

Lampiran 1 Source Code

```
#include <Arduino.h>
// #include <ArduinoJson.h>
#include <BlynkSimpleEsp32.h>
#include <EEPROM.h>

#define AP_SSID "WIRELESS"
#define AP_PASS "Dancow02"
#define BLYNK_TOKEN "7-4r4VLBJkEPXdIh_LNVX_0X9UWiKSE"

#define JUMLAH_SENSOR 6
#define TEBAL_BARANG 3 // 3cm
#define KAPASITAS 5

#define ADDR_TEBAL_BARANG1 0x01
#define ADDR_TEBAL_BARANG2 0x02
#define ADDR_TEBAL_BARANG3 0x03
#define ADDR_TEBAL_BARANG4 0x04
#define ADDR_TEBAL_BARANG5 0x05
#define ADDR_TEBAL_BARANG6 0x06

#define ECHO1 15
#define TRIG1 2
#define ECHO2 4
#define TRIG2 16
#define ECHO3 17
#define TRIG3 5
#define ECHO4 18
#define TRIG4 19
```

```
#define ECHO5 22
#define TRIG5 21
#define ECHO6 23
#define TRIG6 32

const byte ECHO_PIN[6] = {ECHO1, ECHO2, ECHO3, ECHO4, ECHO5,
ECHO6};

const byte TRIG_PIN[6] = {TRIG1, TRIG2, TRIG3, TRIG4, TRIG5, TRIG6};

const byte BLYNK_PIN[6] = {V0, V1, V2, V3, V4, V5};

const uint8_t BLYNK_PIN_TEBAL_BARANG[6] = {V20, V21, V22, V23, V24,
V25};

const uint8_t BLYNK_PIN_STOCK_FLOW_MASUK[6] = {V8, V10, V12, V14,
V16, V18};

const uint8_t BLYNK_PIN_STOCK_FLOW_KELUAR[6] = {V9, V11, V13,
V15, V17, V19};

uint32_t tick;
uint8_t margin = 3;
uint8_t tebal_barang[6];
int stock_barang[6];
int flow_masuk[6];
int flow_keluar[6];
int stock_barang_previous[6];
bool state_notify[JUMLAH_SENSOR] = {false, false, false, false, false, false};

// functions
float get_distance(const byte TRIGPIN, const byte ECHOPIN);
int kalkulasi_jumlah(float jarak, int tebal);
void blynk_connect();
void blynk_stream(int jarak[], const uint8_t pin[]);
```

```
String ipToString(IPAddress ipA);

// blynk input handler
BLYNK_WRITE(V6){
    margin = param.asInt();
    EEPROM.write(0, margin);
    EEPROM.commit();
    Serial.print("Margin sensor: ");
    Serial.println(margin);
}

BLYNK_WRITE(V7){
    if(param.asInt() == 1){
        for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
            flow_keluar[i] = 0;
            flow_masuk[i] = 0;
            Serial.println("Flow Reset");
        }
    }
}

BLYNK_WRITE(V20){
    tebal_barang[0] = param.asInt();
    EEPROM.write(ADDR_TEBAL_BARANG1, tebal_barang[0]);
    EEPROM.commit();
    Serial.printf("Tebal Barang 1: %u\r\n", tebal_barang[0]);
}

BLYNK_WRITE(V21){
```

```
tebal_barang[1] = param.asInt();  
EEPROM.write(ADDR_TEBAL_BARANG2, tebal_barang[1]);  
EEPROM.commit();  
Serial.printf("Tebal Barang 2: %u\r\n", tebal_barang[1]);  
}
```

```
BLYNK_WRITE(V22){  
    tebal_barang[2] = param.asInt();  
    EEPROM.write(ADDR_TEBAL_BARANG3, tebal_barang[2]);  
    EEPROM.commit();  
    Serial.printf("Tebal Barang 3: %u\r\n", tebal_barang[2]);  
}
```

```
BLYNK_WRITE(V23){  
    tebal_barang[3] = param.asInt();  
    EEPROM.write(ADDR_TEBAL_BARANG4, tebal_barang[3]);  
    EEPROM.commit();  
    Serial.printf("Tebal Barang 4: %u\r\n", tebal_barang[3]);  
}
```

```
BLYNK_WRITE(V24){  
    tebal_barang[4] = param.asInt();  
    EEPROM.write(ADDR_TEBAL_BARANG5, tebal_barang[4]);  
    EEPROM.commit();  
    Serial.printf("Tebal Barang 5: %u\r\n", tebal_barang[4]);  
}
```

```
BLYNK_WRITE(V25){  
    tebal_barang[5] = param.asInt();
```

```

EEPROM.write(ADDR_TEBAL_BARANG6, tebal_barang[5]);
EEPROM.commit();
Serial.printf("Tebal Barang 6: %u\r\n", tebal_barang[5]);
}

void setup() {
  Serial.begin(115200);
  for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
    pinMode(TRIG_PIN[i], OUTPUT);
    pinMode(ECHO_PIN[i], INPUT);
  }
  if(!EEPROM.begin(1)){
    Serial.println("EEPROM failed");
    while(true){delay(100);}
  }

  // load margin sensor dari EEPROM
  margin = EEPROM.read(0);

  // load tebal barang dari EEPROM
  for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
    tebal_barang[i] = EEPROM.read(ADDR_TEBAL_BARANG1 + i);
    if(tebal_barang[i] == 0){
      tebal_barang[i] = 1;
    }
    Serial.printf("Tebal Barang %i: %icm\r\n", i+1, tebal_barang[i]);
  }

  Serial.println("Connecting to blynk...");

