
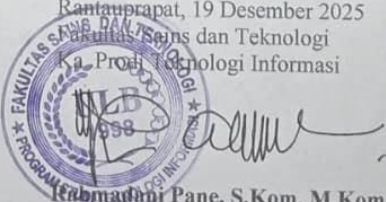



LAMPIRAN

1. Lampiran Surat Izin Penelitian

	FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS LABUHANBATU PROGRAM STUDI : AGROTEKNOLOGI - TEKNOLOGI INFORMASI - SISTEM INFORMASI - MANAJEMEN INFORMATIKA Jl. SM. Raja No. 126-A KM. 3,5 Aek Tapa - Rantauprapat - Sumatera Utara - Pos 21415 Telp./Fax. (0624) 21901
Nomor	: 33 /TI/FST-ULB/XII/2025
Hal	: Permohonan Izin Penelitian
Kepada Yth. Klinik Utama Spesialis Bunda di - Tempat	
Sehubungan dengan rencana Penelitian untuk Skripsi/Tugas Akhir Mahasiswa Program Studi S-1 Teknologi Informasi Fakultas Sains dan Teknologi tersebut dibawah ini :	
Nama	: Amanda Anastasya
NPM	: 2208100009
Program Studi	: T-1 Teknologi Informasi
Judul Tugas Akhir	: RANCANG BANGUN SISTEM MONITORING DETAK JANTUNG PASIEN DAN SPO2 BERBASIS IOT PADA KLINIK SPESIALIS BUNDA RANTAU
Lokasi Penelitian	: Klinik Utama Spesialis Bunda
Untuk keperluan tersebut diatas, agar kiranya dapat memberi izin pelaksanaan penelitian di wilayah Bapak/Ibu. Dalam proses pelaksanaannya segala sesuatu yang berkaitan dengan penelitian tersebut akan diselesaikan oleh mahasiswa yang bersangkutan.	
Demikian hal ini kami sampaikan atas perhatian dan bantuannya diucapkan terima kasih.	
Rantauprapat, 19 Desember 2025	
 Rabmadani Pane, S.Kom, M.Kom NIDN. 0110058601	

2. Lampiran Surat Balasan dari Tempat Penelitian



KLINIK UTAMA SPESIALIS BUNDA
Jl. Ahmad Yani No. 197 Telp. 24912 Rantauprapat
Email ; ksbundaranto@gmail.com

SURAT KETERANGAN

Nomor	: 10/Rtp/KUSB/XII/2025	Kepada Yth :
Perihal	: Surat Balsan Permohonan izin penelitian	Dekan Fakultas Sains dan Teknologi Universitas Labuhanbatu
		Di
		Tempat

Dengan Hormat,


Sehubungan dengan surat tanggal 19 Desember 2025 perihal permohonan izin penelitian untuk skripsi/tugas akhir mahasiswi atas nama **Amanda Anastasya** dengan judul “ **RANCANG BANGUN SISTEM MONITORING DETAK JANTUNG PASIEN DAN SPO2 BERBASIS IOT PADA KLINIK SPESIALIS BUNDA RANTAUPRAPAT**”

Kami sampaikan beberapa hal:

1. Pada dasarnya kami tidak keberatan, maka kami dapat mengizinkan pelaksanaan penelitian tersebut di tempat kami.
2. Izin melakukan penelitian diberikan untuk keperluan akademik
3. Waktu pengambilan data harus dilakukan diwaktu hari kerja

Demikianlah surat keterangan ini diperbuat dengan sebenarnya untuk dipergunakan dengan sebaik-baiknya.

Rantauprapat, 24 Desember 2025
Pimpinan Klinik Spesialis Bunda



KLINIK SPESIALIS Bunda
Jl. A. Yani No. 197 R. Prapat
Telp. (0624) 24912

dr. H. Chairil Anwar Situmorang, Sp.PD-KGH

3. Lampiran Codengan

```

#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define BLYNK_TEMPLATE_ID "TMPL6fScv19X4"
#define BLYNK_TEMPLATE_NAME "Spo2"
#define BLYNK_AUTH_TOKEN "gZplxwiAGMmz8mf1tH-6VbgTyrPYKn8"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define SCREEN_ADDRESS 0x3C

// ===== KONFIGURASI BLYNK =====
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "SOLIN_PROJECT"; // Nama WiFi
char pass[] = "17AGUSTUS2025"; // Kata Sandi WiFi

// Pin Virtual untuk Blynk
#define VPIN_BPM V0 // Pin Virtual untuk BPM
#define VPIN_SPO2 V1 // Pin Virtual untuk SpO2
#define VPIN_STATUS V2 // Pin Virtual untuk Status
#define VPIN_LED V3 // Pin Virtual untuk indikator LED

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
MAX30105 particleSensor;

double avered = 0;
double aveir = 0;
double sumirrms = 0;
double sumredrms = 0;
int i = 0;
int Num = 200;

double ESpO2 = 97.0;
double FSpO2 = 0.95;
double frate = 0.95;
#define TIMETOB00T 3000
#define SCALE 88.0
#define SAMPLING 5
#define FINGER_ON 30000
#define MINIMUM_SPO2 88.0

const byte RATE_SIZE = 4;
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatsPerMinute;
int beatAvg;

enum StatusPengukuran {
  MENUNGGU,

