

# PERANCANGAN SISTEM KENDALI MODEL KAPAL UNTUK UJI *FREE RUNNING DAN TURNING CIRCLE*

## *Design of Ship Model Control Sistem for Free Running and Turning Circle Maneuver Test*

**Chandra Permana dan Totok Soedarto**

UPT Balai Pengkajian dan Penelitian Hidrodinamika, BPPT-Surabaya  
Email: chandra.permana@gmail.com  
totoksoedarto@gmail.com

Diterima: 2 Mei 2015; Direvisi: 28 Mei 2015; Disetujui: 3 Juli 2015

### **Abstrak**

Pada uji *free running* dan *turning circle maneuver*, pergerakan model kapal perlu dikendalikan oleh operator menggunakan *remote control*. Kecepatan pergerakan maju mundur model kapal ditentukan oleh *thrust* akibat putaran *propeller*. Sedangkan perubahan *heading* model kapal ditentukan oleh sudut kemudi *rudder*. Perancangan sistem kendali ini dimaksudkan mengolah inputan kecepatan *propeller* dan sudut kemudi *rudder* yang ditentukan oleh operator melalui *remote control*. Pengolahan inputan tersebut menggunakan aplikasi berbasis *software* NI LabVIEW FPGA yang di-*embedded* kedalam *controller* cRio 9012 National Instruments. Output dari *controller* berupa sinyal digital PWM dengan *duty cycle* tertentu untuk menggerakkan motor DC yang memutar *propeller* dan menggerakkan motor DC lainnya untuk mengubah sudut *rudder*.

**Kata kunci :** *free running, turning circle, kecepatan rotasi propeller, sudut rudder, NI Labview FPGA.*

### **Abstract**

*In the free running and turning circle test, a ship models movement needs to be controlled by the operator using a remote control. The speed of forward or backward movement is determined by thrust of propeller rotation. Movement of the ship model heading is determined by the rudder angle. The design of this control system is intended to process the input speed of the propeller and rudder angle that is determined by the operator via the remote control. The input processing using applications based on NI LabVIEW FPGA software that is embedded into the controller National Instruments cRio 9012. The output of the controller is Pulse Width Modulation (PWM) digital signal with a certain duty cycle to drive DC motor that rotates the propeller and to drive other DC motor which changes rudder angle.*

**Keywords :** *free running, turning circle, propeller rotation speed, rudder angle, NI Labview FPGA.*

## **PENDAHULUAN**

Pada pengujian *free running*, model kapal dijalankan dengan kecepatan tertentu dengan arah heading memiliki sudut tertentu terhadap arah datangnya gelombang. Sedangkan pada pengujian *turning circle maneuver* kapal digerakkan lurus kemudian dibelokkan dengan sudut kemudi sebesar 35 derajat pada kondisi air tenang (Anonim, 2004). Untuk

uji *free running*, kecepatan model kapal ditetapkan pada nilai tertentu sedangkan heading model kapal diatur agar model kapal tetap melaju secara lurus. Pengaturan kecepatan dan heading dilakukan oleh operator menggunakan *remote control*. Pada pengujian *turning circle maneuver*, kecepatan model kapal ditetapkan pada nilai tertentu oleh operator sedangkan untuk heading kapal pertama diatur oleh operator agar

model kapal tetap melaju secara lurus kemudian sudut kemudi diubah sebesar 35 derajat sehingga kapal akan ber-*maneuver* membentuk lingkaran sebanyak minimal tiga kali secara otomatis. Selanjutnya operator akan mengambil alih kembali kendali *heading* kapal. Salah satu faktor yang menentukan kecepatan model kapal adalah besarnya *thrust* akibat putaran *propeller*. Kecepatan motor DC yang memutar *propeller* tergantung pada besarnya nilai rata-rata tegangan yang dialirkan oleh *driver motor* ke motor tersebut. Nilai rata-rata tegangan tersebut bisa diatur dengan nilai *duty cycle* dari *pulse width modulation* (PWM) yang menjadi inputan dari *driver motor*. Sedangkan untuk arah putaran motor baik *Clock Wise* (CW) ataupun *Counter Clock Wise* (CCW) ditentukan polaritas tegangan rata-rata. Arah putaran motor akan menentukan arah putaran *propeller*. Arah putaran *propeller* CW atau CCW akan menjadi faktor lain selain konfigurasi bentuk *propeller* yang menentukan gerakan maju atau mundur model kapal.

