

REPUBLIC INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202302403, 9 Januari 2023

Pencipta

Nama : **MS Hendriyawan A, S.T., M.Eng., Ph.D., Dr. Eng. Arif Pramudwiatmoko, S.T., M.Eng. dkk**
Alamat : Daplokan, RT.001 RW.017, Margomulyo, Sayegan, Sleman, DI Yogyakarta, Sleman, DI YOGYAKARTA, 55561
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **MS Hendriyawan A, S.T., M.Eng., Ph.D., Dr. Eng. Arif Pramudwiatmoko, S.T., M.Eng. dkk**
Alamat : Daplokan, RT.001 RW.017, Margomulyo, Sayegan, Sleman, DI Yogyakarta, Sleman, DI YOGYAKARTA, 55561
Kewarganegaraan : Indonesia
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **KODE PROGRAM ARDUINO UNTUK AKUISISI DATA KUALITAS AIR DAN MENYIMPAN INFORMASI PENGUKURAN PADA CLOUD SERVER**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 28 November 2022, di Yogyakarta

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000435325

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n Menteri Hukum dan Hak Asasi Manusia
Direktur Jenderal Kekayaan Intelektual
u.b.
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto
NIP.196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	MS Hendriyawan A, S.T., M.Eng., Ph.D.	Daplokan, RT.001 RW.017, Margomulyo, Sayegan, Sleman, DI Yogyakarta
2	Dr. Eng. Arif Pramudwiatmoko, S.T., M.Eng.	Glagahombo, RT.002 RW.019, Girikerto, Turi, Sleman, DI Yogyakarta
3	Ari Sugiharto, S.Si., M.Eng.	Perum Nogotirto III B.5, RT.009 RW.017, Nogotirto, Gamping, Sleman, DI Yogyakarta

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	MS Hendriyawan A, S.T., M.Eng., Ph.D.	Daplokan, RT.001 RW.017, Margomulyo, Sayegan, Sleman, DI Yogyakarta
2	Dr. Eng. Arif Pramudwiatmoko, S.T., M.Eng.	Glagahombo, RT.002 RW.019, Girikerto, Turi, Sleman, DI Yogyakarta
3	Ari Sugiharto, S.Si., M.Eng.	Perum Nogotirto III B.5, RT.009 RW.017, Nogotirto, Gamping, Sleman, DI Yogyakarta, 55292



**KODE PROGRAM ARDUINO UNTUK AKUISISI DATA KUALITAS AIR
DAN MENYIMPAN INFORMASI PENGUKURAN PADA CLOUD SERVER**

OLEH:

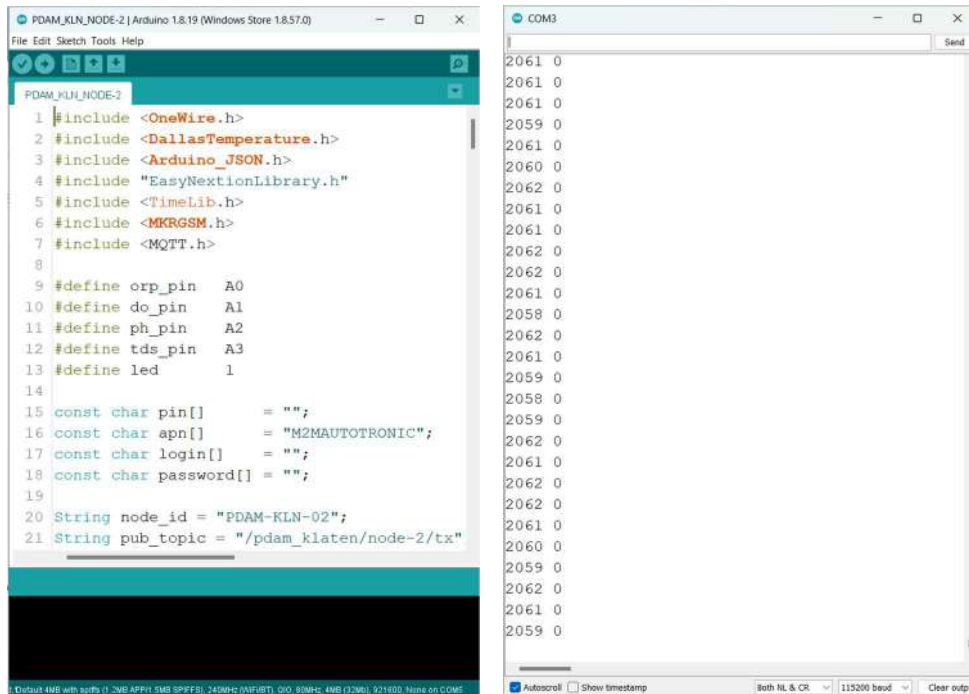
MS HENDRIYAWAN A, S.T., M.Eng., Ph.D.
Dr.Eng. ARIF PRAMUDWIATMOKO, S.T., M.Eng.
ARI SUGIHARTO, S.Si., M.Eng.

