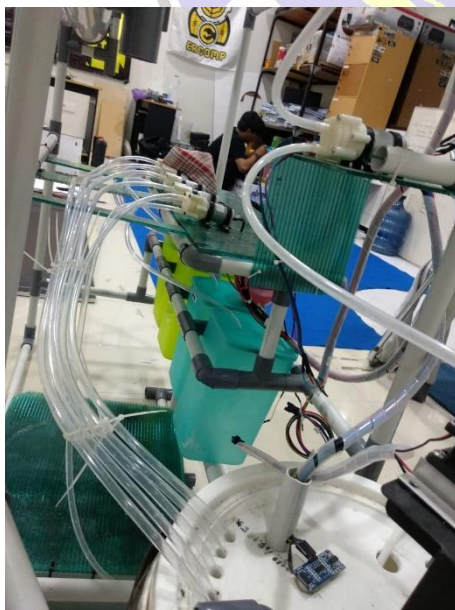
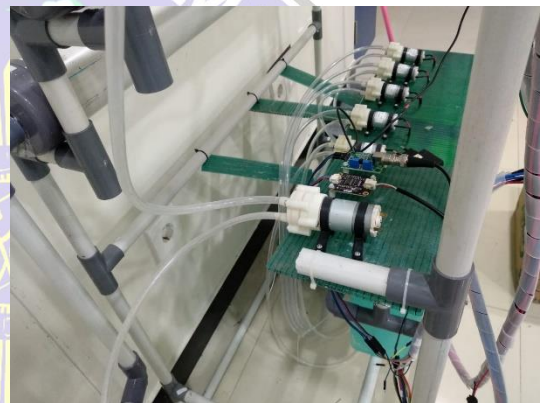
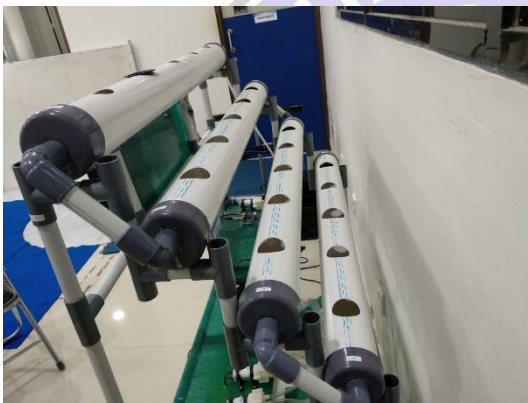
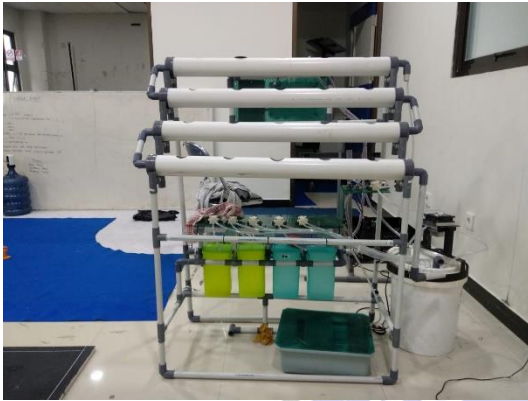


## LAMPIRAN

### GAMBAR ALAT



## PROGRAM SMART HIDROPONIK

---

```
1 //=====LCD I2C=====
2 #include <LiquidCrystal_I2C.h>
3 #include <OneWire.h>
4 LiquidCrystal_I2C lcd(0x3f, 16, 4);
5
6 //=====SENSOR KELEMBAPAN=====
7 #include "DHT.h"
8 #define DHT_PIN 3
9 #define DHTTYPE DHT11
10 #include <DHT.h>
11 DHT dht(DHT_PIN, DHTTYPE);
12
13 //=====SENSOR DS18B20=====
14 #include <DallasTemperature.h>
15 #define ONE_WIRE_BUS 5
16 OneWire oneWire(ONE_WIRE_BUS);
17 DallasTemperature ds18b20(&oneWire);
18
19 //=====SENSOR TDS=====
20 #define TdsSensorPin A0
21 #define VREF 5.0
22 #define SCOUNT 30
23
24 int analogBuffer[SCOUNT];
25 int analogBufferTemp[SCOUNT];
26 int analogBufferIndex = 0;
27 int copyIndex = 0;
28 float averageVoltage = 0;
29 float Nilai_PPM = 0;
30 float temperature = 16;
31 float compensationCoefficient = 1.0+0.02*(temperature-25.0);
32 float compensationVoltage=averageVoltage/compensationCoefficient;
33
34 // median filtering algorithm
35 int getMedianNum(int bArray[], int iFilterLen){
36     int bTab[iFilterLen];
37     for (byte i = 0; i<iFilterLen; i++)
38         bTab[i] = bArray[i];
39     int i, j, bTemp;
40     for (j = 0; j < iFilterLen - 1; j++) {
41         for (i = 0; i < iFilterLen - j - 1; i++) {
42             if (bTab[i] > bTab[i + 1]) {
43                 bTemp = bTab[i];
44                 bTab[i] = bTab[i + 1];
```