Besar sudut kemudi *rudder* kapal diatur menggunakan motor servo. Besaran derajat gerakan motor servo diatur oleh *duty cycle* PWM tertentu. Besaran *duty cycle* PWM yang digunakan untuk mengendalikan besaran kecepatan motor *propeller* dan derajat pergerakan motor servo didapat dari output *receiver remote control* yang sudah diolah terlebih dahulu melalui suatu *controller*.

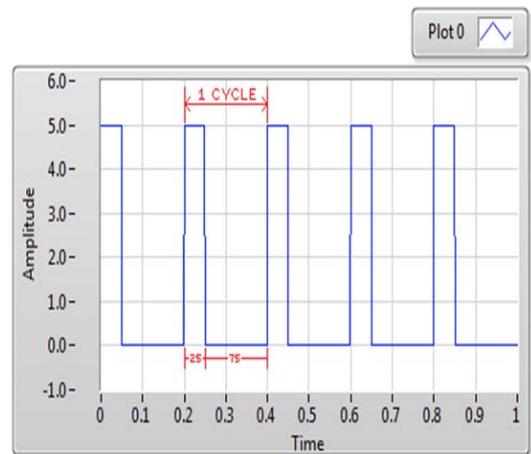
### PULSE WIDTH MODULATION (PWM)

Sinyal *Pulse Width Modulation* (PWM) adalah metoda yang digunakan untuk membangkitkan sinyal analog menggunakan sumber digital (National Instrument, 2014). Sinyal PWM berbentuk sinyal kotak terdiri dari dua komponen utama yang mendefinisikan parameter sinyal tersebut: *duty cycle* dan frekuensi. *Duty cycle* mendeskripsikan banyaknya waktu sinyal pada kondisi *high* (ON) sebagai persentase dari keseluruhan waktu yang diperlukan untuk menyelesaikan satu siklus lengkap sebagaimana dijelaskan pada persamaan (1). Frekuensi menyatakan berapa banyak siklus PWM yang bisa terjadi dalam satu detik. Satu siklus PWM sendiri terdiri dari sinyal dalam kondisi *high* (ON) dan sinyal dalam kondisi low (OFF). Banyaknya waktu (lebar pulsa) pada kondisi ON ditambah banyaknya waktu pada kondisi OFF sama dengan periode PWM untuk menyelesaikan satu *cycle*.

$$DC = \frac{t_{ON}}{t_{ON} + t_{OFF}} \times 100 \tag{1}$$

Dimana: DC : *Duty Cycle* dalam %  
 t<sub>ON</sub>: waktu sinyal ON dalam detik  
 t<sub>OFF</sub>: waktu sinyal OFF dalam detik

Contoh grafik sinyal PWM dengan *duty cycle* 25% dimana t<sub>ON</sub> sebesar 25 milidetik dan t<sub>OFF</sub> sebesar 75 milidetik ditunjukkan pada **Error! Reference source not found.**



Gambar 1 : grafik sinyal PWM dengan *duty cycle* 25%

Dengan mengulangi pembangkitan sinyal digital on dan off pada rate yang cukup cepat dan *duty cycle* tertentu maka tegangan keluaran akan tampak seperti tegangan analog dengan besaran tegangan rata-rata konstan.