UNIVERSITAS TEKNOLOGI YOGYAKARTA
2022

SPEKIFIKASI SISTEM

1. Arduino-IDE

Kode Program dikembangkan menggunakan platform Arduino IDE yang menyediakan fasilitas penyusunan, kompilasi, hingga pengunggahan kode mesin menuju memori sistem tertanam papan Arduino. Gambar 1 menunjukkan tampilan Arduino IDE.



Gambar 1. Arduino IDE

2. Platform Arduino

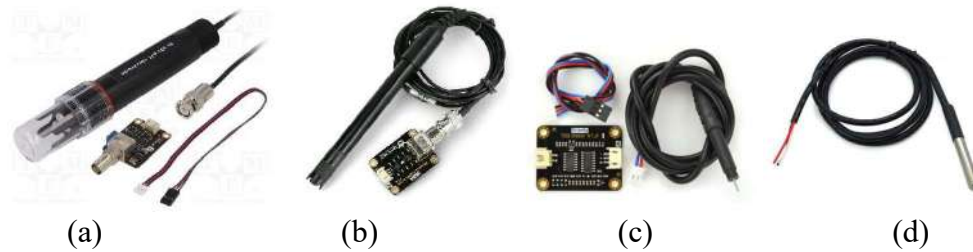
Sistem tertanam yang digunakan adalah Arduino MKR GSM 1400 yang menggunakan jaringan komunikasi nir-kabel GSM untuk koneksi dengan cloud server. Gambar 2 menunjukkan tampilan sistem tertanam yang menjalankan kode program.



Gambar 2. Arduino MKR GSM 1400

3. Sensor Kualitas Air

Sensor yang digunakan untuk akuisisi data kualitas air terbagi menjadi lima bagian, yaitu sensor: pH, DO, TDS, dan Suhu. Gambar 3 menunjukkan jenis-jenis sensor yang akan dibaca oleh kode program yang ditanamkan.



Gambar 3. Jenis-jenis sensor yang akan diakuisisi datanya. (a) Sensor pH. (b) Sensor DO. (c) Sensor TDS. (d) Sensor Suhu

4. Piranti IoT

Adalah penggabungan antara sistem antarmuka platform Arduino dengan elektronik sensor kualitas air yang akan menjalankan kode program, seperti ditunjukkan pada Gambar 4 berikut.



Gambar 4. Sistem elektronik terpadu untuk akuisisi data kualitas air

KODE PROGRAM

1. Deklarasi *Variable*

Bagian ini berfungsi selain untuk mendeklarasikan variabel statis juga mendeklarasikan library apa saja yang digunakan untuk menjalankan aplikasi IoT.

```
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3 #include <Arduino_JSON.h>
4 #include <TimeLib.h>
5 #include <MKRGSM.h>
6 #include <MQTT.h>
7
8 #define do_pin    A1
9 #define ph_pin    A2
10 #define tds_pin  A3
11 #define led      1
12
13 const char pin[]    = "";
14 const char apn[]    = "M2MAUTOTRONIC";
15 const char login[]  = "";
16 const char password[] = "";
17
18 String node_id = "PDAM-01";
19 String pub_topic = "/pdam/node-1/tx";
20 String times;
21 String jsonString;
22 char datasend[100];
23
24 float orp_val, do_val, ph_val, tds_val, suhu;
25 String hari, bulan;
26 char har[] = "";
27 char bln[] = "";
28 char jam[] = "";
29 char mnt[] = "";
30
31 unsigned long lastMillis = 0;
32
33 int counter = 0, pengali = 10;
34 float tot_tds, ave_tds = 0;
35 float tot_do, ave_do = 0;
36 float tot_ph, ave_ph = 0;
37 float tot_su, ave_su = 0;
38
```

2. Deklarasi *Instance*

Bagian ini menjelaskan *instance* yang dibutuhkan untuk menjalankan fungsi tertentu.

```
39 JSONVar myObject;
40 GSMClient net;
41 GPRS gprs;
42 GSM gsmAccess;
43 MQTTClient client;
44
45 OneWire oneWire(0);
46 DallasTemperature ds18b20(&oneWire);
47
```

