

BAB IV

PEMBUATAN ALAT

4.1 Pembuatan Prototipe

Pembuatan prototipe merupakan bentuk awal atau model dasar dari sebuah sistem yang dirancang sebelum diaplikasikan lebih lanjut dan merupakan salah satu tahapan yang penting dalam pengembangan alat. Prototipe berfungsi untuk pengujian fungsionalitas, mengevaluasi efektivitas rancangan, serta memastikan alat yang bekerja sesuai dengan tujuan. Pada tugas akhir ini, prototipe difokuskan pada aspek perancangan dan pembangunan perangkat keras (*Hardware*) yang mampu mensimulasikan bagaimana sistem buka tutup pintu irigasi persawahan dapat bekerja dalam kondisi nyata dengan ukuran yang lebih kecil. Dengan mendeteksi ketinggian air menggunakan *sensor water level (float switch)*, data yang diperoleh dari sensor tersebut digunakan sebagai dasar pengambilan keputusan untuk mengontrol sistem pintu irigasi yang dapat membuka dan menutup secara otomatis. Pada sistem ini, solenoid valve dirancang hanya akan bekerja ketika mikrokontroler mendeteksi level air yang dibaca oleh sensor. Ketika air pada lahan persawahan berada diatas rata-rata ketinggian, maka akan memerintah ke sistem sehingga solenoid valve otomatis menutup pintu irigasi dan begitupun juga sebaliknya, jika air pada lahan persawahan belum sesuai atau masih dalam posisi LOW, maka solenoid valve otomatis akan membuka pintu irigasi. Pada proses pembuatan alat tugas akhir ini terbagi menjadi 2 tahapan yaitu:

a. Pembuatan Perangkat Keras

Pembuatan perangkat keras merupakan proses perancangan, pengembangan, dan realisasi komponen fisik dari suatu sistem yang akan dirancang. Perangkat keras merujuk pada bagian-bagian nyata dari suatu sistem yang dapat dilihat atau disentuh secara langsung, seperti contohnya mikrokontroler, sensor, PCB, aktuator, serta berbagai komponen elektronik pendukung lainnya. Tujuan utama dari pembuatan perangkat keras adalah untuk menghasilkan sistem yang mampu menjalankan fungsi sesuai dengan tujuan yang diinginkan. Seperti pada perancangan tugas akhir ini yaitu untuk sistem kontrol otomatis pada pintu irigasi

persawahan. Tahapan dalam perangkat keras umumnya dimulai dari mengidentifikasi kebutuhan sistem, perancangan awal yang mencakup pembuatan diagram blok dan pemilihan komponen, hingga pembuatan skematik rangkaian serta layout PCB. Selain itu juga dilakukan perakitan komponen pada papan rangkaian, pemrograman mikrokontroler sebagai dasar sistem otomatis serta pengujian terhadap perancangan perangkat keras (*hardware*) untuk memastikan bahwa sistem bekerja sesuai dengan yang dirancang. Maka dari itu, pembuatan perangkat keras (*hardware*) menjadi salah satu bagian penting dalam pengembangan alat tugas akhir ini karena menentukan keberhasilan fungsi dan efektivitas dari alat tersebut dalam aplikasi sesungguhnya.

b. Pembuatan Perangkat Lunak

Pembuatan perangkat lunak merupakan salah satu hal yang sangat penting dalam suatu sistem mikrokontroler karena berisi langkah-langkah kerja secara sistematis yang sesuai dengan bahasa pemrograman yang digunakan. Pembuatan perangkat lunak mencakup berbagai aspek penting seperti perencanaan, pengkodean, pengujian, serta penerapan dan pemeliharaan terhadap kode program yang dioperasikan oleh perangkat keras untuk menyelesaikan tugas tertentu. Tujuan dari pembuatan perangkat lunak adalah untuk menyediakan solusi terhadap permasalahan tertentu, mengotomatisasi proses, serta meningkatkan efisiensi dan produktivitas pengguna. Pengembangan perangkat lunak dalam pembuatan tugas akhir ini adalah kontrol otomatis sistem pintu irigasi yang akan membuka dan menutup secara otomatis. Sistem ini menggunakan mikrokontroler Arduino Nano sebagai unit kendali utama yang mengirimkan sinyal untuk mengaktifkan aktuator.

