

## RANCANG BANGUN ALAT PEMUPUKAN OTOMATIS BERBASIS MIKROKONTROLER ARDUINO UNO MENGGUNAKAN SENSOR WATERFLOW PADA MEDIA POLYBAG

Sukarni <sup>1)</sup>, Anang Efendi <sup>2)</sup>, Budi Prasetyo <sup>3)</sup>, Robiatul Adawiyah<sup>4)</sup>

<sup>1,2,3,4)</sup> Program Studi Teknik Informatika, STT POMOSDA

e-mail: <sup>1</sup>[carnytanjung@gmail.com](mailto:carnytanjung@gmail.com), <sup>2</sup>[afendystt@gmail.com](mailto:afendystt@gmail.com), <sup>3</sup>[khunaji@gmail.com](mailto:khunaji@gmail.com),  
<sup>4</sup>[adawiyah@stt-pomosda.ac.id](mailto:adawiyah@stt-pomosda.ac.id)

### ABSTRAK

Teknologi memudahkan manusia dalam melakukan berbagai aktivitas. Teknologi dapat merusak orang dengan membiarkan mereka bekerja secara otomatis. Perangkat elektronik saat ini hampir semua menggunakan sensor. Saat itulah teknologi mulai dikembangkan pada berbagai bidang terutama pertanian. Salah satunya adalah pemupukan. Aplikasi yang tidak disemprot dan tidak merata akan menghasilkan tanaman buah atau sayur kurang optimal. Dalam hal ini, biaya mempekerjakan orang sangat tinggi, sehingga diperlukan alat yang dapat meminimalkan biaya tenaga kerja dan penggunaan pupuk sesuai dengan dosis yang dibutuhkan. Metode yang dipakai adalah merancang alat pemupukan berbasis Arduino Atmega2560 dan sensor air. Hasil alat bangunan dapat membantu pertanian di masa depan. Hasil dari alat tersebut sebagai bentuk keberhasilan program yang dapat dijalankan sesuai perintah dan kesesuaian hasil volume. Setelah konfirmasi pemupukan otomatis pada pengujian dan perubahan desain alat pemupukan maka perakitan alat dinyatakan berhasil dengan prosentase error selisih volume 0,29%, dan program yang dijalankan sesuai perintah setting mikrokontroler.

**Kata kunci :** *Mikrokontroler, Arduino, Waterflow Sensor*

### PENDAHULUAN

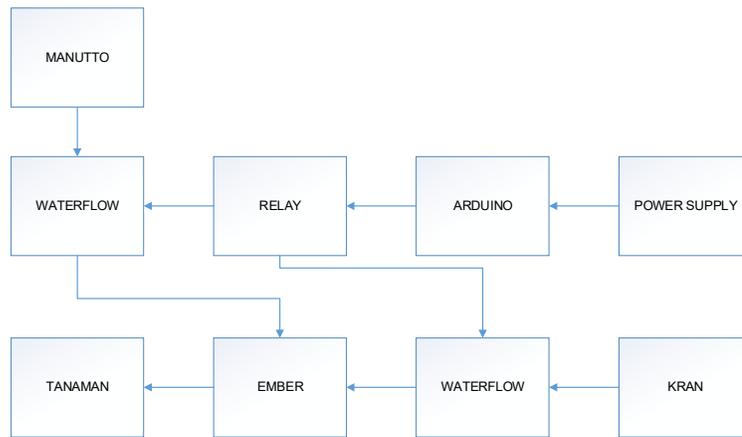
Teknologi memudahkan manusia dalam berbagai aktivitas. Teknologi dapat merusak orang dengan membiarkan mereka bekerja secara otomatis. Contoh produk teknologi adalah mikrokontroler seperti Arduino Uno. Saat itulah teknologi ini mulai dikembangkan berbagai bidang terutama pertanian. Salah satunya adalah pemupukan. Pupuk secara kasar dibagi menjadi pupuk kimia sintetis dan pupuk kimia alami. Pupuk sintetis adalah pupuk yang mengandung senyawa sintetis. Kebanyakan pupuk sintetis adalah pupuk buatan. Di sisi lain, pupuk alami adalah pupuk yang mengandung senyawa alami. Berdasarkan uraian tersebut, peneliti memperkenalkan inovasi dalam pengembangan teknologi inseminasi otomatis dengan judul makalah "Perancangan Alat Pemupukan Otomatis Berbasis Mikrokontroler Arduino Uno Menggunakan Sensor Aliran Air Pada Media Polybag".

