(FAN_R_TENGAH, OFF);
  kendali(LOCK, OFF);
  //inisiasi serial monitor
  Serial.begin(9600);
  //inisiasi komunikasi serial ke nodemcu
  Serial1.begin(9600);
  //inisiasi sensor fingerprint
  mulai_finger();
  //indikator sistem aktif
  Serial.println("SISTEM AKTIF");

  //AKTIFKAN SAAT MENDAFTARKAN SIDIK JARI BARU
  // daftarFinger();
}

void loop() {
  //inisisasi timer
  tampilStart = millis();
  kirimStart = millis();
  start = millis();

  fungsi_utama();

  //kirimNode() setiap 200 milidetik
  if (kirimStart - kirimStop >= 200) {
    kirimStop = kirimStart;
```

```
    kirimKeNode();
   }

   //tampilkan status di serial monitor setiap 2 detik
   if (tampilStart - tampilStop >= 2000) {
    tampilStop = tampilStart;
    tampilkanStatus();
    //dataUS();
   }
}

void kirimKeNode() {

  Serial1.print("K"); Serial1.print(s_alarm_gas);
  Serial1.print("L"); Serial1.print(s_alarm_maling);
  Serial1.print("M"); Serial1.print(s_pintu);
  Serial1.print("N"); Serial1.print(s_finger);
  Serial1.print("O"); Serial1.print(s_lock);
  Serial1.print("P"); Serial1.print(s_kmr_tidur);
  Serial1.print("Q"); Serial1.print(s_ruang_tengah);
  Serial1.print("R"); Serial1.print(s_kmr_mandi);
  Serial1.print("S"); Serial1.print(s_outdoor1);
  Serial1.print("T"); Serial1.print(s_outdoor2);
  Serial1.println("");
/*
   Serial.println("Mengirim data ke node ...");
  Serial.print("K"); Serial.print(s_alarm_gas);
  Serial.print("L"); Serial.print(s_alarm_maling);
  Serial.print("M"); Serial.print(s_pintu);
  Serial.print("N"); Serial.print(s_finger);
  Serial.print("P"); Serial.print(s_lock);
```

```
  Serial.print("Q"); Serial.print(s_kmr_tidur);
  Serial.print("R"); Serial.print(s_ruang_tengah);
  Serial.print("S"); Serial.print(s_kmr_mandi);
  Serial.print("T"); Serial.print(s_outdoor1);
  Serial.print("U"); Serial.print(s_outdoor2);
  Serial.println("");
  Serial.println(" ------------------- Mengirim data selesai -------------------");
*/
}

void bacaNode() {
  //jika terdapat lalu lintas data pada Serial1, masukkan ke dalam variabel
dataNode.
  if (Serial1.available() > 0) {
    String dataNode = Serial1.readStringUntil('\n');
    //Serial.println("Data dari node: ");
    //Serial.println(dataNode);

    //Perintah Lock
    if (dataNode.indexOf("C") >= 0) {
      p_lock = dataNode.substring(dataNode.indexOf("C") + 1,
dataNode.indexOf("#")).toInt();
    }

    //Perintah Kmr Tidur
    else if (dataNode.indexOf("D") >= 0) {
      p_kmr_tidur = dataNode.substring(dataNode.indexOf("D") + 1,
dataNode.indexOf("#")).toInt();
    }

    //Perintah R. Tengah
```

```
    else if (dataNode.indexOf("E") >= 0) {
      p_ruang_tengah = dataNode.substring(dataNode.indexOf("E") + 1,
dataNode.indexOf("#")).toInt();
    }

    //Perintah Kmr Mandi
    else if (dataNode.indexOf("G") >= 0) {
      p_kmr_mandi = dataNode.substring(dataNode.indexOf("G") + 1,
dataNode.indexOf("#")).toInt();
    }

    //Perintah Outdoor 1
    else if (dataNode.indexOf("H") >= 0) {
      p_outdoor1 = dataNode.substring(dataNode.indexOf("H") + 1,
dataNode.indexOf("#")).toInt();
    }