4.2 Pembuatan Perangkat Keras (Hardware)

Pada tahap pembuatan perangkat keras, terdapat beberapa proses penting yang harus dilalui, antara lain yaitu menganalisis teknis untuk merumuskan spesifikasi sistem, perancangan desain alat, serta perakitan rangkaian elektronik dan mekanik. Dalam pembuatan prototipe sistem kendali pintu irigasi otomatis, mikrokontroler Arduino Nano menjadi salah satu komponen utama atau pusat kendali utama. Selain itu komponen-komponen lain yang dibutuhkan masing-masing dipilih berdasarkan performa, kompatibilitas, efisiensi daya dan kesesuaian terhadap kebutuhan sistem

yang dirancang. Dengan ini, semua tahapan yang dilakukan untuk memastikan bahwa perangkat keras yang dihasilkan dapat bekerja secara optimal, stabil, dan sesuai dengan tujuan fungsional dari prototipe tersebut.

4.2.1 Alat dan Bahan

Dalam pembuatan alat tugas akhir ini dibutuhkan beberapa komponen yang digunakan untuk perancangan sistem. Menentukan kebutuhan tersebut merupakan langkah krusial dalam tahap pengembangan, dengan tujuan untuk menyusun daftar lengkap komponen dan sumber daya yang diperlukan. Dengan demikian dalam pembuatan proyek tugas ini dapat meminimalkan kekurangan material, menghemat biaya dan waktu serta menurunkan potensi kegagalan pada saat tahap implementasi. Langkah awal dari pembuatan proyek ini yaitu menyiapkan alat serta bahan yang dibutuhkan. Adapun bahan yang digunakan seperti pada tabel 4.1.

Tabel 4. 1 Bahan yang digunakan

No.	Nama Bahan	Jumlah
1.	Sensor Level Air (Float Switch)	4 buah
2.	Sensor Level Air	1 buah
3.	Arduino Nano	1 buah
4.	Solenoid Valve	4 buah
5.	Pompa Air	1 buah
6.	Pipa PVC	7 buah
7.	Power Supply Switching	1 buah
8.	Buck Konverter MP1584EN	1 buah
9.	LCD	1 buah
10.	Resistor	14 buah
11.	MOSFET irfz446	4 buah
12.	Akrilik	1 buah
13.	Box Komponen	1 buah
14.	Tangki Bak	1 buah
15.	Papan 72 cm x 39 cm	1 buah

Selanjutnya yaitu alat yang digunakan dalam perancangan sistem yang telah dibuat seperti pada tabel 4.2.

Tabel 4. 2 Alat yang digunakan

No.	Nama Alat	Jumlah
1.	Set Obeng	1 buah
2.	Multimeter	1 buah
3.	Solder	1 buah
4.	Cutter	1 buah
5.	Lem	1 buah
6.	Tang Kupas	1 buah
7.	Amplas Kasar	1 buah
8.	Alat Tulis	1 buah
9.	Penggaris	1 buah

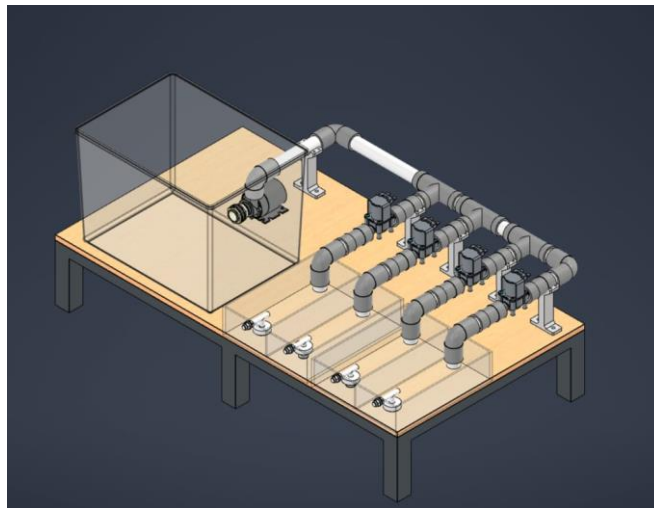
4.2.2 Langkah-Langkah Pembuatan Perangkat Keras