3. Fungsi Koneksi dengan Jaringan GSM dan Cloud Server IoT

Bagian ini berfungsi untuk menghubungkan piranti IoT terhadap jaringan GSM dalam rangka pengiriman paket data menuju *cloud server*.

```
48 void connect() {
49   bool connected = false;
50   Serial.print("connecting to cellular network ...");
51   while (!connected) {
52     if ((gsmAccess.begin(pin) == GSM_READY) &&
53         (gprs.attachGPRS(apn, login, password) == GPRS_READY)) {
54       connected = true;
55     } else {
56       Serial.print(".");
57       delay(1000);
58     }
59   }
60   Serial.print("\nconnecting...");
61   while (!client.connect("", "", "")) {
62     Serial.print(".");
63     delay(1000);
64   }
65   Serial.println("\nconnected!");
66 }
67
```

4. Fungsi Koneksi dengan Jaringan GSM dan Cloud Server IoT

Bagian ini berfungsi untuk membaca waktu secara *online* pada server waktu global kemudian dikonversi menjadi waktu indonesia barat (WIB).

```
68 void baca_utc(){
69   time_t t = gsmAccess.getTime();
70   setTime(t);
71
72   //normalisasi UTC ke WIB
73   int wib = hour()+7;
74   if(wib > 23){
75     wib = wib - 24;
76   }
77   sprintf(har, "%02d", day());
78   sprintf(bln, "%02d", month());
79   sprintf(jam, "%02d", wib);
80   sprintf(mnt, "%02d", minute());
81
```

```
82 //ambil nama hari
83 switch (weekday()){
84     case 1:
85         hari = "Minggu";
86         break;
87     case 2:
88         hari = "Senin";
89         break;
90     case 3:
91         hari = "Selasa";
92         break;
93     case 4:
94         hari = "Rabu,";
95         break;
96     case 5:
97         hari = "Kamis";
98         break;
99     case 6:
100        hari = "Jum'at";
101        break;
102    case 7:
103        hari = "Sabtu";
104        break;
105 }
106
107 //ambil nama bulan
108 switch (month()){
109     case 1:
110         bulan = "Januari";
111         break;
112     case 2:
113         bulan = "Februari";
114         break;
115     case 3:
116         bulan = "Maret";
117         break;
118     case 4:
119         bulan = "April";
120         break;
121     case 5:
122         bulan = "Mei";
123         break;
124     case 6:
125         bulan = "Juni";
126         break;
127     case 7:
128         bulan = "Juli";
129         break;
130     case 8:
131         bulan = "Agustus";
132         break;
133     case 9:
134         bulan = "September";
135         break;
136     case 10:
137         bulan = "Oktober";
138         break;
139     case 11:
140         bulan = "November";
141         break;
142     case 12:
143         bulan = "Desember";
144         break;
145 }
146 times = String(har)+ String(bln) + String(year()-2000) + String(jam) + String(mnt);
147 }
148
```


5. Fungsi Akuisisi Data Analog Sensor pH-DO-TDS-SUHU

Bagian ini berfungsi untuk membaca output analog dari empat jenis sensor.

```
149 void baca_sensor(){
150   do_val = analogRead(do_pin) * (3300.0 / 1024.0);
151   ph_val = analogRead(ph_pin) * (3300.0 / 1024.0);
152   tds_val = analogRead(tds_pin) * (3300.0 / 1024.0);
153
154   ds18b20.requestTemperatures();
155   suhu = ds18b20.getTempCByIndex(0);
156
157   Serial.print(int(do_val));
158   Serial.print(' ');
159   Serial.print(int(ph_val));
160   Serial.print(' ');
161   Serial.print(int(tds_val));
162   Serial.print(' ');
163   Serial.println(suhu);
164 }
165
```

6. Fungsi Proses Inisialisasi (dilakukan sekali)

Bagian ini berfungsi untuk menginisialisasi sistem agar semua fungsi dapat digunakan dengan benar. Bagian ini dilakukan hanya sekali di awal saat sistem *microcontroller* mulai hidup.

```
166 void setup() {
167   Serial.begin(9600);
168   pinMode(led, OUTPUT);
169
170   for(int i=0;i<3;i++){
171     digitalWrite(led, HIGH);
172     delay(500);
173     digitalWrite(led, LOW);
174     delay(1000);
175   }
176
177   client.begin("103.161.185.242", net);
178   connect();
179 }
180
```