Berdasarkan identifikasi masalah, kita dapat merumuskan rumusan masalah. Yaitu bagaimana merancang alat pemupukan otomatis berbasis mikrokontroler Arduino Uno Atmega2560 menggunakan sensor aliran air YF-S401 pada media polybag. Tujuan dari penelitian ini adalah untuk mengembangkan sistem otomatisasi berbasis mikrokontroler Arduino Uno Atmega 2560 dengan sensor aliran air YF-S401 pada media polybag untuk meminimalkan biaya tenaga kerja dan mengefisienkan penggunaan pupuk dan air tanaman.

### PERANCANGAN ALAT

Langkah pertama membuat suatu blok diagram sebagai acuan dari fungsi tertentu dan saling terkait sehingga membentuk suatu sistem. Proses berawal dari sumber listrik yang dialirkan ke arduino, kemudian arduino memerintahkan relay untuk on agar sensor air terbuka agar air dan pupuk dapat mengalir ke tempat pencampuran/ember. Kemudian setelah selesai melakukan pengisian *relay* dan sensor air akan off agar air serta pupuk tidak terus mengalir, lalu setelah

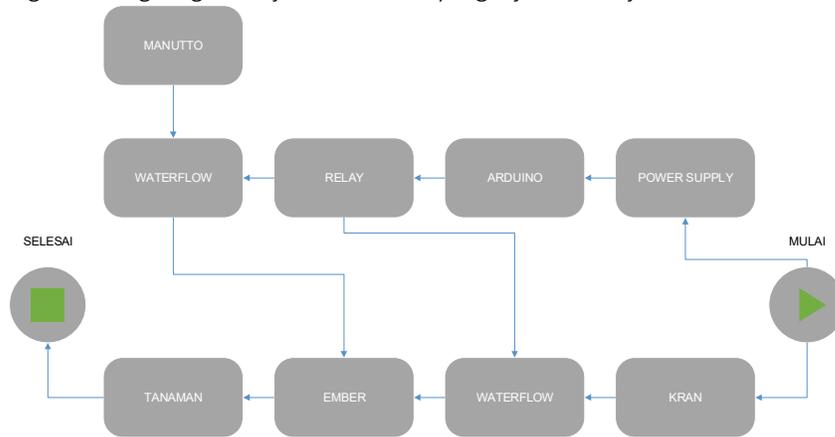
proses pencampuran akan disedot keluar dengan menggunakan pompa air yang akan langsung menuju ke tanaman yang dijadikan objek.



gambar 3.1 Diagram blok perancangan  
(Sumber : data primer diolah 2021)

**Work Flow Diagram**

Berfungsi sistem kerja dari sebuah proses alat. Dimulai dari awal alat tersebut berjalan hingga alat tersebut berhenti. Proses diawali dari *power supply* memberikan tenaga listrik ke arduino, kemudian arduino memerintahkan *relay* untuk on agar sensor *waterflow* terbuka agar air dan pupuk dapat mengalir ke tempat pencampuran/ember. Kemudian setelah selesai melakukan pengisian relay dan sensor *waterflow* akan *off* agar air dan pupuk tidak terus mengalir, lalu setelah proses pencampuran akan disedot keluar dari tempat pencampuran menggunakan pompa air yang akan langsung menuju ke tanaman yang dijadikan objek.



gambar 3.3 workflow diagram  
(Sumber : data primer diolah 2021)

