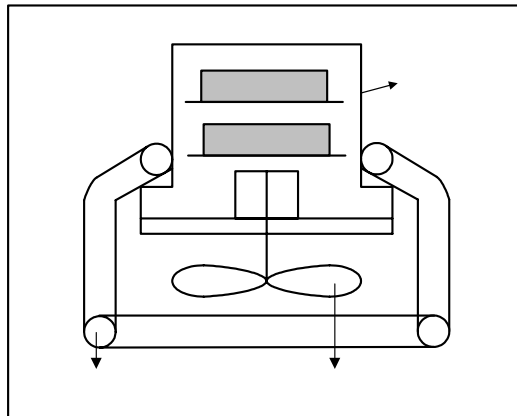


3. PERENCANAAN

Gerakan yang dilakukan oleh robot penyelam adalah gerakan ke atas dan ke bawah. Gerakan ke atas disebabkan oleh adanya putaran dari *propeller* (motor aktif) sedangkan untuk gerakan ke bawah disebabkan oleh berat dari robot tersebut. Oleh karena itu *driver* yang dipakai untuk menggerakkan motor DC cukup 1 arah saja (tidak perlu menggunakan H-bridge).

Untuk badan robot, digunakan toples berbentuk tabung berdiameter 12 cm dengan tinggi 17 cm. Semakin besar volume robot menyebabkan gaya apung yang dialami semakin besar pula. Oleh sebab itu dibutuhkan pemberat agar robot dapat tenggelam.



Gambar 3.1. Desain Robot Penyelam

Gambar 3.1. merupakan desain dari robot penyelam. Dari gambar tersebut dapat dilihat bahwa letak dari *propeller* berada di tengah – tengah penampang robot. Dengan demikian peletakan komponen yang lain, seperti baterai dan rangkaian mikrokontroller serta *driver* motor, tidak boleh sembarangan agar tidak membutuhkan tempat yang besar.

Bagian yang paling rawan dari robot ini adalah pada poros *propeller*. Oleh karena, pada bagian ini terhubung antara lingkungan luar robot (air) dengan bagian dalam robot. Jika air masuk ke dalam badan robot menyebabkan robot tidak dapat berfungsi.

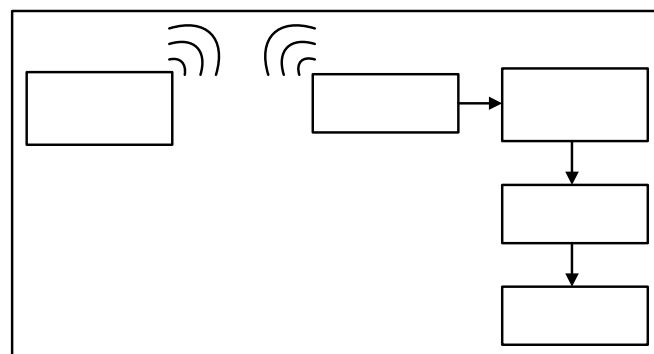
Karena robot penyelam nantinya dapat dikontrol dengan menggunakan *remote control*, maka perencanaan *software* dibutuhkan dalam pembuatan program.

Secara umum, perencanaan dalam pembuatan robot penyelam meliputi tiga bagian utama, yaitu :

- a. Bagian *hardware*, meliputi :
 - Rangkaian mikrokontroler yang dipakai.
 - Mengetahui karakteristik dari *receiver remote control*.
 - Rangkaian *driver* yang digunakan untuk mengontrol motor DC.
- b. Bagian mekanik, meliputi :
 - Analisa dari *gear box* yang digunakan.
 - Perencanaan peletakan komponen di dalam badan robot.
 - Perencanaan *shielding* untuk poros *propeller*.
 - Perencanaan beban yang akan digunakan.
- c. Bagian *software*, meliputi :
 - Pembuatan *flowchart*.
 - Pembuatan program untuk mikrokontroler.

3.1. Perencanaan Perangkat Keras / *Hardware*

Gambar 3.2. merupakan blok diagram perangkat keras / *hardware* dari robot penyelam. Dari blok diagram tersebut dapat diketahui bahwa rangkaian *hardware* yang harus dibuat meliputi : rangkaian mikrokontroler dan rangkaian *driver* motor.