7. Fungsi Proses Utama (dilakukan berulang-ulang)

Bagian ini berfungsi untuk menjalankan proses utama yang menjadi tugas pokok sistem mikrocontroller sesuai dengan tujuannya, yaitu: membaca sensor dan mengirimkan data menuju *cloud server*.

```
181 void loop() {
182   client.loop();
183   if (!client.connected()) {
184     digitalWrite(led,LOW);
185     connect();
186   }
187   else {
188     digitalWrite(led,HIGH);
189   }
190
191   if (millis() - lastMillis > 60000) {
192     lastMillis = millis();
193
194     if (counter < pengali){
195       baca_utc();
196       baca_sensor();
197
198       tot_tds += tds_val;
199       tot_do += do_val;
200       tot_ph += ph_val;
201       tot_su += suhu;
202       counter += 1;
203
204       if (counter == pengali){
205         ave_tds = tot_tds / pengali;
206         ave_do = tot_do / pengali;
207         ave_ph = tot_ph / pengali;
208         ave_su = tot_su / pengali;
209
210         counter = 0;
211         tot_tds = 0;
212         tot_do = 0;
213         tot_ph = 0;
214         tot_su = 0;
215
216         myObject["a"] = times;
217         myObject["b"] = int(ave_do);
218         myObject["c"] = int(ave_ph);
219         myObject["d"] = int(ave_tds);
220         myObject["e"] = ave_su;
221
222         jsonString = JSON.stringify(myObject);
223
224         Serial.println(jsonString);
225         client.publish(pub_topic, jsonString);
226       }
227     }
228   }
229 }
```

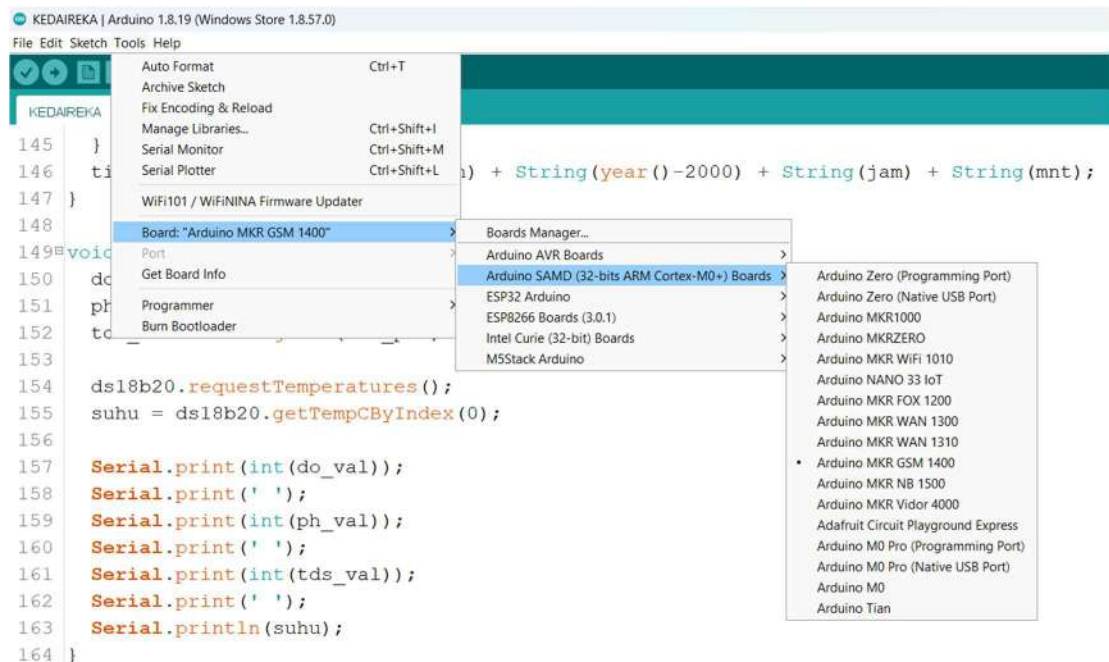
PETUNJUK PENGGUNAAN PROGRAM

1. Menjalankan Arduino IDE

Langkah pertama yang harus dijalankan adalah menuliskan script kode program menggunakan aplikasi Arduino IDE yang bisa diunduh melalui tautan: <https://www.arduino.cc/en/software>. Kemudian lakukan import library yang diperlukan dan ketik semua kode program seperti yang sudah dijelaskan pada bagian sebelumnya.

2. Memilih Platform Arduino

Langkah selanjutnya adalah memilih jenis platform Arduino yang akan menjalankan kode program, yaitu Arduino MKR GSM 1400. Gambar 5 menunjukkan menu pilihan platform Arduino.





Gambar 5. Menu pilihan platform Arduino MKR GSM 1400

3. Memilih Serial Port

Langkah selanjutnya adalah memilih serial port USB yang terhubung dengan platform Arduino dengan menyorot sub menu “**Port**”.

4. Kompilasi dan Unggah Kode Program

Langkah terakhir adalah kompilasi kode program dengan klik tombol  dan pastikan tidak ada notifikasi *error*. Kemudian unggah kode mesin (*.hex) dari hasil kompilasi dengan menekan tombol  hingga menunjukkan notifikasi berhasil mengunggah kode mesin tanpa ada notifikasi *error*. Selanjutnya platform Arduino akan *re-start* dan mulai menjalankan aplikasi berdasarkan kode program yang tertanam.