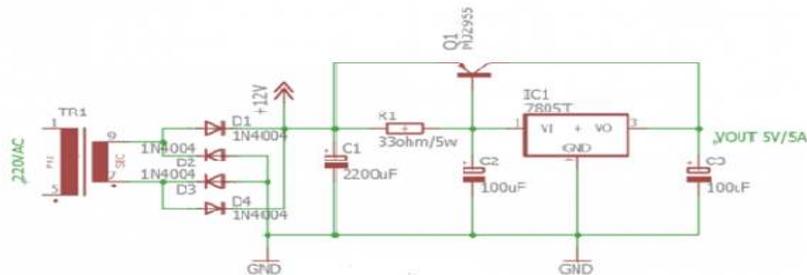
**Alat-alat yang dibutuhkan**

Alat-alat yang dibutuhkan untuk membuat sebuah alat pemupukan otomatis adalah sebagai berikut :

1. Solder	11. pipa ¾
2. Laptop	12. pipa akuarium
3. Gunting	13. pompa air

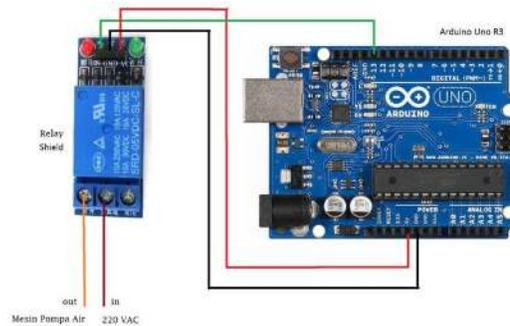
4. Obeng	14. kabel jumper
5. Arduino	
6. power supply	
7. waterflow sensor	
8. LCD	
9. relay	
10. ember	

**Rangkaian alat**



Gambar 3.4 Rangkaian Catu Daya

**Rangkaian Relay**



Gambar 3.5 Rangkaian Relay  
(Sumber : indomaker.com)

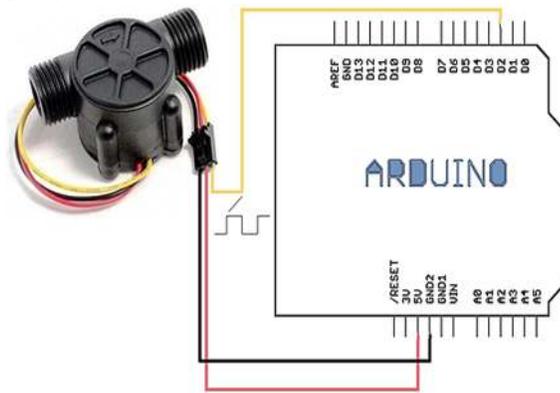
Tabel 3.1 Konfigurasi Output Relay Pompa Air ke Arduino.

Alat	Keterangan	Hubungan Dengan Mikrokontroler
Pompa Air	Vcc	Pin Vcc
	Data	Pin 5
	Gnd	Pin Gnd

Tabel 3.1 Konfigurasi Output Relay Pompa Air ke Arduino.

Alat	Keterangan	Hubungan Dengan Mikrokontroler
Pompa Air	Vcc	Pin Vcc
	Data	Pin 5
	Gnd	Pin Gnd

### Rangkaian Sensor *Water Flow*



Gambar 3.6 Rangkaian Sensor *Water Flow* Ke Mikrokontroler  
(Sumber : Adisanjaya.com)

Tabel 3.2 Konfigurasi Kabel *Input* Sensor *Water Flow* ke Arduino.

Alat	Keterangan	Hubungan Dengan Mikrokontroler
Sensor <i>Water Flow</i>	Vcc	Pin vcc
	Data	Pin 2
	Gnd	Pin gnd

### PENGUJIAN ALAT

#### Pengujian Arduino Uno

Pengujian dilakukan dengan cara meng-*upload* Program *Example* yang tersedia pada *Library Arduino Uno*, Program *Example* yang dicoba akan menghidupkan dan mematikan LED yang terdapat pada *Arduino Uno* secara teratur sesuai dengan waktu *delay* yang di berikan.