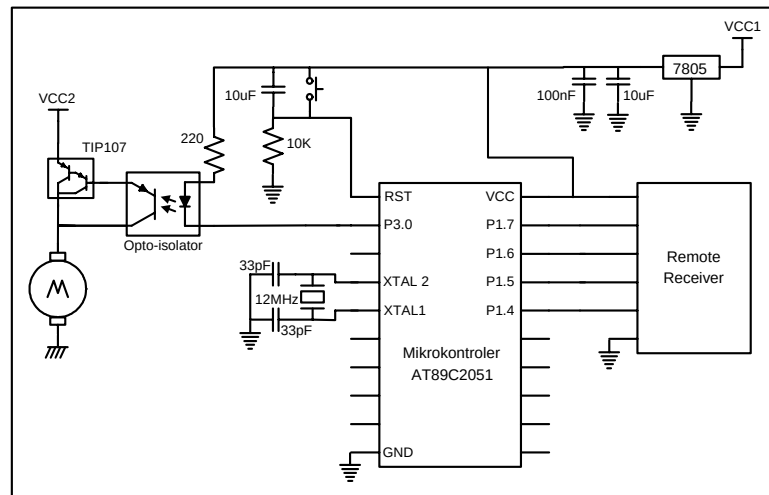
Gambar 3.2. Blok Diagram Perangkat Keras

3.1.1. Mikrokontroler

Pin – pin yang digunakan dalam mikrokontroler AT89C2051 adalah P1.7, P1.6, P1.5, P1.4, dan P3.0. P1.7, P1.6, P1.5, dan P1.4 digunakan sebagai *port* masukan yang diterima dari modul RF, sedangkan P3.0 digunakan sebagai *port* keluaran yang terhubung dengan *driver* motor.

Sebagai *port* masukan, P1.7. terhubung dengan tombol 4 pada rangkaian *receiver remote*. P1.6. terhubung dengan tombol 3 pada rangkaian *receiver remote*. P1.5. terhubung dengan tombol 2 pada rangkaian *receiver remote*. P1.4. terhubung dengan tombol 1 pada rangkaian *receiver remote*.

Gambar 3.3. merupakan skematik dari rangkaian mikrokontroler yang dipakai.



Gambar 3.3. Skematik Mikrokontroler

Catu daya yang digunakan untuk rangkaian menggunakan dua catu daya, yaitu catu daya untuk mikrokontroler (V_{CC1}) dan catu daya untuk motor (V_{CC2}). Catu daya untuk mikrokontroler menggunakan baterai 9 volt yang terhubung dengan 7805CT sehingga menghasilkan keluaran 5 volt yang digunakan untuk catu daya mikrokontroler dan *receiver* RF. Catu daya untuk motor DC akan dibahas pada sub bab perencanaan *driver* motor.

3.1.2. Karakteristik Dari *Receiver Remote Control*

Rangkaian *receiver* dari *remote control* terdiri atas 4 *pin* keluaran dan 2 *pin* untuk catu daya dan *ground*.

Dari pengujian yang telah dilakukan, dapat diketahui karakteristik keluaran dari rangkaian *receiver remote control*. Tegangan keluaran dari *receiver* tergantung dari besar tegangan yang diberikan pada pin catu daya (V_{CC}). Untuk lebih jelas mengenai keluaran dari *receiver remote* dapat dilihat pada tabel 3.1.

Tabel 3.1. Karakteristik Keluaran *Receiver*

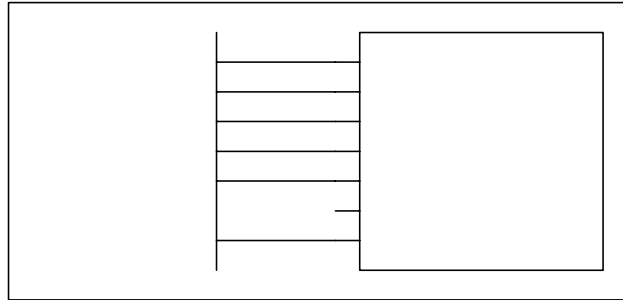
		Keluaran			
		Pin 1	Pin 2	Pin 3	Pin 4
Penekanan	Tombol 1	V_{CC}	0	0	0
	Tombol 2	0	V_{CC}	0	0
	Tombol 3	0	0	V_{CC}	0
	Tombol 4	0	0	0	V_{CC}

Sinyal keluaran yang timbul pada *receiver remote* berupa *latching* tegangan, bukan berupa pulsa. Dengan demikian, didalam perancangan dibutuhkan suatu tombol khusus yang berfungsi untuk me-*reset* semua keadaan yang timbul akibat penekanan tombol, misalnya tombol 4 sebagai tombol *reset*.

Pada saat *receiver remote* sudah diberi tegangan dan tombol belum tertekan, keluaran dari tiap *pin* adalah 0 volt (*ground*).