Jika *driver motor* yang digunakan untuk mensuplai arus ke motor DC menggunakan power suplai maksimum 24 VDC dan *driver motor* itu menggunakan inputan sinyal PWM maka tegangan rata-rata yang dialirkan *driver motor* ke motor DC bisa hitung menggunakan persamaan (2) (Andrianto, 2008)

$$\bar{V} = \% DC * V_{max} \tag{2}$$

Dimana:  $\bar{V}$ : Tegangan rata-rata dalam volt  
 DC : *Duty Cycle* dalam %  
 V<sub>max</sub>: Tegangan maksimum dalam volt

Hubungan tegangan rata-rata dan sinyal PWM dengan *duty cycle* tertentu diperlihatkan pada Gambar 2.



Gambar 1 Hubungan *duty cycle* dengan nilai tegangan rata-rata

Tegangan rata-rata yang dialirkan oleh driver ke motor DC berupa sinyal analog kontinu bukan sinyal yang memiliki kondisi ON-OFF.

Untuk *motor servo*, besaran derajat juga ditentukan oleh *duty cycle* PWM. Biasanya dalam spesifikasi *motor servo* besarnya *duty cycle* tersebut tidak dinyatakan dalam satuan % akan tetapi dinyatakan dalam satuan jumlah waktu kondisi ON dalam milisecond. Misalnya motor servo akan bergerak 90 derajat dari titik 0 derajat bila diberi tegangan PWM dengan *duty cycle* 1050 milidetik.

### KONFIGURASI **HARDWARE**

*Hardware* yang digunakan untuk membentuk sistem kendali model kapal adalah:

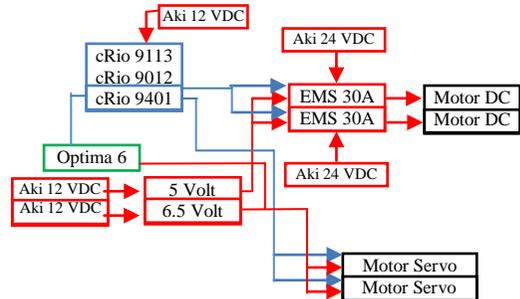
1. 1 unit *Remote Control* HiTEC Optic 6 Sport.
2. 1 unit *Receiver* HiTEC Optima 6.
3. 2 unit Motor DC SS40E8-TO.
4. 2 unit *Motor Servo* HiTEC HS-5646WP.
5. 7 unit Aki 12V 7000 mAh.
6. 1 box *Voltage Regulator* 6.5 Volt.
7. 1 box *Voltage Regulator* 5 volt.
8. 1 unit NI cRio 9012 *Controller*.
9. 1 unit NI cRio 9113 *Chassis*.
10. 1 unit NI cRio 9401 *Digital IO Module*.
11. 2 unit *Motor Driver* EMS 30 A H-Bridge.



Gambar 2 Tuas-tuas pada *remote control* hitec yang digunakan sebagai kendali model kapal

Untuk tuas kendali motor DC terdapat 14 step untuk perubahan kecepatan baik ke atas (gerak maju) dan ke bawah (gerak mundur).

Koneksi antar *hardware* di atas digambarkan pada Gambar 3.



Gambar 3 Koneksi antar *hardware* untuk system kendali model kapal

Penjelasan sistem koneksi antar *hardware* di atas adalah sebagai berikut:

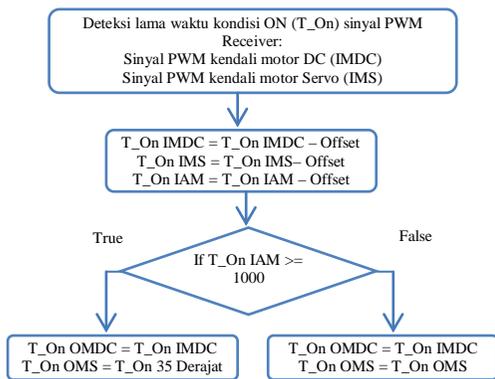
1. *Voltage regulator* 5 volt berfungsi untuk mengubah tegangan 12 VDC dari aki menjadi tegangan 5 VDC untuk digunakan sebagai power suplai driver motor EMS 30A H-Bridge.
2. *Voltage regulator* 6.5 volt berfungsi untuk merubah tegangan 12 VDC dari aki menjadi tegangan 6.5 VDC untuk digunakan sebagai power suplai Receiver HiTEC Optima 6 dan *motor servo* HiTEC HS-5646WP.
3. Receiver HiTEC Optima 6 menerima sinyal kendali dari *Remote Control* HiTEC Optic 6 Sport. Sinyal output dari receiver ini dalam bentuk sinyal PWM terbagi tiga: kendali arah dan kecepatan *propeller*, kendali arah sudut kemudi *rudder*, dan kendali sudut *rudder* manual/ otomatis tiga puluh lima derajat.
4. Sinyal dari receiver diteruskan ke *port input* NI cRio 9401 untuk selanjutnya diolah melalui NI cRio 9012 dan NI cRio 9113. Sinyal PWM output olahan tersebut kemudian diteruskan melalui *port output* NI cRio 9401 ke *motor servo* dan *motor servo*. Perangkat NI cRio disuplai oleh Aki 12 VDC.
5. Sinyal PWM dengan *duty cycle* tertentu tersebut dan tegangan 24 VDC dari aki kemudian diolah *motor driver* menjadi tegangan rerata untuk menentukan kecepatan putar motor DC sesuai dengan persamaan (2)

dan diolah juga menjadi arah putar motor DC CCW atau CW.

6. Sedangkan sinyal PWM yang menuju motor servo akan mengendalikan besarnya derajat pergerakan *motor servo*.

### ALGORITMA SISTEM KENDALI MODEL KAPAL PADA LABVIEW FPGA

Adapun diagram alir algoritma umum untuk pengolahan sinyal PWM dari *receiver* HiTEC hingga bisa digunakan untuk menggerakkan motor DC dan motor servo baik secara manual maupun auto 35 derajat ditunjukkan pada Gambar4:



Gambar4 Diagram alir algoritma umum sistem kendali model kapal

Algoritma ini kemudian diterjemahkan secara detail dalam bentuk *block diagram software* NI LabVIEW FPGA.

LabVIEW FPGA memiliki clock dengan frekuensi 40 MHz(National Instruments, 2013 atau 1 period =  $1/4e^7 = 2.5e^{-8}$  detik. Jika suatu kecepatan *timer loop* diatur kecepatan iterasinya menggunakan *clock* 40 MHz maka 1 iterasi memakan waktu =  $2.5e^{-8}$  detik. Satu iterasi dianggap 1 *tick*. Pada setiap pertambahan 1 *tick timerloop* akan memeriksa digital input (DI) pada NI cRio 9401 apakah dalam kondisi ON (*high*) atau OFF (*low*) dengan algoritma tertentu. Kondisi ON(*High*) = *Boolean True*, Kondisi OFF(*Low*) = *Boolean False*. Nilai *counter tick* pada LabVIEW FPGA sendiri tidak dimulai dari nol akan tetapi pada nilai tertentu yang bertambah sesuai *osilasi clock* pada FPGA

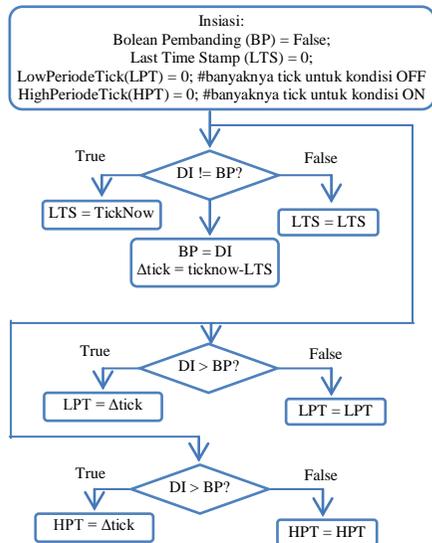
Untuk menentukan sinyal PWM berada pada kondisi ON atau OFF maka DI dari sinyal PWM pada

saat waktu tertentu dibandingkan dengan boolean pembanding (BP) dari DI Boolean pada satu waktu sebelumnya. Adapun tabel kebenaran dari beberapa perbandingan antara boolean DI dan BP dijelaskan pada Tabel1.

