

## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

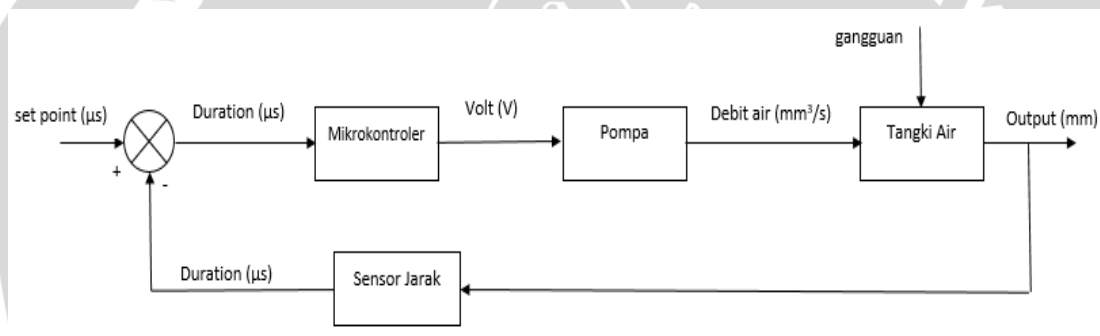
#### 4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan ini terdiri atas:

- Perancangan perangkat keras terdiri atas perancangan Arduino *Uno*, rangkaian catu daya, rangkaian *driver* motor beserta pompa air.
- Perancangan perangkat lunak yaitu perancangan program menggunakan *software* Arduino 1.0.5.

#### 4.2 Perancangan Diagram Blok Sistem

Diagram balok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



**Gambar 4.1** Diagram Balok Sistem (Perancangan)

Keterangan dari blok diagram diatas adalah sebagai berikut:

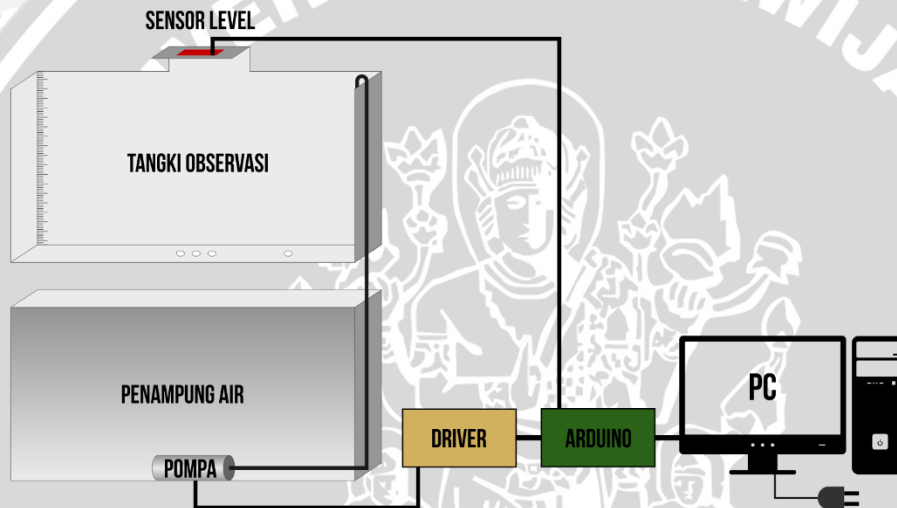
1. *Setpoint* sistem berupa durasi waktu yang dikonversi menjadi jarak ketinggian oleh mikrokontroler.
2. Kontroler yang digunakan adalah kontroler Proporsional Integral Differensial (PID) menggunakan perangkat keras ArduinoUno.
3. Aktuator yang digunakan adalah pompa DC 12 V.
4. Sensor Ultrasonic HC-SR04 sebagai pembaca jarak ketinggian air digunakan untuk *feedback* sistem.
5. Gangguan pada plant yaitu beban lebih selama 15 detik.

#### 4.3 Spesifikasi Alat

Spesifikasi sistem pengendalian level air menggunakan Arduino Uno adalah sebagai berikut :

- Dimensi tangki observasi sebesar 30cm x 20cm x 5cm.