Gambar 3.4. *Receiver Dan Transmitter Remote Control*

Gambar 3.5. Skematik *Receiver Remote*

Gambar 3.4. merupakan gambar dari rangkaian *transmitter* dan *receiver remote control*. Dari gambar tersebut tampak bahwa *remote control* yang digunakan adalah *remote control* 4 kanal. Gambar 3.5. merupakan rangkaian *receiver remote* yang terhubung dengan mikrokontroler.

3.1.3. *Driver Motor*

Motor DC yang digunakan sebagai penggerak *propeller* adalah motor DC dengan tegangan operasional sebesar 9 volt. Catu daya yang digunakan untuk *driver* motor (V_{CC2}) adalah 8 buah baterai charge @ 1,2 volt, sehingga didapat catu daya untuk *driver* motor sebesar 9,6 volt.

Driver yang digunakan untuk mengontrol motor adalah *opto-isolator* yang terhubung dengan transistor. Transistor tersebut digunakan untuk meningkatkan arus yang mengalir pada motor DC. Alasan penggunaan *opto-isolator* sebagai *driver* adalah sebagai pengaman agar arus balik yang timbul tidak merusak mikrokontroler. Oleh karena itu pemasangan *ground* pada motor tidak digabung dengan *ground* pada rangkaian mikrokontrolernya. Rangkaian *driver* motor yang digunakan aktif “low”. Gambar 3.6. merupakan rangkaian lengkap dari *driver* motor.

Rangkaian *driver* motor (pada *opto-isolator*) terdiri atas dua bagian, yaitu bagian LED dan bagian *detector*. Bagian LED berfungsi sebagai kontrol untuk bagian *detector*. Rangkaian dari *driver* bagian LED untuk *opto-isolator* dapat dilihat pada gambar 3.7.

GND

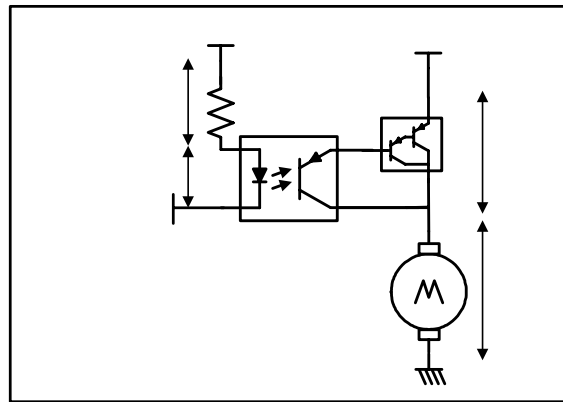
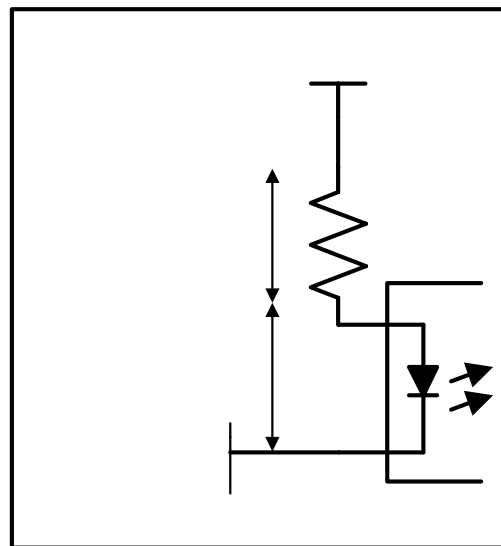
P1.4

P1.5

P1.6

P1.7

VCC

Gambar 3.6. Rangkaian *Driver*Gambar 3.7. Rangkaian *Driver* Bagian LED

Dari *datasheet* diketahui bahwa maksimal I_{OL} tiap *pin* per-*port* sebesar 20 mA dan maksimal total I_{OL} untuk semua *pin* keluaran sebesar 80 mA. I_{OH} sebesar – 12 uA tiap *pin*-nya.

Dari gambar 3.7. diketahui bahwa besar arus yang masuk ke dalam *pin* mikrokontroler ditentukan oleh besar nilai resistor R. Nilai resistor R didapat dengan menggunakan rumus :

$$R = \frac{V_R}{I}$$

dimana nilai V_R didapat dari :

$$V_R = V_{CC1} - V_{LED}$$

diketahui dari *datasheet* bahwa besar nilai *typical* dari V_{LED} sebesar 1,15 volt dan besar nilai *typical* dari I_{LED} sebesar 16mA, sehingga didapat nilai V_R sebesar :

$$V_R = 5 - 1,15$$

$$V_R = 3,85 \text{ Volt}$$

dengan memakai besar nilai arus yang melalui LED dan masuk ke mikrokontroler sama dengan arus *typical* dari LED, dengan demikian nilai resistor R yang dibutuhkan sebesar :