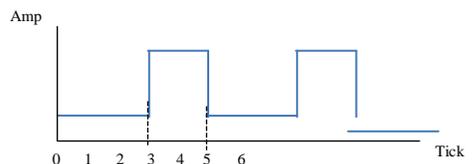
Tabel1 perbandingan dua kondisi boolean DI dan BP

DI	BP	DI!=BP	DI>BP	DI<BP
T	T	F	F	F
T	F	T	T	F
F	T	T	F	T
F	F	F	F	F

Algoritma untuk menentukan lama waktu atau period dari kondisi ON (*HIGH*) atau OFF (*LOW*) suatu sinyal PWM diperlihatkan pada Gambar 6.



Gambar 5 Algoritma untuk menentukan periode/lamanya waktu high/low dari suatu sinyal PWM



Gambar 6 Contoh sinyal PWM untuk contoh perhitungan period *HIGH* atau *LOW*.

Suatu contoh sinyal PWM seperti yang diperlihatkan pada

Gambar 6, penerapan algoritma dari diagram alir

pada Gambar 6 untuk menentukan lama waktu atau period dari kondisi ON (*HIGH*) atau OFF (*LOW*) dijelaskan sebagai berikut:

Nilai *offset* dari masing-masing *High Period* sinyal PWM yang diterima dari receiver HiTEC baik untuk motor DC, motor servo maupun otomatis/manual *rudder* didapat dari persamaan (3):

$$Offset = \frac{HighPeriodeZero}{2.5e - 8} \quad (3)$$

Dimana:

- *offset*(tick)
- *High Period Zero* (detik). Nilai ini merupakan *High Period* pada saat tuas kontrol di *Remote Control* untuk motor DC, motor *Servo* pada posisi netral (nol) sedangkan tuas kontrol

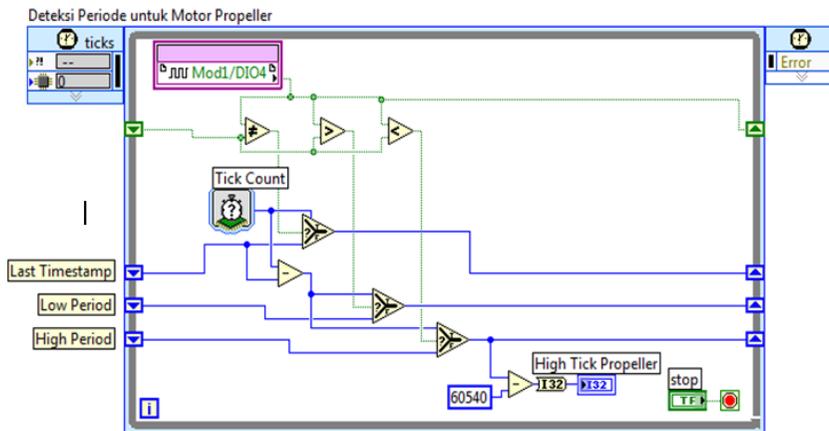
untuk Otomatis/Manual pada posisi manual. Nilai *High Period Zero* ini diukur menggunakan osiloskop.

- Konstanta  $2.5e^{-8}$  (detik/tick) merupakan waktu yang diperlukan satu iterasi *time loop* (1 tick).

*Block diagram* LabVIEW FPGA untuk masing-masing sinyal PWM adalah:

1. *Block diagram* di LabVIEW FPGA untuk mendeteksi *high period* sinyal PWM kendali motor DC *propeller* dari receiver HiTEC yang masuk melalui *portDI4* cRio 9401 ditunjukkan pada gambar 7.