- Tangki dilengkapi dengan 1 katub keluaran air terbuka dan 3 katub keluaran air (buka tutup) dibawah tangki yang digunakan untuk variasi nilai katub keluaran air / beban dan juga gangguan.
- Diameter tiap-tiap katub keluaran air atau beban yaitu 5 mm.
- Air yang digunakan merupakan air dari PDAM.
- Sensor yang digunakan adalah sensor jarak HC-SR04 dengan rentang jarak 30-2000 mm.
- Aktuator menggunakan dua pompa air dengan motor DC yang mendapat sinyal masukan dari *output PWM driver EMS H-Bridge 5A*.
- Skema perancangan sistem dapat dilihat dalam Gambar 4.2.



**Gambar 4.2** Skema Keseluruhan Sistem (Perancangan)

#### 4.4 Prinsip Kerja Sistem

Cara kerja sistem adalah sebagai berikut :

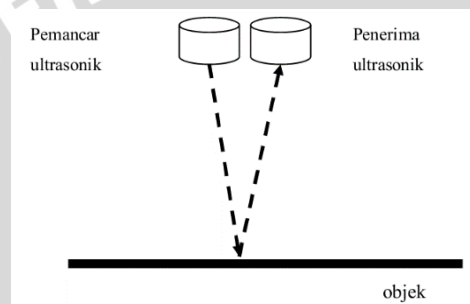
- Catu daya 12 Volt digunakan sebagai catu motor DC pada pompa air.
- Port komunikasi antara Arduino Uno dan PC menggunakan perantara kabel USB dengan kecepatan transfer data sebesar 9600 bps.
- Sinyal kontrol dari Arduino Uno masuk ke *driver EMS H-Bridge 5A*. Driver berfungsi menguatkan sinyal yang dihasilkan Arduino Uno dari 0-5 Volt menjadi 0-12 Volt.
- Motor DC pada pompa air yang terhubung pada tangki observasi yang akan mengisi sampai mencapai set point.

#### 4.5 Perancangan Perangkat Keras (*Hardware*)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan rangkaian catu daya, rangkaian *driver* motor beserta pompa air. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

##### 4.5.1 Sensor ultrasonik HC-SR04 (Perancangan)

Pada perancangan sensor jarak HC-SR04 ini, sensor diletakkan diatas tangki air yang mana jika nilai ketinggian air makin bertambah maka nilai pembacaan sensor semakin kecil. Dapat dikatakan jarak ketinggian air berbanding terbalik dengan nilai durasi pembacaan sensor. Untuk lebih jelasnya dapat digambarkan pada Gambar 4.3.



**Gambar 4.3** Sistem kerja sensor jarak ultrasonik (perancangan)

Rumus pada persamaan (2.2) dapat disederhanakan agar mudah saat dimasukkan di mikrokontroler untuk mendapatkan nilai jarak dalam milimeter dan didapatkan rumus dibawah ini.

$$S = \frac{t_{IN}}{58,2} \times 10$$

Dengan :

$S$  = Jarak sensor ke objek yang dideteksi.

$t_{IN}$  = waktu pemancaran dan penerimaan pantulan.

Untuk mendapatkan nilai ketinggian pada tangki dilakukan pengukuran pada tangki keadaan kosong didapatkan nilai  $S = 226$ , kemudian dilakukan konversi menjadi nilai ketinggian untuk dimasukkan di mikrokontroler. Rumus untuk mendapatkan nilai ketinggian yaitu.

$$H = 226 - S$$

Dengan :

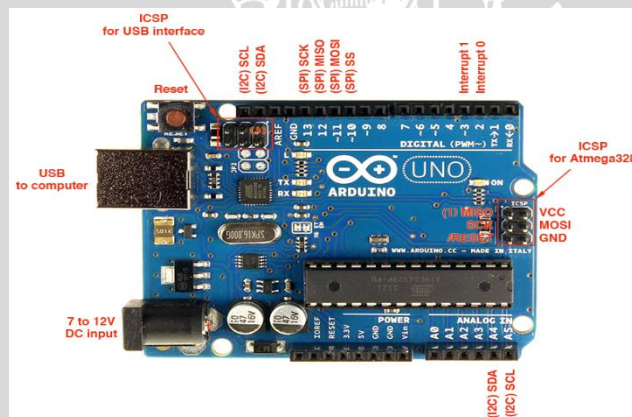


$S$  = Jarak sensor ke objek yang dideteksi (mm)