$$R = \frac{V_R}{I}$$

$$R = \frac{3,85}{16 \cdot 10^{-3}}$$

$$R = 240 \Omega$$

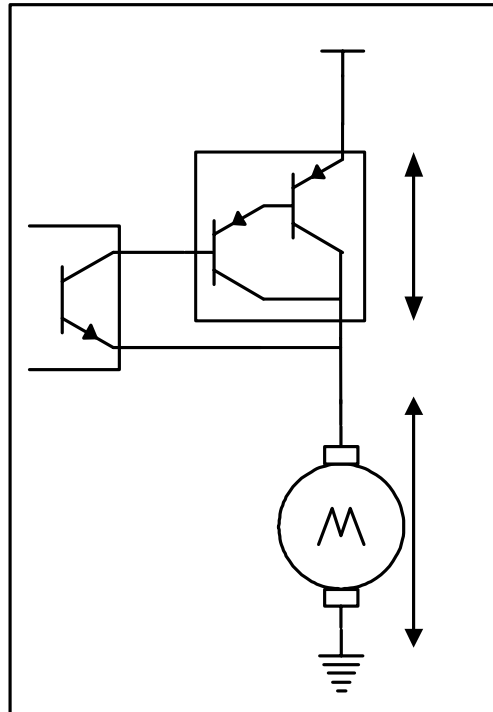
Karena besar nilai hambatan resistor tersebut di pasaran tidak ada, maka digunakan resistor yang mempunyai nilai hambatan yang mendekati dari nilai hambatan resistor tersebut. Oleh karena itu digunakan resistor yang mempunyai nilai hambatan sebesar 220Ω , sehingga arus yang mengalir pada mikrokontroler sebesar :

$$I = \frac{V}{R}$$

$$I = \frac{3,85}{220}$$

$$I = 17,5 \text{ mA}$$

sedangkan arus I_{OL} maksimal per *pin* pada mikrokontroler sebesar 20 mA dan arus maksimal dari LED sebesar 50 mA, sehingga resistor 220Ω dapat dipakai.



Gambar 3.8. Rangkaian *Driver* bagian *Detector*

Dari gambar 3.8. dapat diketahui besar tegangan pada motor saat transistor saturasi, sebesar :

$$V_{motor} = V_{CC2} - V_{CE \text{ saturasi}}$$

karena besar V_{CE} pada saat saturasi sebesar 1,1 volt (TIP 107 merupakan transistor Darlington), sehingga :

$$V_{motor} = 9,6 - 1,1$$

$$V_{motor} = 8,5 \text{ volt}$$

Dari gambar 3.7. didapat persamaan – persamaan dari variabel – variabel arus yang mengalir untuk motor, yaitu:

$$I_{motor} = I_{C.TIP107} + I_{E.opto}$$

karena $I_{C.TIP107} \gg I_{E.opto}$, maka dapat dianggap :

$$I_{motor} = I_{C.TIP107}$$

$$I_{B.TIP107} = I_{C.opto}$$

$$I_{C.TIP107} = hfe.I_{B.TIP107}$$

sehingga didapat persamaan yang menghubungkan antara arus motor dengan arus *opto*, didapat persamaan :

$$I_{motor} = hfe \cdot I_{C.opto}$$

Dari pengukuran dengan menggunakan avometer secara langsung, didapat arus motor DC maksimal sebesar 2,5 A ($I_{motor} = 2,5 \text{ A}$).

Transistor yang dipakai yang terhubung dengan *opto-isolator* adalah transistor dengan tipe TIP 107. Dari *datasheet* TIP 107 diketahui bahwa transistor TIP 107 mempunyai *hfe* minimal sebesar 1000. sehingga didapat besar arus $I_{C.opto}$ yang mengalir sebesar :

$$I_{C.opto} = \frac{I_{motor}}{hfe}$$

$$I_{C.opto} = \frac{2,5}{1000}$$

$$I_{C.opto} = 2,5 \text{ mA.}$$

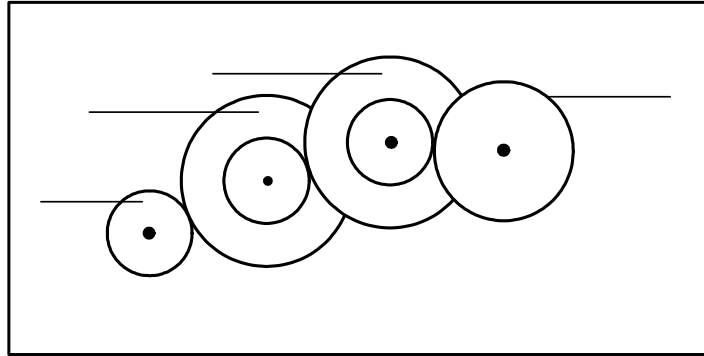
dari *datasheet* diketahui bahwa arus I_C maksimal *opto-isolator* (arus *detector* maksimal) sebesar 50 mA. Oleh karena itu, transistor TIP107 dapat dipakai sebagai penguat arus dari rangkaian *driver* tersebut.