Tabel 4.1 Hasil Tegangan arus listrik Pada *Arduino*

Hasil pengukuran tegangan pada pin 3	Hasil pengukuran tegangan pada pin 3 setelah 3 detik
5,1 v	0 v

Sumber : Data diolah Peneliti

Analisa :

Pengujian mikrokontroler ini dilakukan untuk pengecekan apakah mikrokontroler dapat menjalankan program yang sudah di *setting* pada Aplikasi *Arduino Uno* serta apakah tegangan keluaran dari mikrokontroler ada atau tidak. Program yang digunakan untuk pengujian keluaran *Arduino Uno* yaitu program *digital Write (led, HIGH); delay (3000); digital Write (led, LOW); delay (3000)*; program akan mengeluarkan tegangan pada pin *out* led ketika waktu 3 detik kemudian program akan menghentikan tegangan keluaran selama 3 detik juga. Program akan terus berulang sehingga tegangan yang dihasilkan pun akan terus bergantian sebesar 5,1 V dan 0V. Pemrograman *digitalWrite* pada *Arduino Uno* berfungsi untuk memberikan tegangan pada *output* pin *Arduino Uno* ketika di beri perintah *HIGH* dan menarik tegangan pada pin ketika diberi perintah *LOW*. Dari percobaan yang dilakukan, perintah *digitalWrite HIGH* dan *LOW* pada pin *output* tersebutlah yang menghasilkan data 5,1 V dan 0 V pada tabel hasil percobaan di atas. Berdasarkan pengujian di atas kita dapat melakukan pengontrolan cara kerja alat melalui *Arduino Uno* ini, baik pengontrolan pengisian dengan cara memberikan logika *HIGH* pada *input* pin *relay* pompa air serta memberikan logika *LOW* pada *input* pin *relay* untuk mematikannya. Disini pengimplementasian sistem *mikrokontroler* dapat digunakan sebagai pengontrol pengisian.

### Pengujian dan Analisa Relay

Pengujian *relay* dilakukan agar mengetahui masukan tegangan yang disalurkan oleh *Arduino Uno* dengan cara memberikan nilai masukan tegangan dari *Arduino Uno*.

Tabel 4.2 Hasil Ujicoba Relay

Pin 5 <i>Arduino Uno</i>	Tegangan pada <i>relay</i>	Kondisi
High	5,1 V	On
Low	0 V	Off

(Sumber: Data diolah peneliti, 2021)

Analisa :

Pada hasil pengukuran yang telah dilakukan dapat diketahui, rangkaian *relay* akan aktif jika mendapat tegangan dari *Arduino Uno* begitu juga sebaliknya cara kerja inilah yang berfungsi untuk menyalakan pompa. Sesuai pengukuran hasil tegangan pada *Arduino Uno* sebesar 5,1 V. Hasil ini mendekati dari karakteristik *output* pada *Arduino Uno* yaitu sebesar 5 V.

### Pengujian Dan Analisa Sensor Water Flow

Pengujian Sensor *Water Flow* bertujuan untuk mendapatkan nilai debit aliran dan *pulse frequency* yang terukur pada program sesuai dengan besaran nilai debit dan *pulse frequency* yang diinginkan. Pengujian dilakukan sebanyak 3 Kali dengan masing-masing pengujian melakukan lima percobaan atau kelipatan lima proses pengisian dengan menyesuaikan nilai yang terukur pada Sensor *Water Flow*.

Tabel 4.3 Hasil Pengujian Sensor *Water Flow*

Pengujian	Debit L/minute
Ke-1	0,251
Ke-2	0,25
Ke-3	0,252
Rata-Rata	0,251

(Sumber: Data diolah peneliti, 2021)

Pengujian aliran pemupukan digunakan untuk mendapatkan data pengaruh aliran pupuk dan air terhadap debit pada sistem. Dari hasil pengujian debit aliran yang ditunjukkan tabel 4.3, rata-rata debit aliran adalah 0,251 liter/menit. Selanjutnya adalah perhitungan nilai frekuensi pulsa berdasarkan persamaan 0,25. Jika aliran pemupukan debit alirannya adalah 0,251 liter/menit maka *pulse frequency* adalah:  $Pulse\ frequency = 7,5 \cdot Q = 7,5 \cdot 0,251 = 1,88\ Hz$

### Hasil Pengujian Alat

Dengan diketahui nilai debit aliran dan frekuensi pulsa pada sistem dapat disimpulkan bahwa aliran hanya berpengaruh terhadap lamanya waktu dan *pulse frequency* untuk mencapai nilai volume.