$H$  = Ketinggian air/objek yang dideteksi (mm)

#### 4.5.2 Rangkaian Arduino UNO dan Konfigurasi I/O Arduino UNO

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin *digital input* dan *digital output* (6 di antaranya dapat digunakan sebagai *output PWM*), 6 *input analog*, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya sebesar 5 volt dengan sebuah *adaptor AC ke DC* atau menggunakan baterai untuk. Pada alat ini digunakan Arduino Uno sebagai pusat pengolah utama. Konfigurasi I/O dari Arduino UNO ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Arduino UNO(electroschematics.com)

Tabel fungsi masing masing pin Arduino UNO dapat dilihat dalam Tabel 4.1

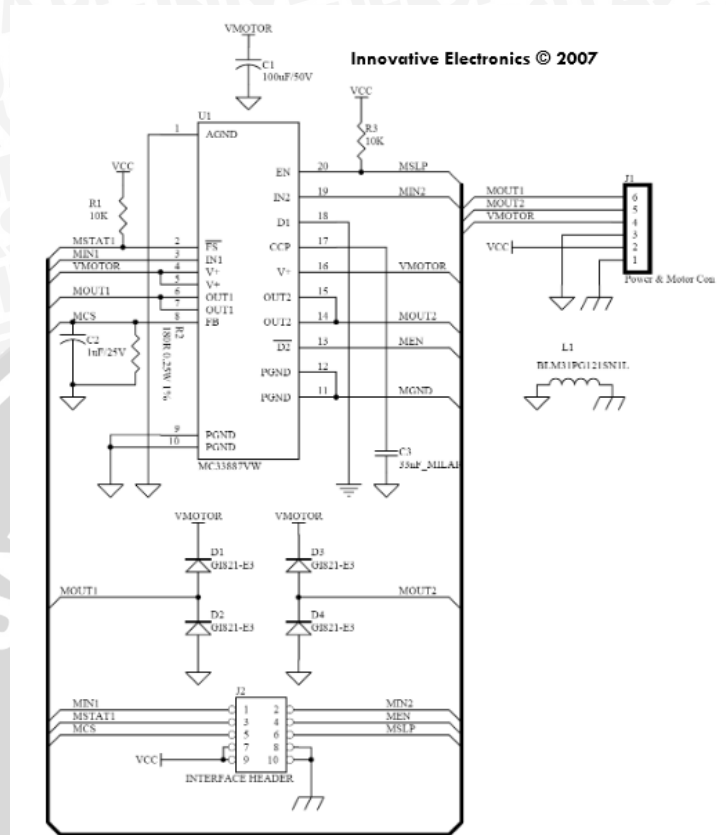
Tabel 4.1 Fungsi Pin Arduino Uno

No.	Pin	Fungsi
1	D5	Digunakan sebagai input trigger sensor ultrasok
2	D6	Digunakan sebagai input echo sensor ultrasonic
3	D10	Digunakan sebagai antar muka <i>driver</i> motor 2

#### 4.5.3 Driver motor EMS H-Bridge 5 A

Modul pengendali motor DC digunakan untuk mengendalikan putaran motor DC yang menjadi penggerak pompa. Rangkaian ini dihubungkan dengan Arduino Uno. *Driver* pengendali pada perancangan ini menggunakan *driver*

EMS H-Bridge 5 A yaitu sebuah perangkat keras berupa rangkaian yang berfungsi untuk menggerakkan motor DC. Rangkaian skematik *driver* motor EMS H-Bridge 5 A ditunjukkan pada Gambar 4.5.



**Gambar 4.5** Skematik Rangkaian Driver EMS H-Bridge 5A (datasheet EMS H-Bridge 5A)

*Interface Header* pada driver EMS H-Bridge 5 A berfungsi sebagai input untuk antarmuka dengan input-output digital serta output analog dari modul H-Bridge. Berikut deskripsi dari masing-masing pin ditunjukkan pada tabel 4.1.