3.2. Perancangan Mekanik

3.2.1. Analisa Gear Box

Pada gambar 3.9. dapat dilihat rangkaian dari roda gigi / *gear box* yang yang digunakan untuk memperkuat torsi dari motor. Roda gigi - roda gigi yang dipakai di dalam *gear box* tersebut meliputi :

- a. 1 buah roda gigi kecil (*spur gear*) yang melekat pada poros motor dengan jumlah gigi adalah 8.
- b. 2 buah roda gigi (*spur gear*) yang memiliki 2 macam roda gigi yaitu roda gigi dengan diameter besar dan roda gigi dengan diameter kecil yang terhubung poros. Jumlah gigi pada roda gigi dengan diameter besar adalah 18 sedangkan pada roda gigi dengan diameter kecil berjumlah 24.
- c. 1 buah roda gigi (*crown gear*) yang berfungsi untuk menggerakkan poros *propeller* dengan jumlah gigi adalah 20.



Gambar 3.9. Gear Box

Roda gigi 2

Dari gambar 3.9. dapat diketahui besar penguatan torsinya. Dengan memisalkan variabel – variabel yang dipakai adalah :

- M1 sebagai torsi dari motor.
- M6 sebagai torsi pada poros propeller.
- z_1 sebagai jumlah gigi pada roda gigi 1 (terhubung dengan poros motor)
- z_2 sebagai jumlah gigi diameter besar dan z_3 sebagai jumlah gigi diameter kecil pada roda gigi 2
- z_4 sebagai jumlah gigi diameter besar dan z_5 sebagai jumlah gigi diameter kecil pada roda gigi 3
- z_6 sebagai roda gigi 4 (yang terhubung dengan poros *propeller*)

Dengan demikian didapat rangkaian gerak dari *gear box* tersebut adalah :

- Roda gigi 1 / z_1 (poros motor) menggerakkan roda gigi 2 dengan diameter besar (z_2).
- Roda gigi 2 dengan diameter kecil (z_3) menggerakkan roda gigi 3 dengan diameter besar (z_4).
- Roda gigi 3 dengan diameter kecil (z_5) menggerakkan roda gigi 6 / z_6 (poros *propeller*).

Telah diketahui sebelumnya bahwa penguatan torsi yang terjadi dapat diketahui dengan menggunakan *speed ratio*. *Speed ratio* adalah perkalian dari semua jumlah gigi pada *driving gear* dibagi dengan perkalian semua jumlah gigi pada *driven gear*, sehingga didapat :

$$i = \frac{z_1}{z_2} \cdot \frac{z_3}{z_4} \cdot \frac{z_5}{z_6}$$

Dengan memasukkan besar nilai – nilai dari $z_1 = 8$, $z_2 = z_4 = 24$, $z_3 = z_5 = 18$, dan nilai dari $z_6 = 20$, dapat diketahui nilai dari *speed ratio* pada *gear box* yang dipakai adalah sebesar :

$$i = \frac{8}{24} \cdot \frac{18}{24} \cdot \frac{18}{20}$$

$$i = 0,225$$

sehingga didapat penguatan torsi sebesar :

$$M_6 = \frac{M_1}{i}$$

$$M_6 = \frac{1}{0,225} \cdot M_1$$

$$M_6 = 4,44 \cdot M_1$$

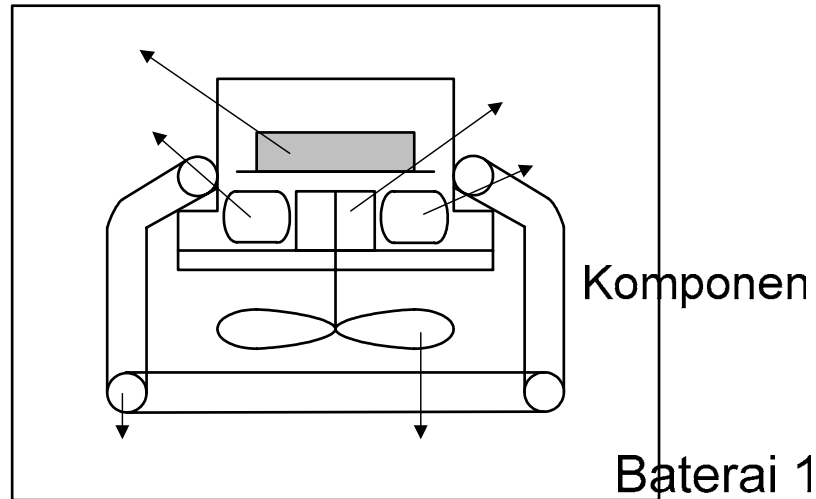
Jadi penguatan torsi pada poros *propeller* yang terjadi (secara perhitungan) sebesar 4,44 kali dari torsi yang dihasilkan oleh motor.