```

```

MENDETEKSI,
MENGUKUR,
TAMPIL_HASIL
};

StatusPengukuran statusSaatIni = MENUNGGU;
unsigned long waktuMulaiStatus = 0;
unsigned long waktuTampilHasil = 0;

const unsigned long DURASI_DETEKSI = 8000;
const unsigned long DURASI_PENGUKURAN = 40000;
const unsigned long DURASI_TAMPIL_HASIL = 20000;

int bpmAkhir = 0;
double spo2Akhir = 0.0;

int jumlahPembacaanBagus = 0;
const int MIN_PEMBACAAN_BAGUS = 10;

// Timer untuk pembaruan Blynk (agar tidak terlalu sering)
unsigned long pembaruanBlynkTerakhir = 0;
const unsigned long INTERVAL_PEMBARUAN_BLYNK = 2000; // Perbarui setiap 2 detik

#define USEFIFO

void setup()
{
  Serial.begin(115200);
  Serial.println("Menginisialisasi...");

  // Inisialisasi OLED
  if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("Alokasi SSD1306 gagal"));
    for(;;);
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println(F("Menginisialisasi..."));
  display.display();
  delay(1000);

  // Koneksi WiFi dan Blynk
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println(F("Menghubungkan WiFi..."));
  display.display();

  Blynk.begin(auth, ssid, pass);

  display.clearDisplay();
  display.setCursor(0, 0);
  display.println(F("WiFi Terhubung!"));
  display.display();
  delay(1000);

```

```

// Inisialisasi sensor MAX30102
while (!particleSensor.begin(Wire, I2C_SPEED_FAST))
{
  Serial.println("MAX30102 tidak ditemukan. Periksa kabel/daya/jumper solder pada papan MH-
ET LIVE MAX30102.");
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println(F("MAX30102"));
  display.println(F("tidak ditemukan!"));
  display.display();
  delay(2000);
}

byte ledBrightness = 0x7F;
byte sampleAverage = 4;
byte ledMode = 2;
int sampleRate = 200;
int pulseWidth = 411;
int adcRange = 16384;
particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth,
adcRange);

particleSensor.enableDIETEMPRDY();

// Tetapkan nilai awal di Blynk
Blynk.virtualWrite(VPIN_BPM, 0);
Blynk.virtualWrite(VPIN_SPO2, 0);
Blynk.virtualWrite(VPIN_STATUS, "Siap");
Blynk.virtualWrite(VPIN_LED, 0); // LED mati

display.clearDisplay();
display.display();
}

void resetPengukuran() {
  avered = 0;
  aveir = 0;
  sumirrms = 0;
  sumredrms = 0;
  i = 0;
  rateSpot = 0;
  beatAvg = 0;
  ESpO2 = 95.0;
  jumlahPembacaanBagus = 0;
  for (byte x = 0; x < RATE_SIZE; x++) {
    rates[x] = 0;
  }
}

void perbaruiBlynk(int bpm, double spo2, String status) {
  if (millis() - pembaruanBlynkTerakhir >= INTERVAL_PEMBARUAN_BLYNK) {
    Blynk.virtualWrite(VPIN_BPM, bpm);
    Blynk.virtualWrite(VPIN_SPO2, spo2);
    Blynk.virtualWrite(VPIN_STATUS, status);
    pembaruanBlynkTerakhir = millis();
  }
}