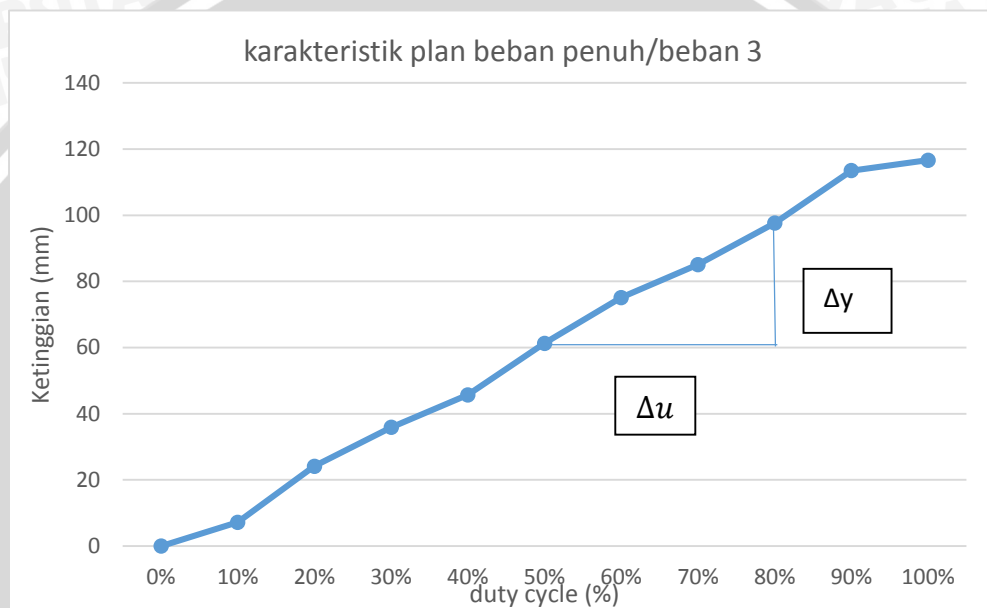
Tabel 4.2 Fungsi Pin EMS H-Bridge 5 A

No. Pin	Fungsi
1	Pin <i>input</i> untuk menentukan <i>output</i> MOUT1
2	Pin <i>input</i> untuk menentukan <i>output</i> MOUT2
3	<p>Pin <i>enable</i> untuk <i>output</i> MOUT1</p> <p>Diberi logika <i>high</i> untuk mengaktifkan <i>half</i> H-Bridge</p> <p>1, diberi logika <i>low</i> secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 1</p> <p>Jika terjadi kondisi <i>fault</i> (<i>thermal shutdown, undervoltage, overvoltage, dsb.</i>), maka pin ini akan ditarik <i>low</i> secara internal oleh modul H-Bridge</p> <p>untuk melaporkan adanya kondisi <i>fault</i></p>
4	<p>Pin <i>enable</i> untuk <i>output</i> MOUT2</p> <p>Diberi logika <i>high</i> untuk mengaktifkan <i>half</i> H-Bridge</p> <p>2, diberi logika <i>low</i> secara eksternal untuk menonaktifkan <i>half</i> H-Bridge 2</p> <p>Jika terjadi kondisi <i>fault</i> (<i>thermal shutdown, undervoltage, overvoltage, dsb.</i>), maka pin ini akan ditarik <i>low</i> secara internal oleh modul H-Bridge</p> <p>untuk melaporkan adanya kondisi <i>fault</i></p>
5	<i>Output</i> tegangan analog yang berbanding lurus dengan arus beban ( <i>range output</i> 0 – 5 Volt)
6	Pin <i>input</i> untuk mengatur kerja modul H-Bridge secara PWM
7,9	Terhubung ke catu daya untuk <i>input</i> (5 Volt)
8,10	Titik referensi untuk catu daya <i>input</i>