    //Perintah Outdoor 2
    else if (dataNode.indexOf("I") >= 0) {
      p_outdoor2 = dataNode.substring(dataNode.indexOf("I") + 1,
dataNode.indexOf("#")).toInt();
    }

    else if (dataNode.indexOf("X") >= 0) {
    }

    //Perintah Timer
    else if (dataNode.indexOf("W") >= 0) {
      p_timer = dataNode.substring(dataNode.indexOf("W") + 1,
dataNode.indexOf("#")).toInt();
    }
```

```
  }


  if (p_lock == 1) kendali(LOCK, ON);
  else kendali(LOCK, OFF);

  if (p_kmr_tidur == 1) { lampu(KMR_TIDUR,
ON);kendali(FAN_KMR_TIDUR, ON);}
  else { lampu(KMR_TIDUR, OFF); kendali(FAN_KMR_TIDUR, OFF); }

  if (p_ruang_tengah == 1) { lampu(RUANG_TENGAH, ON);
kendali(FAN_R_TENGAH, ON);}
  else { lampu(RUANG_TENGAH, OFF); kendali(FAN_R_TENGAH, OFF); }

  if (p_kmr_mandi == 1) lampu(KMR_MANDI, ON);
  else  lampu(KMR_MANDI, OFF);

  if (p_outdoor1 == 1) lampu(OUTDOOR1, ON);
  else lampu(OUTDOOR1, OFF);

  if (p_outdoor2 == 1) lampu(OUTDOOR2, ON);
  else lampu(OUTDOOR2, OFF);

  if (p_timer == 1) {
    lampu(OUTDOOR1, ON);
    lampu(OUTDOOR2, ON);
  }
  else {
    if (p_outdoor1 == 0 && p_outdoor2 == 0){
      lampu(OUTDOOR1, OFF);
    lampu(OUTDOOR2, OFF);
```

```
    }
  }
}

void tampilkanStatus() {
  Serial.println("STATUS MONITOR:");
  Serial.print("MODE     : "); Serial.println(mode);
  Serial.print("GAS      : "); Serial.println(s_alarm_gas);
  Serial.print("MALING   : "); Serial.println(s_alarm_maling);
  Serial.print("BYPASS   : "); Serial.println(s_bypass);
  Serial.print("PINTU    : "); Serial.println(s_pintu);
  Serial.print("FINGER   : "); Serial.println(s_finger);
  Serial.print("LOCK     : "); Serial.println(s_lock);
  Serial.print("K. TIDUR : "); Serial.println(s_kmr_tidur);
  Serial.print("R. TENGAH: "); Serial.println(s_ruang_tengah);
  Serial.print("K. MANDI : "); Serial.println(s_kmr_mandi);
  Serial.print("OUTDOOR 1: "); Serial.println(s_outdoor1);
  Serial.print("OUTDOOR 2: "); Serial.println(s_outdoor2);
  Serial.println(" ------------------------------------");
  Serial.println("STATUS KENDALI:");
  Serial.print("KENDALI ALARM   : "); Serial.println(p_alarm);
  Serial.print("KENDALI BYPASS   : "); Serial.println(p_bypass);
  Serial.print("KENDALI LOCK     : "); Serial.println(p_lock);
  Serial.print("KENDALI K. TIDUR : "); Serial.println(p_kmr_tidur);
  Serial.print("KENDALI R. TENGAH: "); Serial.println(p_ruang_tengah);
  Serial.print("KENDALI K. MANDI : "); Serial.println(p_kmr_mandi);
  Serial.print("KENDALI OUTDOOR 1: "); Serial.println(p_outdoor1);
  Serial.print("KENDALI OUTDOOR 2: "); Serial.println(p_outdoor2);
  Serial.println(" ------------------------------------");
  //HANYA DIAKTIFKAN JIKA DIBUTUHKAN
 // Serial.print("RESET ALARM : "); Serial.println(n_resetAlarm);
```