4.2.2.1 Perancangan Mekanik

Dalam perancangan mekanik pembuatan alat tugas akhir yang berjudul “Rancang Bangun Pintu Irigasi Otomatis Pada Lahan Persawahan Padi Menggunakan *Sensor Water Level*” proses ini berkaitan langsung dengan bentuk fisik, kekuatan struktur, serta mekanisme penggerak pintu irigasi agar alat dapat berfungsi secara optimal. Struktur utama alat ini dirancang menggunakan pipa sebagai jalur aliran air dalam proses irigasi dari sungai ke sawah dan akrilik yang difungsikan sebagai representasi untuk lahan persawahan. Material tersebut dipilih karena memiliki kekuatan yang cukup namun tetap mudah untuk dibentuk dan dirakit. Pada tahap ini, acuan yang digunakan meliputi desain dari alat sejenis serta aspek ergonomi dan kemudahan dalam perawatan.

Desain prototipe alat ini dirancang untuk mempermudah visualisasi keseluruhan alat serta untuk memastikan kesesuaian dimensi antar komponen. Pada bagian akrilik digunakan untuk penempatan *sensor water level (float switch)* yang akan mendeteksi ketinggian air. Sensor ini akan terhubung dengan sistem kontrol

otomatis yang akan mengatur buka-tutup pintu irigasi secara responsif sesuai dengan kondisi ketinggian air. Pada tangki besar yang akan menampung sumber air utama diletakkan satu buah pompa air. Terdapat pipa utama yang bercabang menjadi empat jalur dimana masing-masing dilengkapi dengan satu katup solenoid valve yang mengarah ke petak sawah. Dengan adanya desain ini diharapkan seluruh sistem dapat dibangun secara efisien dan sesuai dengan rancangan awal yang telah direncanakan. Tampilan Desain 3D dapat dilihat pada gambar 4.1.



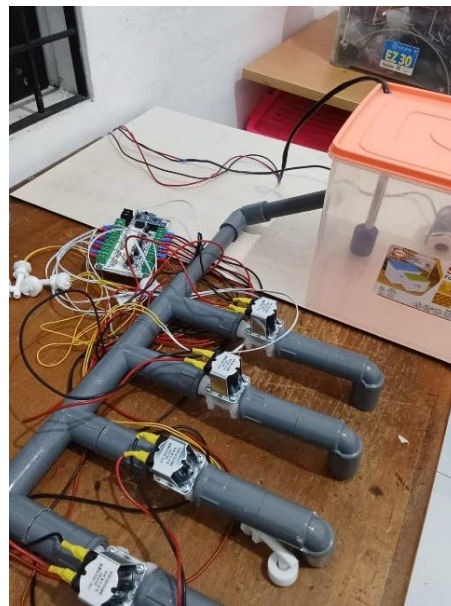
Gambar 4. 1 Desain 3D Rancang Bangun Pintu Irigasi Otomatis

Sesuai dengan desain 3D yang telah dirancang sebelumnya seperti yang telah ditunjukkan pada gambar 4.1 di atas, pada gambar tersebut memperlihatkan desain prototipe sistem kendali pintu irigasi pada persawahan. Desain tersebut mencakup tampilan bak plastik yang direpresentasikan sebagai sungai atau sumber utama air. Di dalam bak tersebut diletakkan komponen seperti *sensor water level* untuk mendeteksi air agar suplai dapat dikontrol dan *water pump* yang berfungsi untuk menaikkan tekanan air dari bak menuju pipa utama agar bisa dialirkan secara stabil. Ditunjukkan seperti pada gambar 4.2 di bawah ini.



Gambar 4. 2 Tampilan Tangki Bak

Selanjutnya pemasangan pipa PVC menggunakan sambungan berbentuk tee dan elbow untuk jalan pengaliran air dimana terdiri dari empat cabang yang terpasang solenoid valve seperti pada gambar 4.3. Setelah rangkaian pipa terbentuk, manifold dipasang menggunakan bracket penyangga yang telah dicetak pada desain 3D pada setiap pipa sehingga dapat terpasang dengan kencang dan tidak goyang seperti gambar 4.5 di bawah ini.