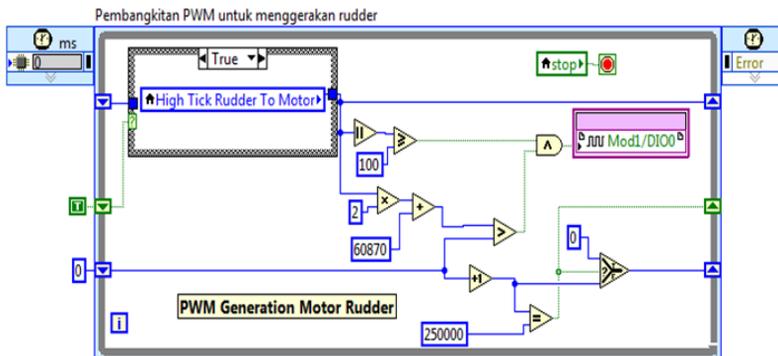
$$Offset = 60540 \text{ tick.}$$



Gambar 7 Block Diagram mendeteksi nilai *high period* sinyal PWM kendali motor DC untuk *Propeller*

2. *Block Diagram* di Labview FPGA untuk membangkitkan sinyal PWM untuk menentukan besaran derajat pergerakan motor

*servo* sudut kemudi *rudder* ditunjukkan pada gambar 8.



Gambar 8 Pembangkitan PWM untuk menggerakkan Motor *Servo Rudder*



Pada *block diagram* ini, *high period* yang berasal dari *block diagram* otomatis/manual rudder menjadi parameter *high period* untuk membangkitkan sinyal PWM yang akan menggerakkan motor *servo*.

Pada setiap iterasi *timer loop*, ada dua kondisi yang harus selalu bernilai *true* untuk menghasilkan kondisi *high(on)* pada *port digital output* 0 cRio 9401 yang terhubung ke motor *servo*

Kondisi 1.

Inisiasi cacah *tick* = 0. Cacah *Tick* = cacah *Tick*+1

$Tick == 250000?$  (250000 totalperiod ON OFF untuk 1 cycle PWM).

Jika iya cacah *tick* reset = 0

Jika tidak cacah *tick*= cacah *tick*.

Apakah (*high period tick* dari *Block Diagram* auto/manual\*2+ *offset high period* motor *servo*) > cacah *tick*?

Jika iya maka kondisi 1 *True*

Jika tidak maka kondisi 1 *False*

Kondisi 2 *true*: jika *high period* absolut (minus hilang)  $\geq 100$ . Selain itu kondisi 2 *false*. Nilai 100

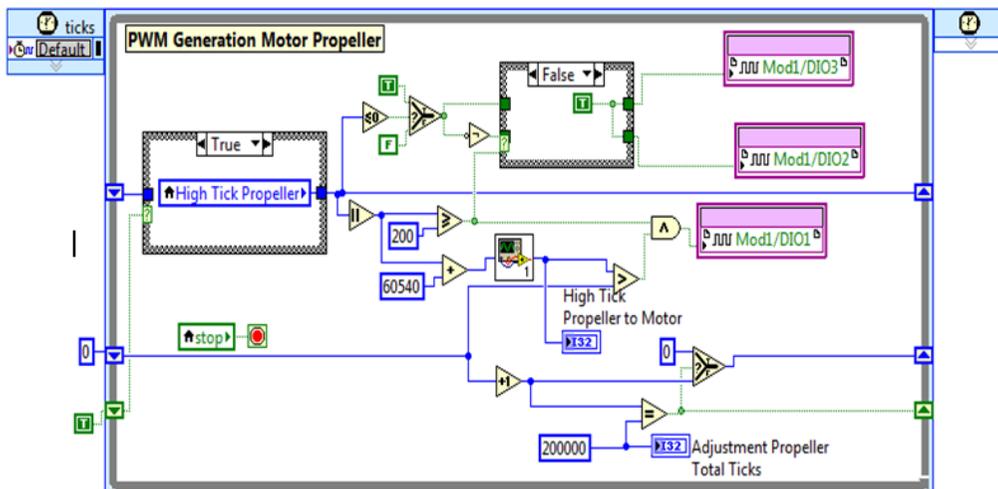
untuk mengkompensasi fluktuasi nilai *high period* dari *receiver* setelah dikurangi *offset*.