Tabel 4.4 Pengujian alat pemupukan

No	Input		Jumlah Input	Output (ml)	Waktu	Selisih	Error
	Manutta	Air					
1	1,5	250	251,5	251,5	20	0	0,3
2	1,5	250	251,5	250,9	20	0,6	0,2
3	1,5	250	251,5	251,5	20	0	0,3
4	1,5	250	251,5	251,5	20	0	0,3
5	1,5	250	251,5	251	20	0,5	0,1
	Rata-rata			251,2			0,2

(Sumber: Data diolah peneliti, 2021)

Berdasarkan hasil pengujian pada tabel 4.4, pengujian di lakukan sebanyak 5 kali percobaan proses pemupukan pada volume 251,5 ml. Selama percobaan di dapatkan rata-rata hasil pengujian sebesar 251,2 ml dengan rata-rata waktu yang di butuhkan selama 20 detik dengan tingkat *error* sebesar 0,2 %.

Cara menghitung tingkat error :

$$\text{Error (\%)} = \frac{\text{Selisih In - Out} \times 100}{\text{Input}}$$

#### Analisa Keberhasilan Alat

Keberhasilan alat di lihat dari keberhasilan program yang dapat berjalan sesuai perintah dan kesesuaian hasil *volume*. Setelah melakukan pengujian pemupukan otomatis dan perubahan desain alat pemupukan pada pengujian, perakitan alat dinyatakan berhasil dengan persentase *error* pada selisih *volume* sebesar 0,29%, serta program yang dijalankan sudah berjalan sesuai dengan perintah yang ter-*setting* pada mikrokontroler.

a. Foto Percobaan Dengan *Error* Tertinggi, Kurang 10 ml



Gambar 4.1 Percobaan *Error* Tertinggi

b. Foto Percobaan Dengan *Error* Terendah, Tepat 250 ml



Gambar 4.2 Percobaan *Error* Terendah

#### PENUTUP

##### Kesimpulan

Berdasarkan hasil pengujian sistem pemupukan otomatis menggunakan sensor aliran air dan mikrokontroler Arduino Atmega2560. Dari hasil pengujian, pemupukan otomatis dilakukan dalam waktu 60 detik setelah satu injeksi ke dalam kantong plastik, dan dua tabung pengisi dirancang secara seri yang masing-masing diisi dengan air dan diisi dengan pupuk, dan mengalir ke wadah yang disebut mangkuk pencampur. .meningkat. Campuran air/air kemudian

mengeluarkan kotoran dari titik pencampuran langsung ke dalam kantong plastik. Setelah 15 kali run untuk mendapatkan produksi 250ml/polybag, alat ini dinyatakan berhasil dengan tingkat kesalahan 0,29%. Dan program yang dieksekusi sudah dijalankan sesuai dengan perintah yang dimasukkan ke mikrokontroler.

#### LAMPIRAN

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// SETUP INTERRUPT 0, 1, 5, 4
byte sensorInt1 = 0;
byte sensorInt2 = 1;
// PIN ARDUINO
byte flow1 = 2;
byte flow2 = 3;

float konst = 4.5;
unsigned long oldTime;

volatile byte count1;
float flow_mlt1;
float total_volume1;
volatile byte count2;
float flow_mlt2;
float total_volume2;

byte valve_air = 31;
byte valve_Pupuk = 29;
byte relay3 = 27;
byte relay4 = 25;

const int buttonPin = 13;
int buttonState = 0;
int mode = 0;

void setup() {
  lcd.init(); // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  // lcd.begin(20, 4);
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);

  lcd.print("TEST LCD i2C");
  lcd.setCursor(0, 1);

  pinMode(flow1, INPUT);
  pinMode(flow2, INPUT);
  digitalWrite(flow1, HIGH);
  digitalWrite(flow2, HIGH);
```