```
 // Serial.print("BYPASS PINTU: "); Serial.println(n_resetPintu);
 // Serial.print("ORANG       : "); Serial.println(status_orang);
  /*
    Serial.println(" -------------------------------------");
    Serial.print("Nilai GAS : "); Serial.println(nilaiGas);
    Serial.println(" -------------------------------------");
    Serial.print("K. Tidur A : "); Serial.print(n_kmr_tidurA); Serial.print(" | ");
Serial.print("K. Tidur B : "); Serial.println(n_kmr_tidurB);
    Serial.print("R Tengah  A: "); Serial.print(n_ruang_tengahA); Serial.print(" | ");
Serial.print("R Tengah B : "); Serial.println(n_ruang_tengahB);
    Serial.print("K. Mandi  A: "); Serial.print(n_kmr_mandiA); Serial.print(" | ");
Serial.print("K. Mandi B : "); Serial.println(n_kmr_mandiB);
  */

Serial.println("==========================================================
====================");
}

void dataUS() {
  Serial.print("n Kmr Tidur: "); Serial.println(n_kmr_tidur);
  Serial.print("n R Tengah: "); Serial.println(n_ruang_tengah);
  Serial.print("n Kmr Mandi: "); Serial.println(n_kmr_mandi);
  Serial.println("-----------------------------------");

}

void mulai_finger() {
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Sensor FINGERPRINT ditemukan!");
```

```
  } else {
    Serial.println("Sensor FINGERPRINT tidak ditemukan!");
    while (1) {
      delay(1);
    }
  }

  finger.getTemplateCount();
  Serial.print("Database fingerprint: "); Serial.print(finger.templateCount);
Serial.println(" templates");
  Serial.println("Sensor FINGERPRINT ready!");
}

uint8_t bacaFinger() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Fingerprint terbaca");
      break;
    case FINGERPRINT_NOFINGER:
      //Serial.println("Fingerprint tidak terdeteksi!");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Komunikasi Error!");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      //Serial.println("Image Error!");
      return p;
    default:
      //Serial.println("Unknown Error!");
      return p;
```

```
  }

  // OK success!

  p = finger.image2Tz();
  switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Imgage terkonversi");
      break;
    case FINGERPRINT_IMAGEMESS:
      //Serial.println("Image buruk");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Komunikasi error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      //Serial.println("Tidak ada fitur fingerprint");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      //Serial.println("Tidak ada fitur fingerprint");
      return p;
    default:
      //Serial.println("Unknown error");
      return p;
  }

  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Fingerprint Cocok!"); Serial.print("--> ID: ");
Serial.print(finger.fingerID);
```

```
    s_finger = DITERIMA;
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    //Serial.println("Komunikasi error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Fingerprint Tidak Cocok!");
    s_finger=DITOLAK;
    return p;
  } else {
    //Serial.println("Unknown error");
    return p;
  }
  return finger.fingerID;
}

//DAFTAR BARU
uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}



uint8_t inputBaru() {

  int p = -1;
```

```
  Serial.print("Menunggu sidik jari untuk didaftarkan dengan urutan ke: ");
Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      // Serial.println(".");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
      //Serial.println("Communication error");
      break;
    case FINGERPRINT_IMAGEFAIL:
      //Serial.println("Imaging error");
      break;
    default:
      //Serial.println("Unknown error");
      break;
    }
  }

  // OK success!

  p = finger.image2Tz(1);
  switch (p) {
    case FINGERPRINT_OK:
      //Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
```

```
    //Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    //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;
}

Serial.println("ANGKAT JARI");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("LETAKKAN JARI YANG SAMA SEKALI LAGI!");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    //Serial.println("Image taken");
    break;
```