```

```

void tampilkanLayarMenunggu() {
    display.clearDisplay();
    display.setTextSize(2);
    display.setCursor(5, 15);
    display.println(F("Letakkan"));
    display.setCursor(5, 35);
    display.println(F("Jari Anda"));
    display.display();
}

void tampilkanLayarMendeteksi() {
    display.clearDisplay();

    display.setTextSize(1);
    display.setCursor(15, 5);
    display.println(F("MENDETEKSI..."));

    display.drawLine(0, 18, 128, 18, SSD1306_WHITE);

    display.setTextSize(1);
    display.setCursor(10, 25);
    display.println(F("Jari terdeteksi!"));
    display.setCursor(5, 38);
    display.println(F("Harap tidak bergerak"));

    int jumlahTitik = (millis() / 500) % 4;
    for (int i = 0; i < jumlahTitik; i++) {
        display.fillCircle(105 + (i * 6), 52, 2, SSD1306_WHITE);
    }

    display.display();
}

void tampilkanLayarMengukur(unsigned long detikTersisa, int bpm, double spo2) {
    display.clearDisplay();

    display.setTextSize(1);
    display.setCursor(10, 0);
    display.print(F("MENGUKUR "));
    display.print(detikTersisa);
    display.print(F("d"));

    display.drawLine(0, 12, 128, 12, SSD1306_WHITE);

    display.setTextSize(1);
    display.setCursor(5, 16);
    display.println(F("Jangan gerakkan jari!"));

    display.setCursor(0, 28);
    display.print(F("Detak: "));
    display.setTextSize(1);
    display.print(bpm);
    display.print(F(" BPM"));

    display.setTextSize(1);
    display.setCursor(0, 40);
    display.print(F("SpO2: "));
    display.print(spo2, 1);
}

```

```

display.print(F("%"));

int lebarProgress = map(DURASI_PENGUKURAN - (detikTersisa * 1000), 0,
DURASI_PENGUKURAN, 0, 128);
display.fillRect(0, 56, lebarProgress, 8, SSD1306_WHITE);
display.drawRect(0, 56, 128, 8, SSD1306_WHITE);

display.display();
}

void tampilkanLayarHasil() {
display.clearDisplay();

display.setTextSize(1);
display.setCursor(0, 2);
display.print(F("Detak Jantung:"));

display.setTextSize(2);
display.setCursor(20, 14);
if (bpmAkhir > 0) {
display.print(bpmAkhir);
} else {
display.print(F("--"));
}
display.setTextSize(1);
display.setCursor(70, 20);
display.print(F("BPM"));

display.drawLine(0, 32, 128, 32, SSD1306_WHITE);

display.setTextSize(1);
display.setCursor(0, 36);
display.print(F("Kadar Oksigen:"));

display.setTextSize(2);
display.setCursor(20, 48);
display.print(spo2Akhir, 1);
display.setTextSize(1);
display.setCursor(75, 54);
display.print(F("%"));

display.display();
}

void loop()
{
Blynk.run(); // Harus dipanggil di loop

uint32_t ir, red, green;
double fred, fir;
double SpO2 = 0;

#ifdef USEFIFO
particleSensor.check();

while (particleSensor.available()) {
#ifdef MAX30105
red = particleSensor.getFIFORed();

```

```

    ir = particleSensor.getFIFOIR();
#else
    red = particleSensor.getFIFOIR();
    ir = particleSensor.getFIFORed();
#endif

switch (statusSaatIni) {
case MENUNGGU:
    perbaruiBlynk(0, 0, "Menunggu Jari");
    Blynk.virtualWrite(VPIN_LED, 0);

    if (ir >= FINGER_ON && millis() > TIMETOBOOT) {
        statusSaatIni = MENDETEKSI;
        waktuMulaiStatus = millis();
        resetPengukuran();
        Serial.println("Jari terdeteksi! Memulai fase stabilisasi...");
        tampilkanLayarMendeteksi();
        Blynk.virtualWrite(VPIN_LED, 128);
    } else {
        static unsigned long pembaruanTampilTerakhir = 0;
        if (millis() - pembaruanTampilTerakhir > 500) {
            tampilkanLayarMenunggu();
            pembaruanTampilTerakhir = millis();
        }
    }
    break;

case MENDETEKSI:
    if (ir < FINGER_ON) {
        Serial.println("Jari diangkat saat deteksi! Dibatalkan.");
        statusSaatIni = MENUNGGU;
        resetPengukuran();
        tampilkanLayarMenunggu();
        Blynk.virtualWrite(VPIN_STATUS, "Dibatalkan");
        Blynk.virtualWrite(VPIN_LED, 0);
        break;
    }

    perbaruiBlynk(0, 0, "Mendeteksi...");
    tampilkanLayarMendeteksi();

    i++;
    fred = (double)red;
    fir = (double)ir;
    avered = avered * frate + (double)red * (1.0 - frate);
    aveir = aveir * frate + (double)ir * (1.0 - frate);
    sumredrms += (fred - avered) * (fred - avered);
    sumirrms += (fir - aveir) * (fir - aveir);

    if (millis() - waktuMulaiStatus >= DURASI_DETEKSI) {
        statusSaatIni = MENGUKUR;
        waktuMulaiStatus = millis();
        Serial.println("Stabilisasi selesai! Memulai pengukuran...");
        Blynk.virtualWrite(VPIN_LED, 255);
        i = 0;
        sumredrms = 0;
        sumirrms = 0;
    }
}