---

---

```

45     bTab[i + 1] = bTemp;
46     }
47     }
48     }
49     if ((iFilterLen & 1) > 0){
50     bTemp = bTab[(iFilterLen - 1) / 2];
51     }
52     else {
53     bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
54     }
55     return bTemp;
56     }
57 //=====SENSOR ULTRASONIK=====
58 const int echoPin = 6;
59 const int trigPin = 7;
60 long duration;
61 int Level_Air;
62 int safetyDistance;
63
64 //=====SENSOR PH=====
65 #define SensorPin A1
66 #define Offset 0.02
67 #define LED 13
68 #define samplingInterval 20
69 #define printInterval 800
70 #define ArrayLenth 40
71 int pHArray[ArrayLenth];
72 int pHArrayIndex=0;
73
74 //=====RELAY PENGENDALI=====
75 const int RelaypompanutrisiAB = 2;
76 const int RelaypompaPhUP = 12;
77 const int RelaypompaPhDOWN = 11;
78 const int RelaypompaBaku = 10;
79 const int Relaypendingin = 9;
80 const int Relaypengaduk = 8;
81
82 //=====NILAI PERBANDINGAN=====
83 const int TEMP_THRESHOLD_UPPER = 25;
84 const int TEMP_THRESHOLD_LOWER = 23;
85 const int TDS_THRESHOLD_UPPER = 1300;
86 const int TDS_THRESHOLD_LOWER = 1000;
87 const int PH_THRESHOLD_UPPER = 7.0;
88 const int PH_THRESHOLD_LOWER = 5.0;
89 const int DISTANCE_THRESHOLD_UPPER = -28;
90 const int DISTANCE_THRESHOLD_LOWER = -27;

```

---



---

```

91 //=====MODUL STEPPER=====
92 const int stepPin = 53;
93 const int dirPin = 51;
94 float Kelembapan, Suhu_Ruangan, Suhu_Air;
95 float Suhu = 25, PH_step;
96 static float Nilai_PH, voltage;
97
98 int ledRelaypompanutrisiAB;
99 int ledRelaypompaPhUP;
100 int ledRelaypompaPhDOWN;
101 int ledRelaypomabaku;
102 int ledRelaypendingin;
103 int ledRelaypengaduk;
104
105 void kirimdata() {
106
107 ledRelaypompanutrisiAB = digitalRead(32);
108 ledRelaypompaPhUP = digitalRead(30);
109 ledRelaypompaPhDOWN = digitalRead(28);
110 ledRelaypomabaku = digitalRead(26);
111 ledRelaypendingin = digitalRead(24);
112 ledRelaypengaduk = digitalRead(22);
113     digitalWrite(trigPin, LOW);
114     delayMicroseconds(2);
115     digitalWrite(trigPin, HIGH);
116     delayMicroseconds(10);
117     digitalWrite(trigPin,LOW);
118     Level_Air = (duration/2)*0.0342-36;
119     duration = pulseIn(echoPin,HIGH);
120     //safetyDistance = Level_Air;
121     Kelembapan = dht.readHumidity();
122     Suhu_Ruangan = dht.readTemperature();
123     Suhu_Air = ds18b20.getTempCByIndex(0);
124     ds18b20.setResolution(12);
125     ds18b20.requestTemperatures();
126     pHArray[pHArrayIndex++]=analogRead(SensorPin);
127     if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
128     voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
129     Nilai_PH = 3.5*voltage+Offset;
130
131 void setup() {
132 Serial.begin(9600);
133 lcd.init();
134 lcd.backlight();
135 lcd.begin(16,4);
136