```
  case FINGERPRINT_NOFINGER:
    //Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    //Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    //Serial.println("Imaging error");
    break;
  default:
    //Serial.println("Unknown error");
    break;
  }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    //Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    //Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    //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!
Serial.print("Berhasil menyimpan sidik jari dengan ID: ");  Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
  Serial.println("Sidik Jari Cocok!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  //Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  //Serial.println("Fingerprints did not match");
  return p;
} else {
  //Serial.println("Unknown error");
  return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  Serial.println("BERHASIL DISIMPAN!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  //Serial.println("Communication error");
```

```
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    //Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    //Serial.println("Error writing to flash");
    return p;
  } else {
    //Serial.println("Unknown error");
    return p;
  }
}

void daftarFinger(){
  Serial.println("SIAP MENDAFTARKAN SIDIK JARI BARU!");
  Serial.println("Masukkan ID:(mulai dari 1 to 127)--> ");
  id = readnumber();
  if (id == 0) {// ID #0 not allowed, try again!
     return;
  }
  Serial.print("Mendaftarkan ID: ");
  Serial.println(id);

  while (!  inputBaru() );
}

void hapusSemua(){

  Serial.println("\n\nHAPUS SEMUA DATABASE!");
  Serial.println("Masukkan 'Y' untuk melanjutkan!");
```

```
  while (1) {
    if (Serial.available() && (Serial.read() == 'Y')) {
      break;
    }
  }

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Sensor FINGERPRINT ditemukan!");
  } else {
    Serial.println("Sensor FINGERPRINT tidak ditemukan!");
    while (1);
  }

  finger.emptyDatabase();

  Serial.println("DATABASE BERHASIL DIHAPUS!");
}

void kendali(int target, int kondisi) {
  switch (target) {
    case ALARM_MALING:
      s_alarm_maling = kondisi;
      digitalWrite(buzzer, kondisi);
      break;

    case ALARM_GAS:
      s_alarm_gas = kondisi;
      digitalWrite(buzzer, kondisi);
```

```
      break;

    case BYPASS:
      s_bypass = !kondisi;
      break;

    case FAN_KMR_TIDUR:
      digitalWrite(r_fan_kmr_tidur, kondisi);
      break;

    case FAN_R_TENGAH:
      digitalWrite(r_fan_ruang_tengah, kondisi);
      break;

    case LOCK:
      s_lock = !kondisi;
      digitalWrite(r_doorlock, kondisi);
      break;
  }
}

void lampu(int ruang, int kondisi) {
  switch (ruang) {
    case KMR_TIDUR:
      s_kmr_tidur = !kondisi;
      digitalWrite(r_kmr_tidur, kondisi);
      break;

    case RUANG_TENGAH:
      s_ruang_tengah = !kondisi;
      digitalWrite(r_ruang_tengah, kondisi);
```

```
      break;

    case KMR_MANDI:
      s_kmr_mandi = !kondisi;
      digitalWrite(r_kmr_mandi, kondisi);
      break;

    case OUTDOOR1:
      s_outdoor1 = !kondisi;
      digitalWrite(r_outdoor1, kondisi);
      break;

    case OUTDOOR2:
      s_outdoor2 = !kondisi;
      digitalWrite(r_outdoor2, kondisi);
      break;
  }
}

void bacaUltrasonik() {
  n_kmr_tidurA = s_kmr_tidurA.ping_cm();
  n_kmr_tidurB = s_kmr_tidurB.ping_cm();
  n_kmr_mandiA = s_kmr_mandiA.ping_cm();
  n_kmr_mandiB = s_kmr_mandiB.ping_cm();
  n_ruang_tengahA = s_ruang_tengahA.ping_cm();
  n_ruang_tengahB = s_ruang_tengahB.ping_cm();

  //KAMAR TIDUR
  //jika sensor A dilewati
  if (n_kmr_tidurA > 0 && n_kmr_tidurA <= batas) kmtA = 1;
```

```
//jika obyek telah meninggalkan sensor A
if (kmtOUT == 1 && n_kmr_tidurA > 0 && n_kmr_tidurA > batas) {
  kmtA = 0;
  kmtOUT = 0;
}
//jika obyek telah mencapai sensor B setelah sebelumnya dari sensor A
if (kmtA == 1 && n_kmr_tidurB > 0 && n_kmr_tidurB <= batas) {
  n_kmr_tidur++;
  kmtA = 0;
  kmtIN = 1;
  kmtOUT = 0;
}