```

```

break;

case MENGUKUR:
{
if (ir < FINGER_ON) {
Serial.println("Jari diangkat saat pengukuran! Dibatalkan.");
statusSaatIni = MENUNGGU;
resetPengukuran();
tampilkanLayarMenunggu();
Blynk.virtualWrite(VPIN_STATUS, "Dibatalkan");
Blynk.virtualWrite(VPIN_LED, 0);
break;
}

unsigned long berlalu = millis() - waktuMulaiStatus;
unsigned long tersisa = DURASI_PENGUKURAN - berlalu;
unsigned long detikTersisa = (tersisa / 1000) + 1;

i++;
fred = (double)red;
fir = (double)ir;
avered = avered * frate + (double)red * (1.0 - frate);
aveir = aveir * frate + (double)ir * (1.0 - frate);
sumredrms += (fred - avered) * (fred - avered);
sumirrms += (fir - aveir) * (fir - aveir);

if (checkForBeat(ir) == true) {
long delta = millis() - lastBeat;
lastBeat = millis();

beatsPerMinute = 60 / (delta / 1000.0);

if (beatsPerMinute < 255 && beatsPerMinute > 20) {
rates[rateSpot++] = (byte)beatsPerMinute;
rateSpot %= RATE_SIZE;

beatAvg = 0;
for (byte x = 0 ; x < RATE_SIZE ; x++)
beatAvg += rates[x];
beatAvg /= RATE_SIZE;

if (beatAvg > 0) jumlahPembacaanBagus++;
}
}

if ((i % Num) == 0 && i > 0) {
if (avered > 0 && aveir > 0 && sumredrms > 0 && sumirrms > 0) {
double R = (sqrt(sumredrms) / avered) / (sqrt(sumirrms) / aveir);

SpO2 = -45.060 * R * R + 30.354 * R + 102.0;

if (SpO2 > 100) SpO2 = 100;
if (SpO2 < 85) SpO2 = 85;

ESpO2 = FSpO2 * ESpO2 + (1.0 - FSpO2) * SpO2;

if (ESpO2 > 100) ESpO2 = 100;
if (ESpO2 < 85) ESpO2 = 85;
}
}
}

```

```

    Serial.print("Rasio R: ");
    Serial.print(R, 4);
    Serial.print(" | SpO2 Mentah: ");
    Serial.print(SpO2, 2);
    Serial.print(" | SpO2 Terfilter: ");
    Serial.println(ESpO2, 2);
}

sumredrms = 0.0;
sumirrms = 0.0;
}

perbaruiBlynk(beatAvg, ESpO2, "Mengukur...");

if ((i % 20) == 0) {
    tampilkanLayarMengukur(detikTersisa, beatAvg, ESpO2);
}

if (berlalu >= DURASI_PENGUKURAN) {
    bpmAkhir = beatAvg;
    spo2Akhir = ESpO2;

    Serial.println("===== PENGUKURAN SELESAI =====");
    Serial.print("BPM Akhir: ");
    Serial.println(bpmAkhir);
    Serial.print("SpO2 Akhir: ");
    Serial.println(spo2Akhir);
    Serial.print("Pembacaan bagus: ");
    Serial.println(jumlahPembacaanBagus);
    Serial.println("=====");
    Serial.println("Beralih ke status TAMPIL_HASIL...");

    // Perbarui hasil akhir ke Blynk
    Blynk.virtualWrite(VPIN_BPM, bpmAkhir);
    Blynk.virtualWrite(VPIN_SPO2, spo2Akhir);
    Blynk.virtualWrite(VPIN_STATUS, "Selesai!");
    Blynk.virtualWrite(VPIN_LED, 0);

    statusSaatIni = TAMPIL_HASIL;
    waktuTampilHasil = millis();
    tampilkanLayarHasil();
}
break;
}

case TAMPIL_HASIL:
    static unsigned long cetakDebugTerakhir = 0;
    if (millis() - cetakDebugTerakhir > 1000) {
        Serial.print("Status TAMPIL_HASIL - Waktu tersisa: ");
        Serial.print((DURASI_TAMPIL_HASIL - (millis() - waktuTampilHasil)) / 1000);
        Serial.println(" detik");
        cetakDebugTerakhir = millis();
    }

    if (millis() - waktuTampilHasil >= DURASI_TAMPIL_HASIL) {
        Serial.println("Waktu tampil hasil selesai. Kembali ke MENUNGGU.");
        statusSaatIni = MENUNGGU;
    }
}

```

```
    resetPengukuran();  
    tampilkanLayarMenunggu();  
    Serial.println("Siap untuk pengukuran berikutnya");  
  }  
  break;  
}  
  
particleSensor.nextSample();  
}  
#endif  
}
```