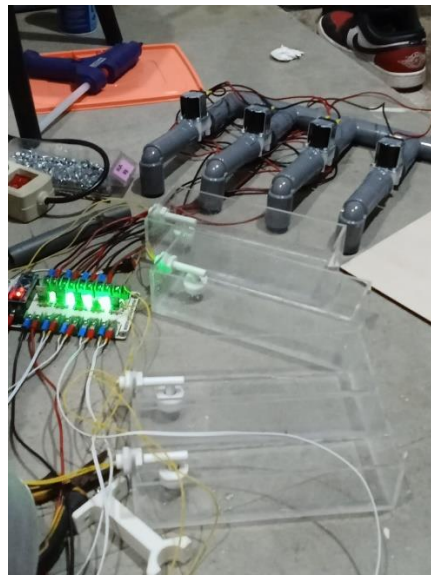


Gambar 4. 3 Perakitan pipa PVC dan Selonoid Valve

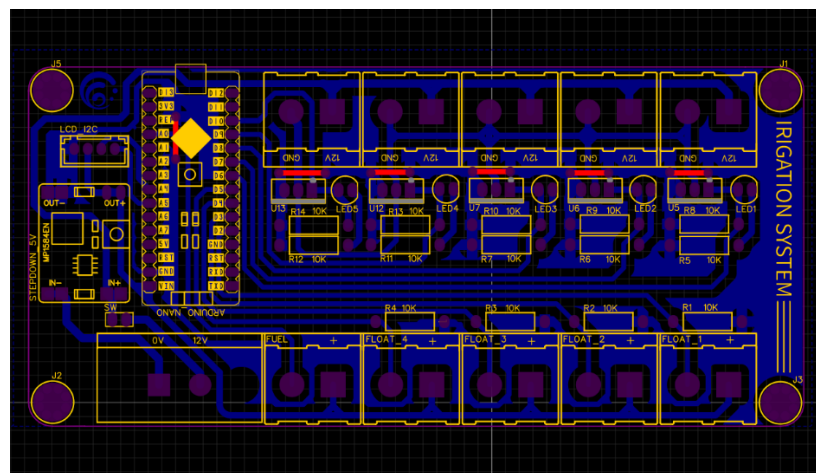


Gambar 4. 4 Pembuatan Bracket Penyangga

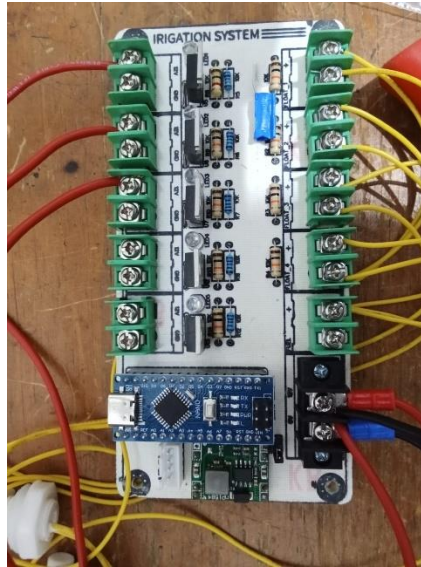
Selanjutnya pembentukan akrilik yang direpresentasikan sebagai sawah dalam perancangan alat tugas akhir ini yaitu rancang bangun pintu irigasi otomatis. Dimana setiap kotak sawah terdapat *sensor water level (float switch)* seperti yang ditunjukkan gambar 4.5 di bawah ini.



Gambar 4. 5 Pembentukan akrilik dan pemasangan Sensor Float Switch



Komponen-komponen yang terdapat pada gambar 4.7 yaitu, mikrokontroler Arduino nano sebagai pusat utama kendali, driver MOSFET sebagai saklar elektronika untuk mengendalikan aktuator, serta resistor sebagai pembatas arus serta pengatur tegangan sinyal, dan terminal konektor untuk penghubung kabel yang berasal dari sensor, pompa serta sumber daya utama. Setelah itu dilakukan proses layout PCB, yaitu dengan menata posisi komponen di jalur tembaga pada papan sirkuit. Komponen besar seperti terminal konektor ditempatkan ditepi papan agar mudah diakses, sedangkan Arduino Nano ditempatkan di tengah bawah sebagai pusat kendali. Jalur daya dibuat lebih tebal agar mampu menghantarkan arus yang besar menuju solenoid valve, sedangkan jalur sinyal dibuat kecil karena hanya membawa arus logika dan mikrokontroler. Tahap selanjutnya yaitu melakukan penyolderan komponen secara hati-hati sesuai arah kaki dan jalur yang telah ditentukan agar tidak terjadi hubungan pendek antar jalur .