//ulangi proses di atas utk sensor B
if (n_kmr_tidurB > 0 && n_kmr_tidurB <= batas)kmtB = 1;
if (kmtIN == 1 && n_kmr_tidurB > 0 && n_kmr_tidurB > batas) {
  kmtB = 0;
  kmtIN = 0;
}
if (kmtB == 1 && n_kmr_tidurA > 0 && n_kmr_tidurA <= batas) {
  n_kmr_tidur--;
  kmtB = 0;
  kmtIN = 0;
  kmtOUT = 1;
}

//RUANG TENGAH
if (n_ruang_tengahA > 0 && n_ruang_tengahA <= batas) rtA = 1;
if (rtOUT == 1 && n_ruang_tengahA > 0 && n_ruang_tengahA > batas) {
  rtA = 0;
  rtOUT = 0;
```

```
}
if (rtA == 1 && n_ruang_tengahB > 0 && n_ruang_tengahB <= batas) {
  n_ruang_tengah++;
  rtA = 0;
  rtIN = 1;
  rtOUT = 0;
}
if (n_ruang_tengahB > 0 && n_ruang_tengahB <= batas)rtB = 1;
if (rtIN == 1 && n_ruang_tengahB > 0 && n_ruang_tengahB > batas) {
  rtB = 0;
  rtIN = 0;
}
if (rtB == 1 && n_ruang_tengahA > 0 && n_ruang_tengahA <= batas) {
  n_ruang_tengah--;
  rtB = 0;
  rtIN = 0;
  rtOUT = 1;
}

//KAMAR MANDI
if (n_kmr_mandiA > 0 && n_kmr_mandiA <= batas) kmA = 1;
if (kmOUT == 1 && n_kmr_mandiA > 0 && n_kmr_mandiA > batas) {
  kmA = 0;
  kmOUT = 0;
}
if (kmA == 1 && n_kmr_mandiB > 0 && n_kmr_mandiB <= batas) {
  n_kmr_mandi++;
  kmA = 0;
  kmIN = 1;
  kmOUT = 0;
}
```

```
  if (n_kmr_mandiB > 0 && n_kmr_mandiB <= batas)kmB = 1;
  if (kmIN == 1 && n_kmr_mandiB > 0 && n_kmr_mandiB > batas) {
    kmB = 0;
    kmIN = 0;
  }
  if (kmB == 1 && n_kmr_mandiA > 0 && n_kmr_mandiA <= batas) {
    n_kmr_mandi--;
    kmB = 0;
    kmIN = 0;
    kmOUT = 1;
  }
}

void bacaReset() {
  n_resetAlarm = digitalRead(resetAlarm);
  n_resetPintu = digitalRead(resetPintu);

  if (n_resetAlarm == LOW) {
    s_alarm=1;
  }
  if (n_resetPintu == LOW) {
    kendali(BYPASS, ON);
  }
}

void bacaPintu() {
  int n_pintu = digitalRead(sensorPintu);
  if (n_pintu == LOW) s_pintu = DITUTUP;
  else s_pintu = DIBUKA;
}
```

```
void bacaGas() {
  nilaiGas = analogRead(sensorGas);

  if (nilaiGas >= maxGas) {
   s_alarm_gas=1;
  }
}


//bunyikan buzzer ON dan OFF sebanyak n dengan jeda sebesar durasi
void bunyi_ditolak(int n, int durasi) {
  for (int x = 0; x < n; x++) {
    digitalWrite(buzzer, HIGH); delay(durasi);
    digitalWrite(buzzer, LOW); delay(durasi);
  }
}


void fungsi_utama() {
//panggil semua fungsi
  bacaUltrasonik();
  bacaReset();
  bacaPintu();
  bacaGas();
  bacaFinger();
  bacaNode();