```

---

---

```
137 dht.begin();
138 ds18b20.begin();
139
140 pinMode(TdsSensorPin,INPUT);
141 pinMode(echoPin, INPUT);
142 pinMode(trigPin, OUTPUT);
143 pinMode(32, OUTPUT);
144 pinMode(30, OUTPUT);
145 pinMode(28, OUTPUT);
146 pinMode(26, OUTPUT);
147 pinMode(24, OUTPUT);
148 pinMode(22, OUTPUT);
149 pinMode(stepPin, OUTPUT);
150 pinMode(dirPin, OUTPUT);
151 pinMode(LED, OUTPUT);
152 pinMode(Relaypompabaku, OUTPUT);
153 pinMode(Relaypendingin, OUTPUT);
154 pinMode(RelaypompanutrisiAB, OUTPUT);
155 pinMode(RelaypompaPhDOWN, OUTPUT);
156 pinMode(RelaypompaPhUP, OUTPUT);
157 pinMode(Relaypengaduk, OUTPUT);
158
159 digitalWrite(Relaypompabaku, HIGH);
160 digitalWrite(Relaypendingin, HIGH);
161 digitalWrite(RelaypompanutrisiAB, HIGH);
162 digitalWrite(RelaypompaPhDOWN, HIGH);
163 digitalWrite(RelaypompaPhUP, HIGH);
164 digitalWrite(Relaypengaduk, HIGH);
165 digitalWrite(32, HIGH);
166 digitalWrite(30, HIGH);
167 digitalWrite(28, HIGH);
168 digitalWrite(26, HIGH);
169 digitalWrite(24, HIGH);
170 digitalWrite(22, HIGH);
171
172 {
173 lcd.clear();
174 lcd.setCursor(6,1);
175 lcd.print("SMART");
176 lcd.setCursor(-1,2);
177 lcd.print("HIDROPONIK");
178 delay(500);
179 }
180 }
181
182 void loop() {
```

---

---

```

183
184 ledRelaypompanutrisiAB = digitalRead(32);
185 ledRelaypompaPhUP    = digitalRead(30);
186 ledRelaypompaPhDOWN  = digitalRead(28);
187 ledRelaypompaBaku    = digitalRead(26);
188 ledRelaypendingin    = digitalRead(24);
189 ledRelaypengaduk     = digitalRead(22);
190
191 {
192   {
193   lcd.clear();
194   Serial.print("SUHU AIR NUTRISI : ");
195   Serial.println(Suhu_Air,1);
196   lcd.setCursor(0, 1);
197   lcd.print("TEMP :");
198   lcd.setCursor(7, 1);
199   lcd.print(Suhu_Air, 1);
200   lcd.print((char)223);
201   lcd.print("C");
202
203   Serial.print("NILAI TDS      : ");
204   Serial.println(Nilai_PPM,0);
205   lcd.setCursor(-4, 2);
206   lcd.print("PPM:");
207   lcd.setCursor(1, 2);
208   lcd.print(Nilai_PPM,0);
209   lcd.print(" ");
210
211   Serial.print("NILAI PH      : ");
212   Serial.println(Nilai_PH,1);
213   lcd.setCursor(5, 2);
214   lcd.print("PH:");
215   lcd.setCursor(9, 2);
216   lcd.print(Nilai_PH,1);
217   lcd.print(" ");
218
219   Serial.print("LEVEL AIR NUTRISI : ");
220   Serial.println(Level_Air);
221   lcd.setCursor(0,0);
222   lcd.print("LEVEL: ");
223   lcd.setCursor(7,0);
224   lcd.print(Level_Air);
225   lcd.print(" Cm");
226
227   Serial.print("KELEMBAPAN    : ");
228   Serial.println(Kelembapan);