3.2.2. Peletakan Komponen Mekanik

Peletakan komponen di dalam badan robot diatur sedemikian rupa hingga tidak ada ruang yang kosong di dalam robot tersebut. Hal tersebut dikarenakan semakin banyak ruang yang kosong (semakin banyak udara) di dalam badan robot menyebabkan gaya apung / gaya dorong ke atas yang diterima oleh robot menjadi lebih besar sehingga beban yang digunakan sebagai pemberat menjadi lebih besar pula.

Hal yang perlu diperhatikan dalam meletakkan komponen adalah komponen – komponen yang berat seperti baterai diletakkan pada lapisan bawah dan komponen – komponen yang ringan di atasnya. Hal tersebut dilakukan agar robot menjadi lebih seimbang didalam air.

Gambar peletakan komponen dari robot dapat dilihat pada gambar 3.10. Dari gambar tersebut dapat dilihat bahwa letak poros *propeller* harus berada di tengah – tengah agar robot seimbang pada saat naik dan pada saat menyelam.

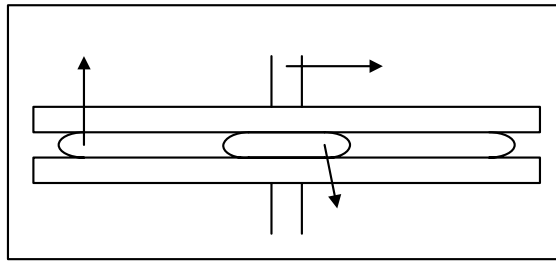


Gambar 3.10. Penempatan Komponen

3.2.3. Perencanaan *Shielding* Poros *Propeller*

Poros *propeller* merupakan bagian yang paling rawan pada robot tersebut karena poros *propeller* merupakan bagian yang bergerak dan menghubungkan secara langsung antara lingkungan di luar robot. Jika poros *propeller* tidak rapat atau bocor, maka air dari luar dapat masuk ke bagian dalam robot yang dapat menyebabkan robot tidak berfungsi.

Untuk mencegah masuknya air pada poros *propeller* dibutuhkan suatu *shielding*. *Shielding* untuk poros *propeller* menggunakan karet oring yang di sekelilingnya diberi *sealer*. Karet oring yang dipilih adalah karet yang pas dengan diameter poros *propeller*. Karet oring yang pas adalah karet oring yang mampu “menggigit” batang poros dari *propeller* tetapi tidak terlalu kencang. Hal tersebut disebabkan jika karet oring longgar atau tidak menggigit batang poros, maka air dapat merembes melalui celah oring. Sedangkan jika karet oring yang dipakai terlalu kencang, menyebabkan poros *propeller* menjadi sukar berputar atau berat untuk berputar. Gambar 3.11. merupakan cara *shielding* yang dipakai pada poros *propeller*.

Gambar 3.11. *Shielding Poros Propeller***Sealer**

3.2.4. Perencanaan Beban yang Digunakan

Beban yang digunakan terbuat dari besi beton. Bentuk dari beban dibuat melingkar mengelilingi robot dengan kaki – kaki sebanyak 6 buah. Pada bagian bawah kaki, ditambah besi beton yang menghubungkan keenam kaki tersebut. Hal ini dilakukan untuk mencegah patahnya kaki – kaki beban tersebut.

Berat beban yang digunakan tergantung dari besar PWM yang digunakan. Pada saat kondisi motor mati, tidak ada gaya dorong yang timbul dari motor yang menyebabkan robot dapat naik ke atas. Sehingga berat beban yang diperlukan untuk menjaga agar robot dalam kondisi setimbang / melayang adalah sama dengan gaya apungnya. Berat beban inilah yang disebut dengan beban ideal.

Pada saat kondisi motor aktif, ada gaya yang timbul dari motor yang menyebabkan robot terangkat ke atas. Oleh sebab itu diperlukan beban yang lebih berat untuk menjaga agar robot dalam kondisi melayang tersebut. Semakin besar siklus kerja yang digunakan, semakin besar pula gaya yang menyebabkan robot terangkat ke atas (sehingga diperlukan beban yang lebih berat untuk menjaga agar robot tetap melayang).