#### 4.5.4 Perancangan Algoritma Kontroler

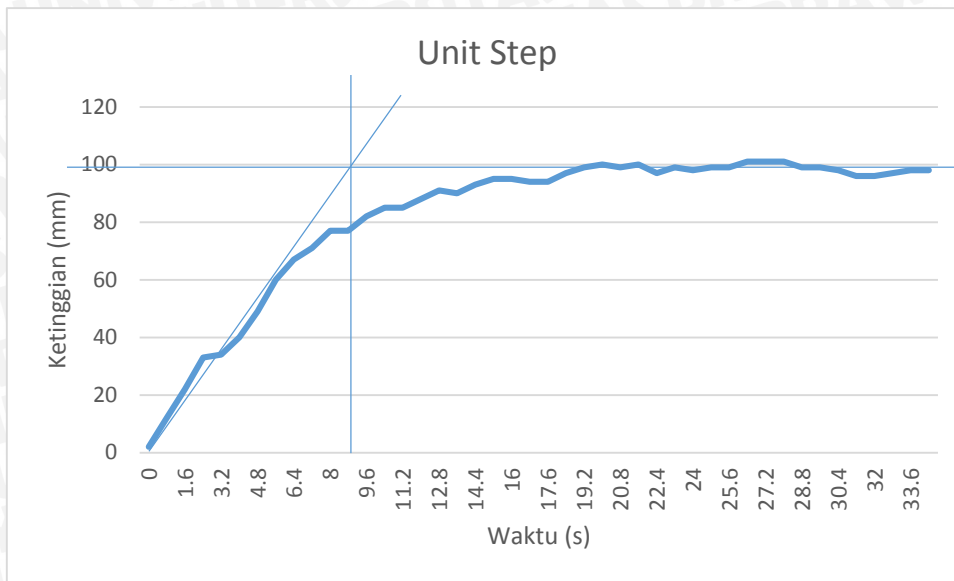
Untuk menemukan fungsi alih plant untuk mendapatkan karakteristik statis gain keadaan mantap dengan cara plant di rancang secara *open loop* dan diberikan unit step masukan *duty cycle* 0 – 100 % dan di lihat bagaimana respon keadaan mantapnya. Untuk menemukan parameter kontroler PID digunakan metode ziegler-nichols 1, dimana plant diberi tegangan masukan ke plant dan dilihat bagaimana hasilnya.



**Gambar 4.6** Grafik Karakteristik statis gain keadaan mantap

$$k = \frac{\Delta y}{\Delta u} = \frac{97,6 - 61,3}{80 - 50} = 1,21$$

Untuk menemukan parameter kontroler PID digunakan metode ziegler-nichols 1, dimana plant diberi tegangan masukan ke plant dan akan mendapatkan nilai L dan T.



**Gambar 4.7** Grafik Output Plant dengan Beban Maksimum

$$L = 0,8 \quad T = 8,8$$

Sehingga di dapatkan fungsi alih sebagai berikut :

$$G(s) = \frac{ke^{-Ls}}{Ts + 1} = \frac{1,21e^{-0,4}}{8,8s + 1}$$

Dan diperoleh nilai  $K_p$ ,  $K_i$ ,  $K_d$  :

$$K_p = 1.2 \times \frac{T}{L} = 1.2 \times \frac{8,8}{0,8} = 11$$

$$K_i = \frac{K_p}{T_i} = \frac{42}{2 \times L} = \frac{11}{2 \times 0,8} = 6,87$$

$$K_d = K_p \times T_d = 11 \times (0.5 \times 0.8) = 4,4$$

Dari perolehan nilai  $K_p$ ,  $K_i$ ,  $K_d$  maka di dapatkan persamaan transformasi laplace kontroler PID sebagai berikut :

$$M(s) = \left( K_p + \frac{K_i}{s} + K_d s \right) E(s) \dots \dots \dots (1)$$

Persamaan diatas belum bisa dimasukkan kedalam mikrokontroler maka dari itu persamaan kontinyu diatas harus diubah kedalam bentuk diskrit melalui Transformasi Z. Dalam Transformasi Z dibutuhkan waktu sampling ( $T_s$ ). Dengan menggunakan metode *Backward Difference*, ganti operator  $s$  dalam persamaan analog bentuk  $s$  dengan persamaan.



$$S = \frac{1-z^{-1}}{Ts} \dots\dots\dots(2)$$

Persamaan (2) disubstitusikan ke dalam persamaan (1). Perhitungan dilakukan terpisah antara Kp, Ki dan Kd dan diperoleh.