Kondisi 2 harus *true* agar motor *servo* tidak bergerak acak pada saat tuas kontrol pada posisi nol gerakan acak ini bisa dikarenakan fluktuasi nilai *high period* hasil dari pengurangan *high period* motor *servo* yang dari *receiver* terhadap *offset*.

Perkalian faktor 2 dengan *high period tick* dari *Block Diagram* auto/manual dimaksudkan agar bukaan derajat dari motor *servo* lebih besar dibandingkan jika

hanya *high period tick* dari *Block Diagram* auto/manual. Dengan dikalikan faktor 2, gerakan motor *servo* sudut kemudi *rudder* ke arah kanan-kiri bisa mencapai nilai 75 derajat yang merupakan kemampuan maksimum dari motor *servo* tersebut.

3. *Block Diagram* di Labview FPGA untuk membangkitkan sinyal PWM untuk menentukan besaran kecepatan dan arah putar motor DC *propeller* ditunjukkan pada 9.



Gambar 9 Pembangkitan PWM untuk menggerakkan motor DC Propeller

Dari data sheet motor driver EMS 30A didapat tabel kebenaran untuk arah putar dan kondisi stop motor DC yang ditunjukkan pada Tabel 2 (Anonim, 2009).

Port DO2 cRio 9401 terhubung dengan terminal MIN1 pada *driver* motor EMS 30A. Port DO3 terhubung dengan terminal MIN2. Port DO1 terhubung dengan terminal MPWM.

Tabel2 Tabel kebenaran driver motor DC EMS 30A

Status kerja modul-Bridge	Input dan Status					Output	
	MPWM	MIN1	MIN2	MEN1	MEN2	MOUT1	MOUT2
Forward	H	H	L	H	H	V MOT	MGND
Reverse	H	L	H	H	H	MGND	V MOT
Brake to GND	H	L	L	H	H	MGND	MGND
Brake to VCC	X	H	H	H	H	V MOT	V MOT

Terminal MIN1 dan MIN2 menentukan arah putar motor sedangkan terminal MPWM menentukan kecepatan putar motor berdasarkan *high period* yang dibangkitkan.

Pada *block diagram* ini, *high period* yang berasal dari *block diagram* deteksi *high period* sinyal PWM motor DC menjadi parameter *high period* untuk membangkitkan sinyal PWM yang akan menggerakkan motor DC *Propeller*.

Pada setiap iterasi *timer loop*, ada dua kondisi yang harus selalu bernilai *true* untuk menghasilkan kondisi *high(on)* pada *port digital output 1 cRio 9401* yang terhubung ke terminal MPWM *driver motor*.

Kondisi 1.

Inisiasi cacah *tick* = 0. Cacah *Tick* = cacah *Tick*+1  
 $Tick == 200000?$  (200000 total *period ON OFF* untuk 1 *cycle* PWM untuk *driver* motor DC).

Jika iya cacah *tick* reset = 0

Jika tidak cacah *tick* = cacah *tick*.

*High period* sinyal PWM *output* dari *block diagram* Gbr 8+ *offset high period* motor DC masuk ke modul subvi "adjustment tick".

*High period* tersebut perlu diatur sebelum masuk ke motor *driver* dikarenakan jika hanya menggunakan *high period* yang diterima dari *receiver* maka perubahan kecepatan motor DC untuk masing-masing kenaikan step dari 14 step tidak terlalu signifikan. *Adjustment* dilakukan dengan menentukan batas atas dan batas bawah untuk *high period* yang dihasilkan *receiver* pada setiap *step*. Batas atas-bawah ini untuk mengkompensasi fluktuasi nilai *tick high period* yang mungkin muncul. Jika *high period* yang dihasilkan *receiver* berada pada rentang batas atas-bawah setiap *step* maka *high period* tersebut dirubah nilainya menjadi *high period* tertentu (*duty cycle* tertentu) yang nilainya konstan untuk *step* tersebut. *High period* tertentu ini yang kemudian dikirim ke *driver* motor.