Gambar 4. 7 Tampilan PCB Komponen

Penggunaan box berwarna biru seperti gambar 4.8 sebagai peletakan papan PCB sistem irigasi yang berfungsi untuk melindungi rangkaian elektronik dari debu dan kelembapan. Pada bagian box ini dilengkapi dengan modul LCD yang berfungsi untuk menampilkan informasi atau status sistem seperti status pompa, sensor air atau level air.



Gambar 4. 8 Box PCB

Langkah terakhir dari pembuatan perangkat keras adalah melakukan pengecekan menyeluruh untuk memastikan semua komponen terpasang dengan benar. Pengecekan dilakukan untuk memastikan tidak ada kesalahan pemasangan atau kebocoran pada sistem. Gambar 4.9 menunjukkan tampilan keseluruhan alat.



Gambar 4. 9 Keseluruhan Alat

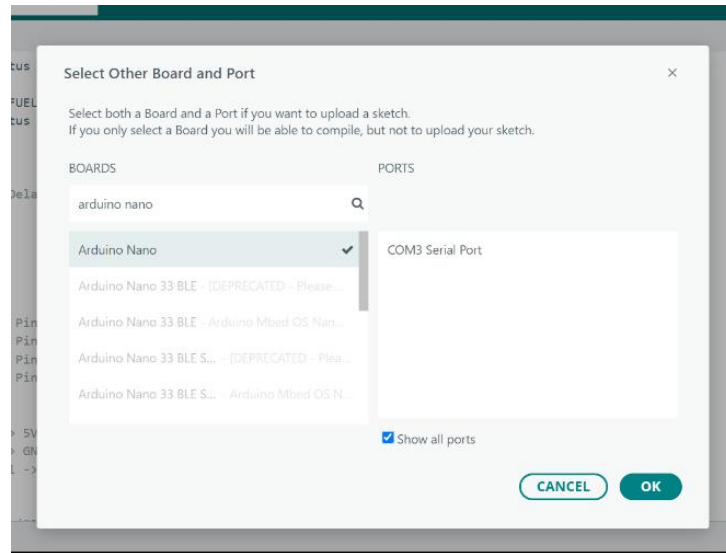
4.3 Pembuatan Perangkat Lunak (Software)

Pembuatan perangkat lunak pada sistem irigasi otomatis, dimulai dengan pembuatan perancangan logika kerja sistem yang akan mengatur hubungan antar sensor, miktokontroler dan akuator yang dikembangkan menggunakan bahasa pemrograman Arduino IDE. Sebuah alat pemrograman sumber terbuka yang dapat digunakan dengan berbagai tipe mikrokontroler termasuk Arduino Nano. Perangkat lunak ini dibuat untuk mengelola logika operasional sistem, mulai dari membaca data sensor ketinggian air, memproses data, hingga mengendalikan akuator seperti solenoid valve dan pompa. Seluruh proses pemrograman dilakukan dengan memperhatikan efisiensi sistem, akurasi respons, dan kemampuan untuk berinteraksi secara handal dengan semua komponen perangkat keras yang terlibat. Tahapan berikut ini menjelaskan langkah-langkah dalam pembuatan program menggunakan Arduino IDE.

4.3.1 Pembuatan Program Pada Arduino IDE

Pada tahap ini, perangkat lunak digunakan untuk mengendalikan sistem kendali pintu irigasi otomatis yang dirancang menggunakan Arduino IDE. Proses ini diawali dengan membuka aplikasi Arduino IDE yang telah diinstal pada laptop.

Selanjutnya dilakukan konfigurasi awal dengan memilih board yang sesuai yaitu Arduino Nano melalui Select Board dan menentukan port komunikasi (COM port) yang terhubung dengan perangkat. Seperti ditunjukkan pada gambar 4.10.