```

---

---

```

229 lcd.setCursor(-4,3);
230 lcd.print("LMB/TMP: ");
231 lcd.setCursor(5,3);
232 lcd.print(Kelembapan,0);
233 //lcd.print((char)223);
234 //lcd.print("C");
235
236 Serial.print("SUHU RUANGAN  :");
237 Serial.println(Suhu_Ruangan);
238 lcd.setCursor(7,3);
239 lcd.print("/");
240 lcd.setCursor(8,3);
241 lcd.print(Suhu_Ruangan,0);
242 lcd.print((char)223);
243 lcd.print("C");
244 Serial.println();
245 delay(500);
246 }
247
248 //=====SENSOR ULTRASONIK=====
249 if(Level_Air <= DISTANCE_THRESHOLD_UPPER)
250 {
251   digitalWrite(26, LOW);
252   lcd.clear();
253   digitalWrite(Relaypompabaku, HIGH);
254 }
255 else if(Level_Air >= DISTANCE_THRESHOLD_LOWER)
256 {
257   digitalWrite(26, HIGH);
258   lcd.clear();
259   digitalWrite(Relaypompabaku, LOW);
260   lcd.setCursor(1,1);
261   lcd.print("POMPA AIR BAKU");
262   lcd.setCursor(2,2);
263   lcd.print("(ON)");
264   delay(2000);
265   digitalWrite(Relaypompabaku, HIGH);
266   delay(3000);
267 }
268
269 //=====SENSOR DS18B20=====
270 if(Suhu_Air >= TEMP_THRESHOLD_UPPER)
271 {
272   digitalWrite(24, HIGH);
273   lcd.clear();
274   digitalWrite(Relaypendingin, LOW);

```

---



---

```

275  lcd.setCursor(4,1);
276  lcd.print("PENDINGIN");
277  lcd.setCursor(2,2);
278  lcd.print("(ON)");
279  delay(1000);
280  }
281  else if(Suhu_Air <= TEMP_THRESHOLD_LOWER)
282  {
283  digitalWrite(24, LOW);
284  lcd.clear();
285  digitalWrite(Relaypendingin, HIGH);
286  }
287
288  //=====SENSOR TDS=====
289  if(Nilai_PPM >= TDS_THRESHOLD_UPPER)
290  {
291  digitalWrite(32, HIGH);
292  digitalWrite(22, HIGH);
293  lcd.clear();
294  digitalWrite(RelaypompanutrisiAB, HIGH);
295  digitalWrite(Relaypengaduk, HIGH);
296  }
297  else if(Nilai_PPM <= TDS_THRESHOLD_LOWER)
298  {
299  digitalWrite(32, LOW);
300  digitalWrite(22, LOW);
301  lcd.clear();
302  digitalWrite(RelaypompanutrisiAB, LOW);
303  lcd.setCursor(0,1);
304  lcd.print("POMPA NUTRISI AB");
305  lcd.setCursor(2,2);
306  lcd.print("(ON)");
307  delay(500);
308  digitalWrite(RelaypompanutrisiAB, HIGH);
309  delay(5000);
310  }
311
312  //=====SENSOR PH=====
313  if(Nilai_PH >= PH_THRESHOLD_UPPER)
314  {
315  digitalWrite(28, HIGH);
316  lcd.clear();
317  digitalWrite(RelaypompaPhDOWN, LOW);
318  lcd.setCursor(2,1);
319  lcd.print("POMPA PH DOWN");
320  lcd.setCursor(2,2);

```

---



---

```
321 lcd.print("(ON)");
322 delay(2000);
323 digitalWrite(RelaypompaPhDOWN, HIGH);
324 delay(3000);
325 digitalWrite(RelaypompaPhUP, HIGH);
326 }
327 else if(Nilai_PH <= PH_THRESHOLD_LOWER)
328 {
329 digitalWrite(30, HIGH);
330 lcd.clear();
331 digitalWrite(RelaypompaPhUP, LOW);
332 lcd.setCursor(2,1);
333 lcd.print("POMPA PH UP");
334 lcd.setCursor(2,2);
335 lcd.print("(ON)");
336 delay(2000);
337 digitalWrite(RelaypompaPhUP, HIGH);
338 delay(3000);
339 digitalWrite(RelaypompaPhDOWN, HIGH);
340 }
341 else
342 {
343 digitalWrite(28, LOW);
344 digitalWrite(30, LOW);
345 lcd.clear();
346 digitalWrite(RelaypompaPhDOWN, HIGH);
347 digitalWrite(RelaypompaPhUP, HIGH);
348 }
349 }
350 }
```

---