```
// Set relay
pinMode(valve_air, OUTPUT);
pinMode(valve_Pupuk, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);
digitalWrite(valve_air, LOW);
digitalWrite(valve_Pupuk, LOW);
digitalWrite(relay3, LOW);
digitalWrite(relay4, LOW);

count1 = 0.00;
flow_mlt1 = 0.00;
total_volume1 = 0.00;
oldTime = 0.00;
count2 = 0.00;
flow_mlt2 = 0.00;
total_volume2 = 0.00;
attachInterrupt(sensorInt1, countPulse1, FALLING);
attachInterrupt(sensorInt2, countPulse2, FALLING);
}

void loop() {
  lcd.setCursor(0, 0);
  buttonState = digitalRead(buttonPin);
  flow();

  if (buttonState == HIGH) {
    mode = mode + 1; delay(300);
  }
  if (mode == 3) {
    mode = 0;
  }
  if (mode == 0) {
    lcd.clear();
    lcd.print("Tekan tombol");
    lcd.setCursor(0, 1);
    Serial.println("tekan tombol");
    delay(300);
  }
  else if (mode == 1) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Tanaman Sawi");
    Serial.println("Tanaman Sawi");
    lcd.setCursor(2, 2);
    lcd.print("Air = ");
    lcd.print(total_volume1);
    lcd.print(" mL ");
    lcd.setCursor(2, 3);
```

```
lcd.print("Pupuk = ");
lcd.print(total_volume2);
lcd.print(" mL ");
delay(300);

if (total_volume1 >= 12)
{ digitalWrite(valve_Pupuk, HIGH);
}
if (total_volume2 >= 2000) {
digitalWrite(valve_air, HIGH);
}
}
else if (mode == 2) {
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Tanaman Selada");
Serial.println("Tanaman Selada");
lcd.setCursor(2, 2);
lcd.print("Air = ");
lcd.print(total_volume1);
lcd.print(" mL ");
lcd.setCursor(2, 3);
lcd.print("Pupuk = ");
lcd.print(total_volume2);
lcd.print(" mL ");
delay(300);

if (total_volume1 >= 12) {
digitalWrite(valve_Pupuk, HIGH);}
if (total_volume2 >= 2000) {
digitalWrite(valve_air, HIGH);
}
}
}

void flow() {
if ((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt1);
detachInterrupt(sensorInt2);

flow_mlt1 = (((1000.0 / (millis() - oldTime)) * count1) / konst / 60) * 1000; flow_mlt2 = (((1000.0 / (millis() - oldTime)) * count2) / konst / 60) * 1000;

oldTime = millis();
total_volume1 += flow_mlt1;
total_volume2 += flow_mlt2;

Serial.print(total_volume1);
Serial.print(" mL ");
Serial.print(total_volume2);
```

```
Serial.println(" mL ");

count1 = 0;
count2 = 0;

attachInterrupt(sensorInt1, countPulse1, FALLING);
attachInterrupt(sensorInt2, countPulse2, FALLING);
}
}

void countPulse1() {
count1++;
}
void countPulse2() { count2++;
}
```

#### DAFTAR PUSTAKA

- Arifin, J., Zulita, L. N., & Hermawansyah, H. (2016). Perancangan Murottal Otomatis Menggunakan Mikrokontroler Arduino Mega 2560. *Jurnal Media Infotama*, 12(1).
- PUIMERA, Y. I., & Danang, D. (2018). Rancang Bangun Alat Penyortiran Barang Otomatis Berbasis Arduino Pada Pt Wahana Prestasi Logistik Semarang. *Elkom: Jurnal Elektronika dan Komputer*, 11(1), 38-44.
- Sokop, S. J., Mamahit, D. J., & Sompie, S. R. (2016). Trainer perifer al antarmuka berbasis mikrokontroler arduino uno. *Jurnal Teknik Elektro Dan Komputer*, 5(3), 13-23.
- Suratman, Y. Y. A. (2018). Analisis Pendapatan Usahatani Sawi (*Brassica juncea* L.) di Kelurahan Landasan Ulin Utara Kecamatan Liang Anggang Kota Banjarbaru. *ZIRAA'AH MAJALAH ILMIAH PERTANIAN*, 43(2), 133-140.
- Susanti, E., & Hasbi, M. (2019). Desain Sistem Gerak Robot Quadruped Berbasis Arduino Menggunakan Bluetooth HC-05. *SIGMA TEKNIKA*, 2(1), 20-31.