Gambar 4. 10 Pemilihan Board dan port Arduino Nano

Setelah mengatur *board* pada Arduino IDE, langkah selanjutnya yaitu mengunduh *library* yang akan digunakan untuk mendukung kerja komponen pada sistem. Pada alat ini menggunakan *library* seperti LiquidCrystal.I2C.h.

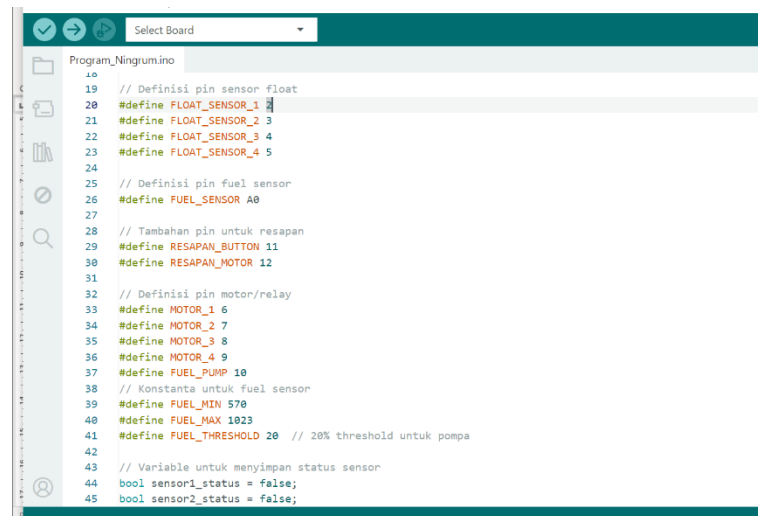
```

14  #include <Wire.h>
15  #include <LiquidCrystal_I2C.h>
16

```

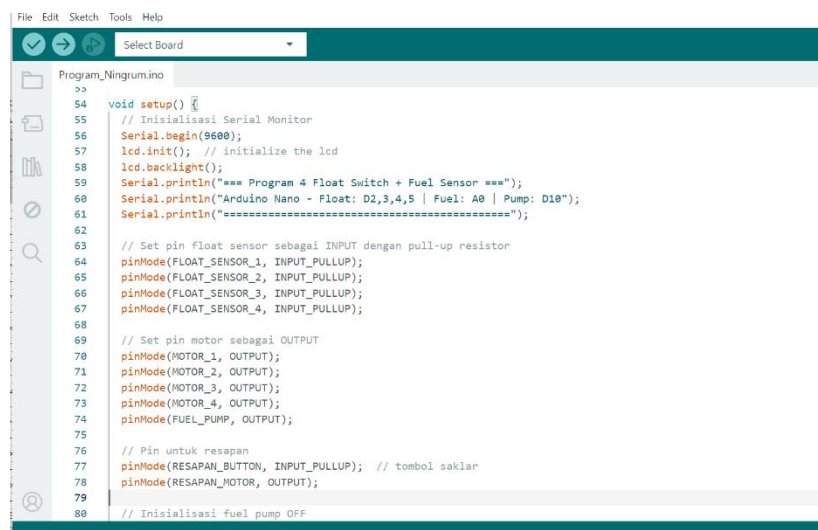
Gambar 4. 11 Tampilan Library pada Arduino IDE

Setelah itu, dilanjutkan dengan melakukan deklarasi pin pada program Arduino IDE untuk mengetahui setiap sensor dan actuator yang terhubung dengan mikrokontroler. Seperti pada gambar 4.12 menunjukkan penentuan pin modul pada Arduino IDE.



Gambar 4. 12 Pin Modul pada Arduino IDE

Selanjutnya melakukan pemrograman void setup() pada Arduino IDE seperti pada gambar 4.13. Bagian void setup() di Arduino IDE digunakan untuk menginisialisasi pengaturan awal sistem dan hanya dijalankan sekali pada saat pertama kali sistem diidupkan atau direset. Pada tahap ini komunikasi serial diaktifkan, serta modul-modul sensor dikonfigurasi agar siap digunakan.



```

90 // Inisialisasi fuel pump OFF
91 digitalWrite(FUEL_PUMP, LOW);
92
93 Serial.println("System Ready!");
94 Serial.println("Float Format: S1|S2|S3|S4 (HIGH=Tidak Terendam, LOW=Terendam)");
95 Serial.println("Fuel Format: Raw|Percentage|Pump Status");
96 Serial.println("Fuel Range: 883-899 -> 0-100% | Pump ON >20%, OFF <20%");
97 Serial.println("=====");
98
99 lcd.setCursor(0, 0);
100 lcd.print("System Start...");
101 delay(1000);
102 lcd.clear();
103
104
105