  //KONDISI ORANG MASUK
  if (s_finger == MENUNGGU && s_pintu == DITUTUP && s_bypass == 1 &&
flag == 2) {
```

```
    status_orang = SUDAH_KELUAR;
    s_bypass = 0;
    flag = 0;
  }
  else if (s_finger == DITERIMA && s_pintu == DITUTUP && s_bypass == 0
&& flag == 0) status_orang = MAU_MASUK;
  else if (s_finger == DITERIMA && s_pintu == DIBUKA && s_bypass == 0
&& flag == 0) {
    status_orang = SEDANG_MASUK;
    flag = 1;
  }
  else if (s_finger == DITERIMA && s_pintu == DITUTUP && s_bypass == 0
&& flag == 1) {
    status_orang = SUDAH_MASUK;
    flag = 0;
  }
  else if (s_finger == MENUNGGU && s_pintu == DITUTUP && s_bypass == 1
&& flag == 0) status_orang = MAU_KELUAR;
  else if (s_finger == MENUNGGU && s_pintu == DIBUKA && s_bypass == 1
&& flag == 0) {
    status_orang = SEDANG_KELUAR;
    flag = 2;
  }
  else if (s_finger == MENUNGGU && s_pintu == DITUTUP && s_bypass == 0
&& flag == 0) {
    status_orang = STANDBY;
  }

  //KENDALI DOOR LOCK

  if (status_orang == SUDAH_KELUAR || status_orang == SUDAH_MASUK) {
```

```
    delay(300);
    kendali(LOCK, OFF);
    s_finger = MENUNGGU;
    s_bypass = 0;
    delay(100);
  } else if ((status_orang == MAU_MASUK || status_orang ==
SEDANG_MASUK || status_orang == MAU_KELUAR || status_orang ==
SEDANG_KELUAR) && p_lock == 0) {
    kendali(LOCK, ON);
  }

  //FINGER DITOLAK
 // if (s_finger == DITOLAK)timerFinger = millis();
  if (s_finger == DITOLAK){
    kirimKeNode();
    bunyi_ditolak(3, 100);
    digitalWrite(buzzer, LOW);
    s_finger==DITOLAK;
    kirimKeNode();
    delay(10000);
    s_finger = MENUNGGU;
    delay(500);
  }
  else if (s_finger == MENUNGGU) timerFinger = 0;

  //KENDALI OTOMATIS LAMPU KAMAR
  if(p_kmr_tidur == 0 && p_kmr_mandi == 0 && p_ruang_tengah == 0){
     if (n_kmr_tidur > 0)  {
    lampu(KMR_TIDUR, ON);
    kendali(FAN_KMR_TIDUR, ON);
  }
```

```
  else if (n_kmr_tidur <= 0) {
    lampu(KMR_TIDUR, OFF);
    kendali(FAN_KMR_TIDUR, OFF);
  }

  if (n_kmr_mandi > 0) lampu(KMR_MANDI, ON);
  else if (n_kmr_mandi <= 0) lampu(KMR_MANDI, OFF);

  if (n_ruang_tengah > 0) {
    lampu(RUANG_TENGAH, ON);
    kendali(FAN_R_TENGAH, ON);
  }
  else if (n_ruang_tengah <= 0) {
    lampu(RUANG_TENGAH, OFF);
    kendali(FAN_R_TENGAH, OFF);
  }
  }

  //DETEKSI PEMBOBOLAN
  if (s_bypass != 1) {
    if (s_alarm == 0 && s_pintu == DIBUKA && (s_finger == DITOLAK ||
s_finger == MENUNGGU)) kendali(ALARM_MALING, DIBOBOL);
    if (s_alarm_gas == 1) kendali(ALARM_GAS, BOCOR);
  }
  if (s_alarm == 1) {
    kendali(ALARM_GAS, AMAN);
    kendali(ALARM_MALING, AMAN);
    delay(5000);
    s_alarm = 0;
  }
}
```