Cek apakah (*high period adjustment*) > cacah *tick*

Jika iya maka kondisi 1 *True*

Jika tidak maka kondisi 1 *False*

Kondisi 2 *true*: jika *high period* absolut (minus hilang)  $\geq 200$ . Selain itu kondisi 2 *false*. Nilai 200 untuk mengkompensasi fluktuasi nilai *high period* dari *receiver* setelah dikurangi *offset*.

Kondisi 2 harus *true* agar putaran motor DC tidak acak pada saat tuas kontrol pada posisi nol. Gerak acak ini bisa dikarenakan fluktuasi nilai *high period* hasil dari pengurangan *high period* motor DC yang dari

*receiver* terhadap *offset*.

Jika kondisi dua *false*, maka nilai DO2 dan DO3 = ON sehingga motor berhenti berputar sesuai tabel kebenaran pada Gbr 14.

Jika kondisi dua *true*, maka untuk:

- Output *high period* dari *block diagram* Gbr 8  $\leq 0$ , DO 2 ON dan DO 3 OFF. Motor berputar pada arah CW.
- Output *high period* dari *block diagram* Gbr 8  $> 0$ , DO 2 OFF dan DO 3 ON. Motor CCW.

## KESIMPULAN

Sistem kendali model untuk uji *free running* dan *turning circle maneuver* difokuskan pada pengendalian kecepatan dan arah motor *propeller* untuk pengendalian kecepatan dan arah maju mundur kapal serta pengendalian derajat bukaan motor *servo* untuk pengendalian gerak sudut kemudi *rudder*. Pada pengujian *turning circle maneuver* didesain sistem kendali otomatis yang menggerakkan sudut kemudi *rudder* pada posisi 35 derajat tanpa harus dijaga pada posisi tersebut oleh operator.

Pada pembangkitan sinyal *high period* PWM untuk motor *servo* kemudi *rudder*, diperlukan faktor pengali 2 terhadap nilai *high period* PWM dari *receiver*. Dengan demikian nilai *high period* yang dikirim ke motor *servo* kemudi *rudder* menjadi lebih besar dan bukaan derajat sudut kemudi bisa maksimum hingga 75 derajat (baik ke arah kiri-kanan). Pada pembangkitan sinyal *high period* PWM untuk motor DC diperlukan suatu *adjustment* untuk setiap *step* perubahan dituas kontrol. Ini dikarenakan perubahan kecepatan yang terlalu kecil jika hanya berdasarkan sinyal *high period* dari *receiver*. Arah putar dan stop-gerak putaran motor DC merujuk ke tabel kebenaran *driver* motor.

## DAFTAR PUSTAKA

- Andrianto, H. (2008). *Pemograman Mikrokontroler AVR ATMEGA16 Menggunakan Bahasa C (CodeVision AVR)*. Bandung. Penerbit Informatika.
- Anonim, (2004) *Laporan Simulasi Numerik Manuver (Kapal Patroli Cepat 40 M)*; UPT – BPPH Surabaya.
- Anonim. (2009). *EMS 30A H-Bridge Manual*.

Innovativeelectronic.com. Surabaya.

National Instrument. (2014). *What is a Pulse Width Modulation (PWM) Signal and What is it Used For?*.<http://digital.ni.com/public.nsf/allkb/294E67623752656686256DB800508989>.

National Instrument. (2013).*Reconfigurable Chassis for CompactRio cRIO-911x*.  
[www.ni.com/datasheet/pdf/en/ds-227](http://www.ni.com/datasheet/pdf/en/ds-227)