3.2.5. Tekanan Hidrostatik

Tekanan hidrostatik yang dialami oleh robot pada kedalaman 70 cm (dasar akuarium) sebesar :

$$P = P_0 + \rho_{air} \cdot g \cdot h$$

Dimana :

$$P_0 = 1 \text{ atm} = 10^5 \text{ Pa}$$

$$\rho_{air} = 1000 \text{ Kg/m}^3$$

$$g = 9,8 \text{ m/s}^2$$

$$h = 70 \text{ cm} = 0,7 \text{ m}$$

sehingga didapat :

$$P = 10^5 + 1000 \cdot 9,8 \cdot 0,7$$

$$P = 10^5 + 6860$$

$$P = 106860 \text{ Pa} = 1,07 \text{ atm}$$

Jadi tekanan hidrostatik yang dialami oleh robot sebesar 1,07 atm pada kedalaman 70 cm.

3.3. Perencanaan Perangkat Lunak / *Software*

Kecepatan putar motor dikontrol dengan menggunakan PWM yaitu dengan memberikan pulsa hidup dan mati. Lebar pulsa hidup dan mati ditentukan oleh besar *delay* yang digunakan pada program.

Pada program robot tersebut terdapat 3 proses utama, yaitu proses untuk menjalankan prosedur konstan, proses untuk menjalankan prosedur naik, dan proses untuk menjalankan prosedur turun.

Prosedur konstan yang dimaksud adalah prosedur yang digunakan untuk membuat robot dapat berenang secara konstan di ketinggian tertentu. Prosedur ini bekerja pada saat tombol 1 ditekan. Prosedur ini terletak pada program utama.

Prosedur naik adalah prosedur yang digunakan supaya robot dapat berenang naik ke permukaan. Prosedur ini dimulai pada saat tombol 2 yang terhubung dengan P1.5. ditekan.

Prosedur turun adalah prosedur yang digunakan agar robot dapat menyelam atau turun ke bawah. Prosedur ini berkerja pada saat tombol 3 yang terhubung dengan P1.6. ditekan.

Robot dikontrol dengan menggunakan 4 tombol untuk 5 kondisi yang terjadi. Kondisi tersebut antara lain :

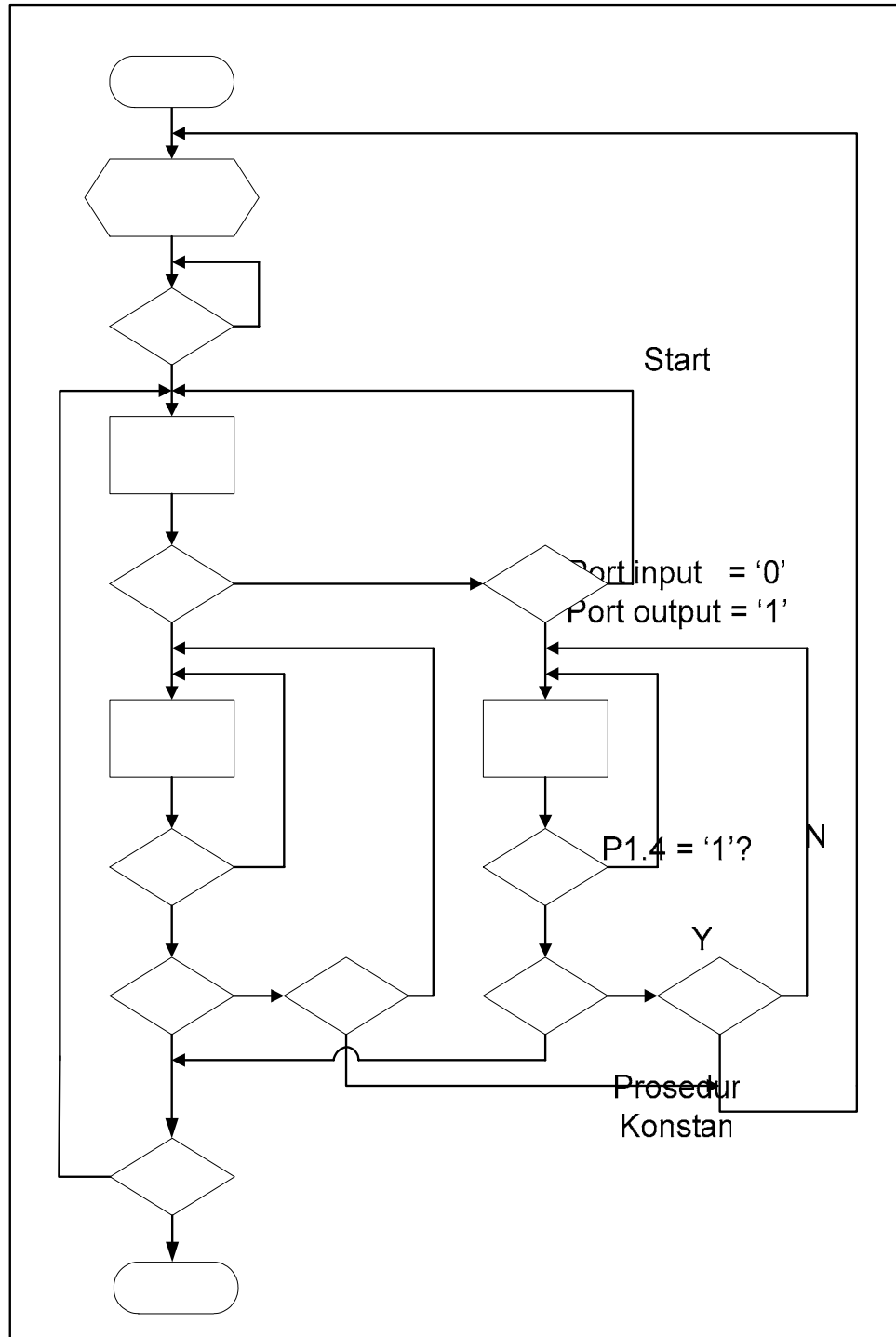
- Penekanan tombol 1 digunakan sebagai kondisi untuk *start* robot.
- Penekanan tombol 2 digunakan sebagai kondisi agar robot dapat berenang naik.
- Penekanan tombol 3 digunakan sebagai kondisi agar robot dapat menyelam (gerakan turun).