```

```
blynk_connect();  
Serial.print("Connected: ");  
Serial.println(ipToString(WiFi.localIP()));  
  
// get initial value  
for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){  
    stock_barang[i] = kalkulasi_jumlah(get_distance(TRIG_PIN[i],  
ECHO_PIN[i]), tebal_barang[i]);  
    if(stock_barang[i] < 0){  
        stock_barang[i] = 0;  
    }  
    stock_barang_previous[i] = stock_barang[i];  
}  
}  
  
void loop() {  
    // baca jarak sensor  
    float jarak[6];  
    for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){  
        jarak[i] = get_distance(TRIG_PIN[i], ECHO_PIN[i]);  
    }  
  
    // kalkulasi jumlah barang  
    for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){  
        stock_barang[i] = kalkulasi_jumlah(jarak[i], tebal_barang[i]);  
        if(stock_barang[i] < 0){  
            stock_barang[i] = 0;  
        }  
    }  
  
    // notifikasi barang habis
```

```

if(stock_barang[i] == 0){
  if(state_notify[i] == false){
    state_notify[i] = true;
    Blynk.notify(String("Barang " + String(i+1) + " Habis"));
  }
}
if(stock_barang[i] > 0){
  state_notify[i] = false;
}
}

if(millis() - tick > 1000){
  tick = millis();
  for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
    Serial.print("Jarak Sensor ");
    Serial.print(i);
    Serial.print(": ");
    Serial.print(jarak[i]);
    Serial.print(" | Stock: ");
    Serial.println(stock_barang[i]);
  }
  Serial.println();

  // kalkulasi flow tiap 1 detik
  for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
    if(stock_barang[i] > stock_barang_previous[i]){
      flow_masuk[i] += stock_barang[i] - stock_barang_previous[i];
      stock_barang_previous[i] = stock_barang[i];
    } else if(stock_barang[i] < stock_barang_previous[i]){

```

```

    flow_keluar[i] += stock_barang_previous[i] - stock_barang[i];
    stock_barang_previous[i] = stock_barang[i];
  }
}

// kirim data ke blynk
blynk_stream(stock_barang, BLYNK_PIN);
blynk_stream(flow_masuk, BLYNK_PIN_STOCK_FLOW_MASUK);
blynk_stream(flow_keluar, BLYNK_PIN_STOCK_FLOW_KELUAR);
}

void blynk_stream(int value[], const uint8_t pin[]){
  for(uint8_t i = 0; i < JUMLAH_SENSOR; i++){
    Blynk.virtualWrite(pin[i], value[i]);
  }
}

void blynk_connect(){
  Blynk.begin(BLYNK_TOKEN, AP_SSID, AP_PASS, "iot.serangkota.go.id",
8080);
}

String ipToString(IPAddress ipA){
  String s="";
  for (int i=0; i<4; i++)
    s += i ? "." + String(ipA[i]) : String(ipA[i]);
  return s;
}

```



```
float get_distance(const byte TRIGPIN, const byte ECHOPIN){
    long duration;
    float distance;
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);
    duration = pulseIn(ECHOPIN, HIGH);
    distance = duration * 0.034 / 2;
    return distance;
}

int kalkulasi_jumlah(float jarak, int tebal){
    int stock;
    if(jarak - margin > tebal){
        stock = KAPASITAS - ((int(jarak) - margin) / tebal);
    } else {
        stock = KAPASITAS;
    }
    return stock;
}
```