- Kp

$$P \rightarrow Mp(z) = Kp E(z)$$

- Ki

$$I \rightarrow Mi(z) = \frac{Ki}{Ts(1-z^{-1})} E(z)$$

$$Mi(z) = \frac{Ki \times Ts}{(1-z^{-1})} E(z)$$

$$Mi(z) - Mi(z) z^{-1} = Ki \times Ts E(z)$$

$$Mi(z) = Mi(z) z^{-1} + Ki \times Ts E(z)$$

- Kd

$$D \rightarrow Md(z) = \left( Kd \left( \frac{1-z^{-1}}{Ts} \right) \right) E(z)$$

$$Md(z) = \frac{Kd}{Ts} (E(z) - E(z) Z^{-1})$$

Sehingga;

$$M(z) = \left( Kp + \frac{Ki}{s} + Kds \right) E(z)$$

$$M(z) = Kp E(z) + (Mi(z) z^{-1} + Ki \times Ts E(z)) + \left( \frac{Kd}{Ts} (E(z) - E(z) Z^{-1}) \right)$$

Dari persamaan diatas diubah kedalam persamaan beda sehingga didapatkan persamaan dibawah ini;

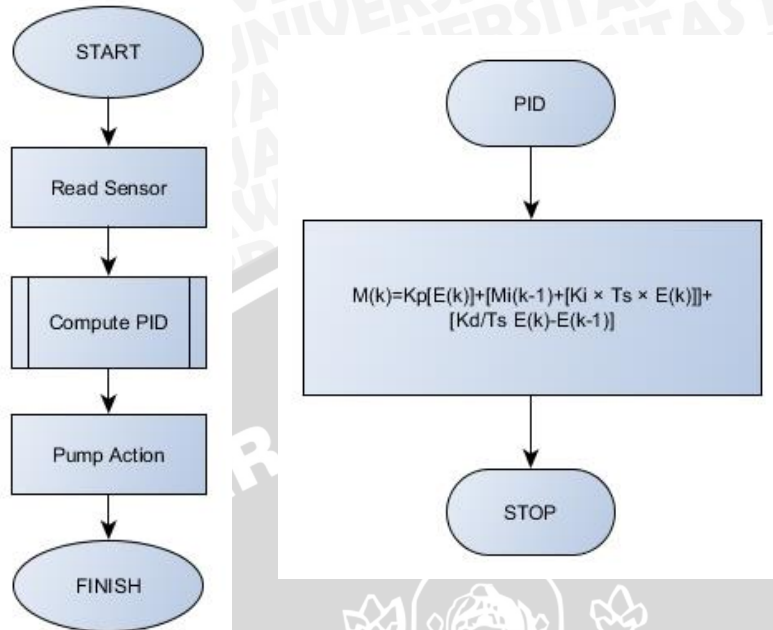
$$M(k) = Kp E(k) + (Mi(k-1) + Ki \times Ts E(k)) + \left( \frac{Kd}{Ts} (E(k) - E(k-1)) \right)$$

Dimana (k-1) adalah kondisi sebelumnya. Persamaan diatas lalu dimasukkan kedalam program pada mikrokontroler.

#### 4.6 Perancangan Perangkat Lunak

#### 4.6.1 Flowchart Sistem Keseluruhan

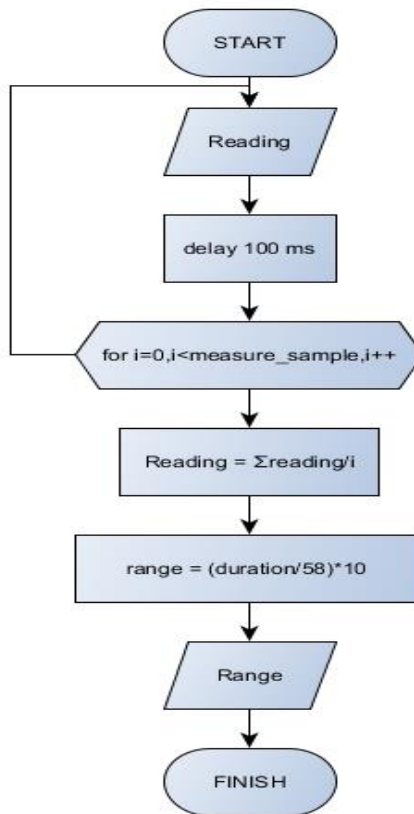
Flowchart keseluruhan sistem ditunjukkan pada Gambar 4.8.