```

Gambar 4. 13 Tampilan Void Setup pada Arduino IDE

Setelah void setup() selesai dijalankan, selanjutnya akan masuk pada fungsi void loop() yang merupakan inti utama dari pemrograman yang akan bekerja untuk menjalankan logika sistem secara berulang-ulang selama alat dalam keadaan menyala. Pemrograman dapat dilihat pada gambar 4.13

```

94
95 void loop() {
96 // ===== CEK MODE RESAPAN =====
97 if (digitalRead(RESAPAN_BUTTON) == LOW) { // Tombol resapan air aktif (karena pull-up)
98   Serial.println("!!! RESAPAN MODE AKTIF !!!");
99
100   // Matikan semua sistem utama
101   digitalWrite(MOTOR_1, LOW);
102   digitalWrite(MOTOR_2, LOW);
103   digitalWrite(MOTOR_3, LOW);
104   digitalWrite(MOTOR_4, LOW);
105   digitalWrite(FUEL_PUMP, LOW);
106
107   // Jalankan motor resapan air selama tombol ditekan
108   digitalWrite(RESAPAN_MOTOR, HIGH);
109
110   // Selama tombol masih ditekan, sistem lain berhenti total
111   while (digitalRead(RESAPAN_BUTTON) == LOW) {
112     Serial.println(">>> Resapan air aktif - semua sistem berhenti <<<");
113     delay(500);
114   }
115
116   // Setelah tombol dilepas
117   digitalWrite(RESAPAN_MOTOR, LOW);
118   Serial.println("!!! RESAPAN AIR MODE NONAKTIF - Sistem kembali normal !!!");
119 }
120
121
122 // ===== BACA FLOAT SENSORS =====
123 sensor1_status = digitalRead(FLOAT_SENSOR_1);
124 sensor2_status = digitalRead(FLOAT_SENSOR_2);
125 sensor3_status = digitalRead(FLOAT_SENSOR_3);
126 sensor4_status = digitalRead(FLOAT_SENSOR_4);
127
128 // ===== BACA FUEL SENSOR =====
129 fuel_raw = analogRead(FUEL_SENSOR);
130
131 // Mapping nilai 883-899 ke 0-100%
132 fuel_percentage = map(fuel_raw, FUEL_MIN, FUEL_MAX, 100, 0);
133
134 // Batasi nilai percentage antara 0-100
135 if (fuel_percentage < 0) fuel_percentage = 0;
136 if (fuel_percentage > 100) fuel_percentage = 100;
137
138
139 // ===== TAMPILKAN STATUS FLOAT SENSORS =====
140 Serial.print("Float - S1:");
141 Serial.print(sensor1_status ? "HIGH" : "LOW ");
142 Serial.print(" | S2:");
143 Serial.print(sensor2_status ? "HIGH" : "LOW ");
144 Serial.print(" | S3:");
145 Serial.print(sensor3_status ? "HIGH" : "LOW ");
146 Serial.print(" | S4:");
147 Serial.print(sensor4_status ? "HIGH" : "LOW ");