- Penekanan tombol 4 kemudian tombol 1 digunakan sebagai *reset* dari tombol 2 dan tombol 3 dan kembali ke program utama.
- Penekanan tombol 4 kemudian tombol 2 digunakan sebagai kondisi untuk mematikan robot tersebut.

Robot akan bergerak pada saat tombol 1 ditekan. Pada saat itu robot akan berenang menempati posisi di kedalaman tertentu di dalam air. Pada saat tombol 2 ditekan, robot akan berenang naik dan untuk me-*reset*-nya dibutuhkan penekanan tombol 4 kemudian tombol 1. Pada saat tombol 3 ditekan, robot akan menyelam dan untuk me-*reset*-nya dibutuhkan penekanan tombol 4 kemudian tombol 1. Jika tidak terdapat penekanan pada tombol 2 ataupun tombol 3, maka robot akan tetap berada pada posisinya. Untuk mematikan robot dibutuhkan adanya penekanan tombol 4 kemudian tombol 2.

3.3.1. *Flowchart*

Flowchart atau diagram alir dari program yang akan disusun dapat dilihat pada gambar 3.12.



Gambar 3.12. Flowchart Program

3.3.2. Program List

Berikut ini merupakan program yang dipakai oleh mikrokontroler :

```

ORG 0H
    SJMP MAIN

MAIN:
    JB P1.4, UTAMA        ; menunggu penekanan tombol 1
    SJMP MAIN

UTAMA:
    JB P1.5, NAIK          ; penekanan tb.2, lompat ke naik
    JB P1.6, TURUN        ; penekanan tb.3, lompat ke turun
    JB P1.7, EMPAT        ; penekanan tb.4, lompat ke empat
    CLR P3.0              ; program konstan di kedalaman tertentu
    ACALL DELAY            ;
    SETB P3.0              ;
    ACALL DELAY            ;
    SJMP UTAMA

EMPAT:
    MOV R0, #20            ; besar pengulangan timer (timer 1 detik)
LOOP1:
    MOV TMOD, #00010000B   ; pengaktifan timer
    MOV TL1, #0B0H         ;
    MOV TH1, #3CH         ;
    CLR TF1                ;
    SETB TR1               ;
    JNB TF1, $             ; menunggu pelimpahan timer
    CLR TR1
    JB P1.5, MAIN          ; penekanan tb.2, lompat ke main
    JB P1.4, UTAMA        ; penekanan tb.1, lompat ke utama
    DJNZ R0, LOOP1
    SJMP UTAMA

NAIK:
    CLR P3.0              ; prosedur untuk naik
    JB P1.7, EMPAT
    SJMP NAIK

TURUN:
    SETB P3.0             ; prosedur untuk turun
    JB P1.7, EMPAT
    SJMP TURUN

DELAY:
    MOV R2, #4FH          ; prosedur delay
DELAY1:
    MOV R3, #0FFH
    DJNZ R3, $
    DJNZ R2, DELAY1
    RET

END

```