**Gambar 4.8** Flowchart Keseluruhan Sistem (Perancangan)

#### 4.6.2 Flowchart Pembacaan Sensor jarak

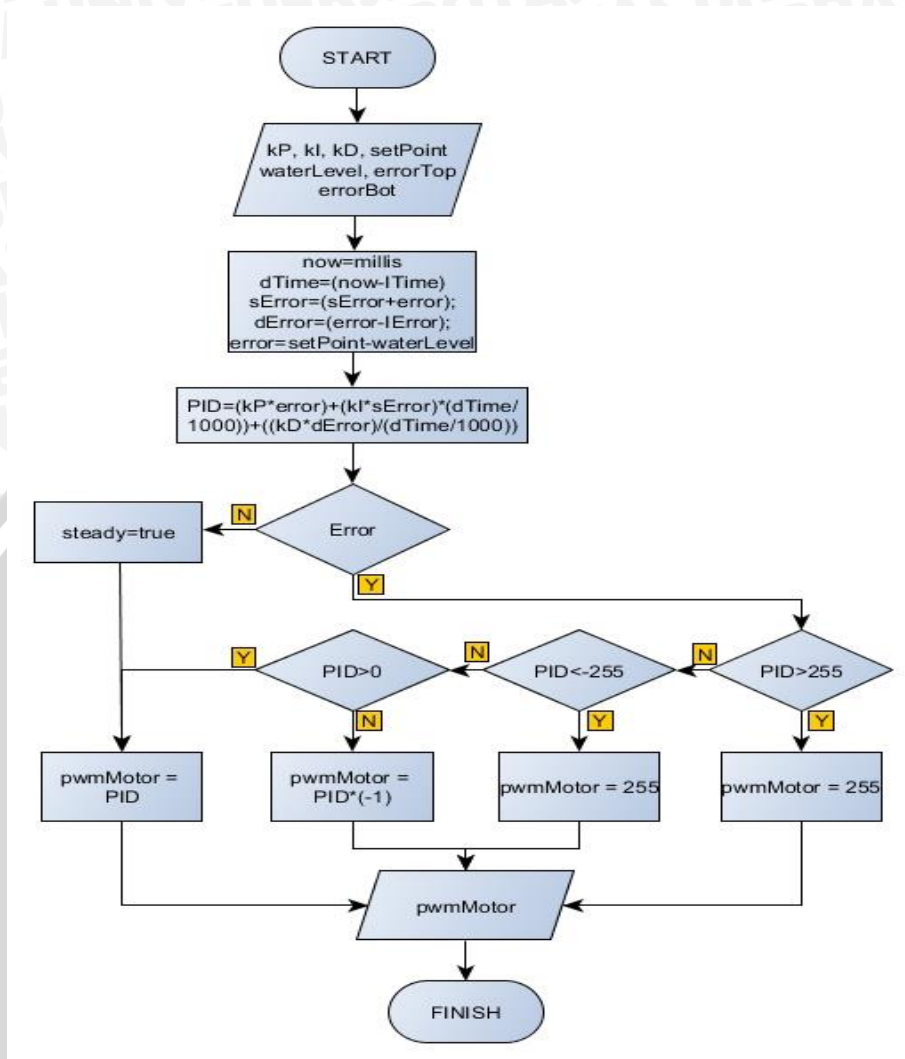
Flowchart pembacaan sensor jarak ditunjukkan pada Gambar 4.9.



**Gambar 4.9** Flowchart Pembacaan Sensor jarak (Perancangan)

#### 4.6.4 Flowchart Perhitungan Parameter PID

Flowchart perhitungan parameter PID ditunjukkan pada Gambar 4.10.



Gambar 4.10 Flowchart Perhitungan Parameter PID (Perancangan)