```

```

148
149 // Hitung jumlah sensor yang terendam (status LOW)
150 int sensors_active = 0;
151 if (!sensor1_status) sensors_active++;
152 if (!sensor2_status) sensors_active++;
153 if (!sensor3_status) sensors_active++;
154 if (!sensor4_status) sensors_active++;
155
156 Serial.print(" | Terendam: ");
157 Serial.print(sensors_active);
158 Serial.print("/4");
159
160 // ===== TAMPILKAN STATUS FUEL SENSOR =====
161 Serial.print(" | Fuel - Raw:");
162 Serial.print(fuel_raw);
163 Serial.print(" | ");
164 Serial.print(fuel_percentage, 1);
165 Serial.print(" | Pump:");
166 Serial.print(fuel_pump_status ? "ON " : "OFF");
167
168 // Alert jika fuel rendah
169 if (fuel_percentage <= FUEL_THRESHOLD) {
170   Serial.print(" | *** FUEL LOW ***");
171 }
172
173 Serial.println();
174
175 // ===== KONTROL MOTOR FLOAT SWITCHES =====
176 digitalWrite(MOTOR_1, !sensor1_status); // Motor ON jika sensor LOW (terendam)
177 digitalWrite(MOTOR_2, !sensor2_status); // Motor ON jika sensor LOW (terendam)
178 digitalWrite(MOTOR_3, !sensor3_status); // Motor ON jika sensor LOW (terendam)
179 digitalWrite(MOTOR_4, !sensor4_status); // Motor ON jika sensor LOW (terendam)
180
181 // ===== ALERT JIKA SEMUA FLOAT SENSOR TERENDAM =====
182 if (sensors_active == 0) {
183   Serial.println(" SEMUA WADAH PENUH ");
184   digitalWrite(FUEL_PUMP, LOW); // Pompa OFF jika fuel <= 20%
185   fuel_pump_status = false;
186   // Alert di Serial Monitor saja
187 } else {
188   // ===== KONTROL FUEL PUMP =====
189   if (fuel_percentage > FUEL_THRESHOLD) {
190     digitalWrite(FUEL_PUMP, HIGH); // Pompa ON jika fuel > 20%
191     fuel_pump_status = true;
192   } else {
193     digitalWrite(FUEL_PUMP, LOW); // Pompa OFF jika fuel <= 20%
194     fuel_pump_status = false;
195   }
196 }
197

```

Gambar 4. 14 Tampilan Void Loop pada Arduino IDE

Untuk berikutnya yaitu membuat program pada tampilan LCD seperti pada gambar 4.14. Dengan adanya tampilan ini pengguna dapat mengetahui kondisi sistem secara tepat karena program mampu menampilkan data real time, status sensor dan pompa air secara visual pada layar LCD.

```

File Edit Sketch Tools Help
Select Board

Program_Ningrum.ino
198 // ===== TAMPILKAN KE LCD =====
199 // Baris 1: Fuel Level + Bar Grafik
200 lcd.setCursor(0, 0);
201 lcd.print("FUEL: ");
202 lcd.print((int)fuel_percentage);
203 lcd.print("% ");
204 lcd.print("\n");
205 int bars = map(fuel_percentage, 0, 100, 0, 5);
206 for (int i = 0; i < 5; i++) {
207   if (i < bars) lcd.print("#");
208   else lcd.print(" ");
209 }
210 lcd.print("\n");
211
212 // Baris 2: TANKS status dan counter
213 lcd.setCursor(0, 1);
214 lcd.print("TANKS: ");
215 lcd.print(sensor1_status ? "1" : "-");
216 lcd.print(" ");
217 lcd.print(sensor2_status ? "2" : "-");
218 lcd.print(" ");
219 lcd.print(sensor3_status ? "3" : "-");
220 lcd.print(" ");
221 lcd.print(sensor4_status ? "4" : "-");
222 lcd.print("\n");
223 lcd.print(sensors_active);
224 lcd.print("/4");

```

```

226
227 // Baris 3: Pump + nilai raw
228 lcd.setCursor(0, 2);
229 lcd.print("PUMP: ");
230 lcd.print(fuel_pump_status ? "ON " : "OFF");
231 lcd.setCursor(11, 2);
232 lcd.print("RAW: ");
233 lcd.setCursor(15, 2);
234 lcd.print(fuel_raw);
235
236 // Baris 4: Status kondisi
237 lcd.setCursor(0, 3);
238 if (fuel_percentage <= FUEL_THRESHOLD) {
239   lcd.print(" *** FUEL LOW *** ");
240 } else if (sensors_active == 0) {
241   lcd.print(" *** ALL FULL *** ");
242 } else {
243   lcd.print(" SYSTEM NORMAL ");
244 }
245
246 delay(1000); // Delay 1 detik sebelum pembacaan berikutnya
247 }
248
249 /*

```

Gambar 4. 15 Kondisi Tampil